Tratamiento de la Degradación debida al Canal en Sistemas de Reconocimiento Remoto

Angel Manuel Gómez García

Departamento de Teoría de la Señal, Telemática y Comunicaciones

Universidad de Granada

Editor: Editorial de la Universidad de Granada
Autor: Ángel Manuel Gómez García
D.L.: Gr. 2162 - 2006
ISBN: 84-338-4138-6

D. Antonio M. Peinado Herreros y **Dña. Victoria Sánchez Calle**, Profesores Titulares de Universidad del Departamento de Teoría de la Señal, Telemática y Comunicaciones

CERTIFICAN:

Que la memoria titulada: **"Tratamiento de la Degradación debida al Canal en Sistemas de Reconocimiento Remoto"** ha sido realizada por **Angel M. Gómez García** bajo nuestra dirección en el Departamento de Teoría de la Señal, Telemática y Comunicaciones de la Universidad de Granada para optar al grado de Doctor en Informática.

Granada, a 28 de Septiembre de 2006

Fdo. Antonio M. Peinado Herreros
Director de la Tesis

Fdo. Victoria Sánchez Calle
Director de la Tesis

A mis padres y a Angeles.

Agradecimientos

Quisiera expresar mi más sincero agradecimiento a todas aquellas personas que, directa o indirectamente, han contribuido a la realización de la presente Tesis Doctoral. En especial a los profesores Antonio Peinado y Victoria Sánchez, directores de esta tesis, por su enorme interés y dedicación, su ilusión y su guia. Al profesor Ben Milner y su grupo de investigación en Norwich, por su cálida acogida en las frías tierras inglesas. A mis compañeros de departamento, especialmente a los que seguimos atrincherados en la Facultad de Ciencias, por su incondicional apoyo e inestimable ayuda. Y a mis familiares y amigos, por estar a mi lado durante las horas bajas.

Resumen

El acceso a la información en cualquier momento y lugar se ha convertido en algo no tan sólo deseable sino casi necesario. Sin embargo, los nuevos dispositivos móviles, cada vez más pequeños, dificultan y entorpecen dicho acceso, haciéndose patente la necesidad de interfaces de usuario mejoradas. La recuperación de información por medio de algo tan cotidiano como la voz se plantea como posible solución pero, desgraciadamente, la introducción de un subsistema de reconocimiento en tales dispositivos plantea serias limitaciones. Esto ha impulsado la aparición de un nuevo paradigma, denominado reconocimiento remoto de la voz, en donde el reconocimiento se realiza fuera del propio dispositivo móvil, en un servidor remoto.

En esta aproximación, el dispositivo de usuario ha de transmitir la voz, o una representación de ésta apropiada para el reconocimiento, por un canal de comunicación. Actualmente, el panorama de las telecomunicaciones está marcado por el desarrollo de las redes de telefonía móvil (como GSM) y las redes IP de conmutación de paquetes (e Internet), cuyo crecimiento ha ido en constante aumento en los últimos años. Estos dos tipos de redes cuentan con la ventaja de tener un alcance prácticamente global, posibilitando el reconocimiento de la voz y el acceso a la información de forma ubicua. Sin embargo, como desventaja, ambas redes resultan propensas a errores. En la telefonía móvil se dispone de un canal de radio altamente expuesto al ruido, mientras que en las redes IP, al no estar diseñadas para la transmisión de datos en tiempo real, serán propensas a la pérdida de paquetes. Como es de esperar, estos errores de canal tienen un impacto negativo importante en el rendimiento del reconocedor.

El interés de esta tesis se centra en el análisis de la influencia sobre el reconocimiento de voz de los dos tipos de redes introducidos anteriormente, así como a la propuesta y posterior desarrollo de diferentes soluciones para prevenir,

reducir y compensar los efectos degradantes de estos canales. Con este fin ambos canales serán unificados bajo el concepto de canal con pérdidas, en donde segmentos completos de información o bien se pierden o se descartan. Como resultado de estas pérdidas, y dependiendo de cómo y qué información de voz se transmita (la voz completa codificada o sólo los parámetros necesarios para el reconocimiento), dos degradaciones de distinta naturaleza afectarán la señal de voz y sus parámetros, reduciendo severamente la precisión del reconocimiento.

Si la voz se transmite codificada empleando un codificador de voz predictivo, las memorias a largo plazo presentes generalmente en estos codecs dan lugar a una degradación adicional y posterior al error o pérdida, al que denominamos ruido de memoria. En esta tesis evaluamos la influencia de este ruido de memoria sobre el reconocimiento y proponemos dos técnicas para tratarlo. La primera intenta mitigar este ruido una vez se ha producido, modelándolo en el dominio del cepstrum mediante factores aditivos, de forma similar a como se reduce el ruido acústico. La segunda, en cambio, trata de prevenir su aparición por medio de la transformación directa de los parámetros de voz en vectores de características para el reconocimiento, a lo que nos referiremos como transparametrización.

Por su parte, la degradación causada debido a la propia pérdida de información puede tratarse en el receptor mediante técnicas de mitigación. En esta tesis presentamos diferentes técnicas estadísticas, como la estimación MMSE o la estimación MAP. Dichas técnicas ofrecen buenos resultados gracias a la fuerte similitud que presenta la voz consigo misma a corto plazo. Sin embargo, por esta misma razón, están limitadas a pérdidas consecutivas (o ráfagas) cortas, donde sea improbable que la voz evolucione a otro sonido. Así, se hacen necesarias técnicas que fragmenten las ráfagas en otras más cortas. Con tal objetivo, en esta tesis se propone el uso de códigos FEC específicos así como el entrelazado de tramas. Igualmente, se examina el tratamiento de las pérdidas en el propio reconocedor, combinando este procesamiento con el uso de códigos FEC y entrelazado.

Índice general

Índice de figuras

ÍNDICE DE FIGURAS

Índice de tablas

Capítulo 1

INTRODUCCIÓN

Aunque con limitaciones, el reconocimiento automático del habla por medio de un computador es un hecho hoy en día. El esfuerzo investigador realizado en el campo del reconocimiento de voz por una parte, y por otra la evolución de los computadores, cada día más potentes y baratos, han hecho posible que aplicaciones que hace poco parecían propias de la ciencia-ficción estén hoy ya superadas. Así, actualmente es posible encontrar, entre otros, productos comerciales de dictado y reconocimiento de habla continua, sistemas de navegación y control mediante la voz, así como sistemas de diálogo para el acceso y recuperación de información contenida en bases de datos.

Paralelamente a este desarrollo de las tecnologías de reconocimiento, han surgido nuevos canales digitales para la transmisión de la voz. Las redes GSM de telefonía móvil permiten la comunicación oral en prácticamente cualquier momento y lugar, desvinculando completamente de su ubicación física tanto al receptor como al emisor. Por otra parte, las redes de conmutación de paquetes basadas en el protocolo IP, gracias a su enorme capacidad para la interconexión de redes diversas, han dado origen a una red de redes global o Internet. El panorama actual de las telecomunicaciones esta marcado por el desarrollo de estas dos redes, cuyo crecimiento ha ido en constante aumento en los últimos años. Prueba del ello son los datos sobre el número de suscriptores conectados a la red GSM, cercano a los 1500 millones, y la estimación realizada por el *Internet Systems Consortium* en Enero de 2006, que apunta hacia un mínimo de unos 395 millones de ordenadores conectados directamente a la Red (esto es, sin tener en cuenta las posibles subredes internas que pudieran colgar de ellos).

Las redes GSM e IP han supuesto una verdadera revolución en el panorama de las telecomunicaciones, posibilitando una miríada de nuevas y prometedoras aplicaciones a las

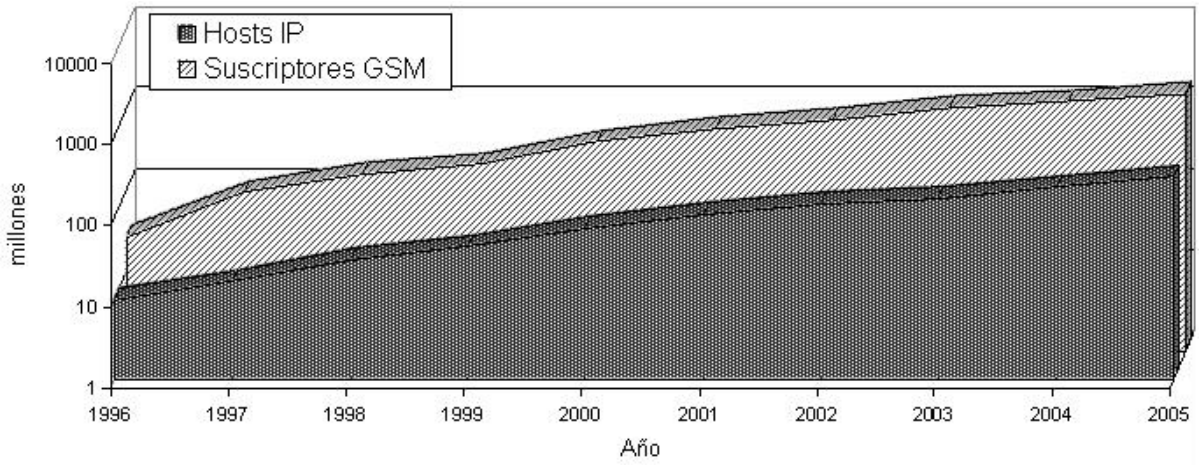


Figura 1.1: Evolución de las redes GSM e IP en los últimos años. Fuentes: Internet Systems Consortium (IP), y EMC World Cellular Database (GSM).

que el reconocimiento de voz no es ajeno. Así, en Octubre de 1998 se formó el *W3C Voice Browsers Working Group* [1] como centro de coordinación entre los diferentes grupos que trabajan en especificaciones relacionadas con el habla en Internet, como por ejemplo, la gestión de diálogos y extensiones a los actuales estándares Web (*VoiceXML*), formatos para la especificación de gramáticas (*SRGS - Speech Recognition Grammar Specification*), interpretación semántica, etc. Por su parte, las redes GSM han impulsado un nuevo reto, el reconocimiento de voz a través de dispositivos móviles que permitan el acceso a servicios activados por voz.

El acceso a la información mientras se está en movimiento se ha convertido en una ventaja estratégica y su demanda está en auge en los últimos años. De hecho, la tercera generación de móviles (3G) o UMTS (*Universal Mobile Telecommunication System*) tiene por objetivo proporcionar a cada usuario un ancho de banda varias veces superior a GSM, a fin de permitir una mayor transferencia de información y, sobre todo, de nuevos contenidos multimedia. Sin embargo, debido a las reducidas dimensiones de los nuevos dispositivos móviles, los interfaces tradicionales (particularmente el teclado) dificultan el acceso a los servicios de información. La interacción oral con dichos servicios se propone como un nuevo medio de acceso, más rápido y mucho más natural. Desgraciadamente, existen serios inconvenientes para introducir en un dispositivo móvil un subsistema de reconocimiento de voz. Entre ellos, la limitada capacidad de cálculo, que condiciona la potencia y flexibilidad del reconocedor, así como el precio que llegaría a alcanzar. Ante estas limitaciones se plantea la idea de realizar el reconocimiento fuera del propio dispositivo móvil, esto es, en un servidor remoto.

Esta aproximación resulta igualmente atractiva para los nuevos servicios de voz que se quieren añadir a Internet. Esto se debe a la inversión de dependencias que implica entre el sistema de reconocimiento, proveedor de servicios y clientes. En un sistema tradicional, el sistema de reconocimiento depende del usuario. Esto no sólo supone que el usuario debe encargarse del mantenimiento de su sistema de reconocimiento (por ejemplo, mediante actualizaciones), sino que limita al proveedor de servicios a las posibilidades del reconocedor del usuario. En una arquitectura de reconocimiento remoto, el proveedor propone tanto los servicios como el reconocedor, adaptando este último a las necesidades de los primeros, facilitando la adición de nuevos servicios y liberando al usuario de su mantenimiento. Por otra parte, al compartir el reconocedor entre múltiples usuarios, se optimiza el uso de recursos. Además, la gran aceptación de las redes IP ha originado una fuerte demanda de posibilidades interconexión con las redes móviles que, en gran parte, ha propiciado la aparición de una generación móvil intermedia conocida como 2.5G que introduce la conmutación de paquetes de radio o GPRS (General Packet Radio Service).

Si bien este nuevo paradigma, conocido como reconocimiento remoto de la voz, posibilitaría una ingente cantidad de servicios y aplicaciones, también presenta algunos problemas que deben resolverse. Entre estos problemas destacan la presencia de ruido acústico, ya que se puede operar en cualquier lugar, y la influencia de un canal de comunicaciones, puesto que la voz debe ser transmitida desde el terminal del usuario hasta la localización del servidor. En este trabajo nos centramos en este último problema. Esto es, estudiaremos la influencia sobre el reconocimiento de voz de los dos tipos de redes introducidos anteriormente, a saber, las redes de móviles comunicaciones basadas en el estándar europeo GSM y las redes actuales de conmutación de paquetes basadas en el protocolo TCP/IP, proponiendo diferentes soluciones para prevenir, corregir y compensar los efectos degradantes de estos canales. Además, atenderemos a las diferentes arquitecturas propuestas para llevar a cabo el reconocimiento remoto y su disponibilidad actual.

Así, en el siguiente capítulo revisaremos brevemente los conceptos implicados en el reconocimiento del habla, centrándonos a aquellos aspectos que resulten relevantes para el reconocimiento remoto, como la parametrización de la voz, el reconocimiento mediante modelos de Markov y los criterios de evaluación comúnmente utilizados. En dicho capítulo presentaremos también el sistema de reconocimiento del habla empleado como base durante la evaluación de las técnicas propuestas.

En el capítulo 3 examinaremos los problemas derivados del canal durante el reconocimiento sobre las redes GSM e IP. Para ello, describiremos sucintamente estas redes y presentaremos las dos arquitecturas propuestas actualmente para realizar reconocimiento

remoto de la voz. Una parte importante del capítulo estará dedicada a identificar los errores o efectos degradantes que el canal causa a la información de voz transmitida, así como al estudio de la influencia que tiene cada uno de estos errores de cara al reconocimiento. Como veremos, a pesar de sus diferencias, los canales GSM e IP pueden considerarse como canales que pierden segmentos completos de información. Según como se codifique la voz, diferentes degradaciones se producirán durante el reconocimiento.

Cuando la voz se transmite usando un codificador predictivo, como ocurre en GSM, además de la propia pérdida de información, un efecto degradante adicional se extiende más allá de los errores debido a la memoria del codec. El capítulo 4 estará dedicado a la mitigación de este efecto. Para conocer los mecanismos que lo originan, examinaremos los codificadores de voz más extendidos de GSM, recurriendo a la transparametrización como técnica que evita parcialmente su aparición.

Los capítulos 5 y 6 estarán dedicados al tratamiento de los efectos degradantes ocasionados por la propia pérdida de información. En el capítulo 5 propondremos técnicas que sólo requieren de la colaboración del receptor para la mitigación de las pérdidas. En el capítulo 6, en cambio, se propondrán técnicas que se valdrán del resto de etapas implicadas en el reconocimiento remoto. En ambos capítulos se trabajará sobre voz codificada específicamente para ser reconocida, dentro del paradigma conocido como reconocimiento distribuido de la voz.

Finalmente, un capítulo dedicado a las conclusiones, contribuciones realizadas y líneas futuras de investigación cierra este trabajo.

Capítulo 2

RECONOCIMIENTO AUTOMATICO DEL HABLA

2.1. El reconocimiento automático de la voz

La comunicación oral es sin duda una de las capacidades más fascinantes del ser humano. Sin este medio para comunicar ideas fluídamente probablemente la sociedad actual no habría podido alcanzar el grado de desarrollo social, intelectual y tecnológico que actualmente posee. El intercambio de ideas por medio de la voz nos resulta una tarea cotidiana que realizamos con toda naturalidad aunque, si es analizado en profundidad, el proceso de comunicación oral no es ni mucho menos sencillo. Precisamente, la naturalidad que caracteriza la comunicación oral inspira la continua creación de instrumentos capaces de producir y reconocer voz que, imitándonos a nosotros mismos, sean capaces de llevar a cabo tareas guiadas a través de algo tan cotidiano como un diálogo.

El reconocimiento automático de la voz trata de reproducir de forma automática una parte del proceso de la comunicación oral. Ésta consiste en la decodificación automática de la señal acústica producida por el aparato fonador humano, a fin de reconstruir el mensaje contenido en ella. El reconocimiento de voz por una máquina es también conocido como Reconocimiento Automático del Habla (RAH). El objetivo último del RAH es la comunicación hombre-máquina, abarcando un amplio espectro de aplicaciones: acceso a sistemas de información automáticos, ayuda a minusválidos, traducción automática, transacciones bancarias automáticas, control oral de sistemas, etc.

El RAH es un paradigma relativamente reciente. Hasta que el espectrógrafo no fue desarrollado en la década de los años 40, no comenzaron los primeros trabajos referentes

al reconocimiento automático de voz. Uno de los primeros, capaz de discernir los diez dígitos ingleses con una tasa de reconocimiento de hasta el 98 % para un sólo locutor, fue desarrollado en los Laboratorios Bell en 1952 por Davis, Bidulph y Balashek. Aunque muy rudimentario, este sistema ya establecía las pautas generales de lo que es un sistema de reconocimiento. Más adelante, en los años 60, comienza de forma generalizada la utilización de computadores para RAH, debido a la habilidad de éstos para realizar procesamientos complejos y tareas de almacenamiento. Desde entonces, el estudio y desarrollo de nuevas técnicas, junto con un desarrollo espectacular de los computadores, han permitido grandes avances en el RAH, llegando a la aparición de aplicaciones comerciales de reconocimiento de voz continua.

2.1.1. Planteamiento general del problema

El objetivo de los sistemas de reconocimiento automático del habla consiste en permitir la interacción hombre-máquina a través de la voz. Estos sistemas intentan reproducir la forma en que la señal de voz llega a ser comprendida por el ser humano. Por analogía se suelen distinguir varios niveles de procesamiento [2]:

- Nivel Acústico. Este nivel se corresponde con todo el proceso que se realiza en el oído. La señal se procesa hasta obtener una serie de características fundamentales, eliminando las redundancias.
- Nivel Fonético. Las características de las etapa anterior se comparan y se identifican con las unidades sonoras básicas (fonemas, sílabas, palabras,...) que conoce el receptor.
- Nivel Sintáctico. Empleando una serie de reglas, que conforman la gramática del lenguaje reconocido, comunes al emisor y al receptor, las unidades sonoras básicas se combinan formando unidades conceptuales sintácticamente correctas.
- Nivel Semántico. En esta etapa se pretende realizar una comprensión del mensaje, eliminándose interpretaciones sin sentido.

En cada uno de estos niveles se introduce información generalmente parcial e imprecisa acerca de la comunicación oral, por ejemplo: cómo se produce la voz, cómo es percibida por el oído humano, la morfología y gramática del lenguaje, etc., siendo necesario hacer cooperar todas estas informaciones. Por esta razón, estos niveles pueden estar absolutamente interrelacionados entre sí, no siendo posible abordar un nivel de forma independiente.

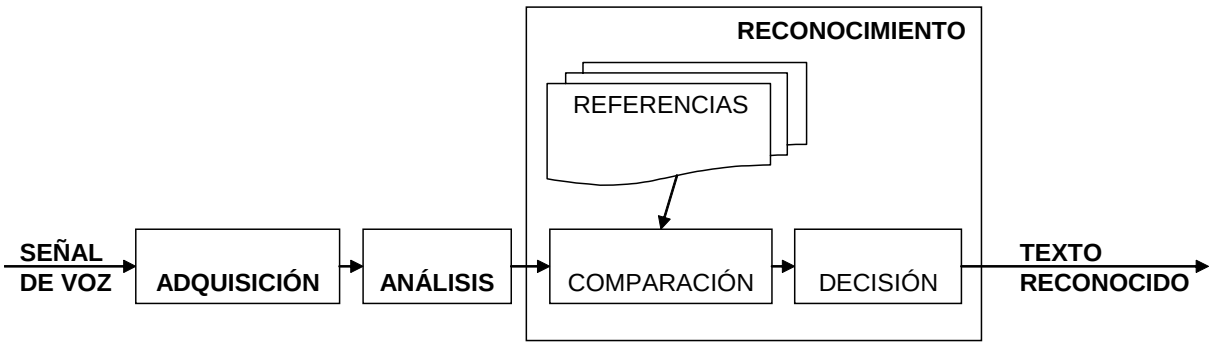


Figura 2.1: Diagrama funcional básico de un sistema de reconocimiento.

Los sistemas de reconocimiento suelen considerar el RAH como un problema de reconocimiento de formas. La estructura general de un sistema de reconocimiento planteado como un problema de reconocimiento de formas se muestra en la figura 2.1 [3]. En ella se pueden distinguir los siguientes bloques:

- El bloque de adquisición. En primer lugar es necesario adquirir la señal para realizar el reconocimiento. En este bloque se realizan las operaciones necesarias para obtener la señal digital de voz. En el micrófono, la onda acústica es convertida en una señal eléctrica analógica. Mediante un proceso de amplificación, filtrado, muestreo y codificación, se convierte esta señal analógica en una señal digital.

- El bloque de análisis o de representación tiene por objeto representar la señal de voz de una forma adecuada para el reconocimiento. En este bloque se suelen incluir varias operaciones que conducen a la vectorización de la señal. A la salida de este bloque la señal queda usualmente representada como una secuencia de *vectores de características*.

- El bloque de reconocimiento. En el bloque de reconocimiento se dispone de un conjunto de referencias que representan los distintos objetos a reconocer. En este bloque se realiza la asociación entre vectores de características y referencias reconocidas, reconstruyendo el texto correspondiente al mensaje oral.

Mientras que el bloque de análisis adecua la señal de voz para su tratamiento posterior, el bloque de reconocimiento constituye el núcleo del sistema de reconocimiento. De forma general en este bloque se dispone de: 1) un conjunto de referencias que representan los distintos objetos a reconocer, 2) un módulo de comparación y 3) un módulo de decisión.

El conjunto de referencias se obtiene mediante una fase previa al reconocimiento denominada *fase de entrenamiento*. Estas referencias representan las unidades de reconocimiento básicas que se emplean en el sistema RAH, pudiendo ir desde fonemas a frases completas. Los vectores de características recibidos de la etapa anterior son comparados, según la aproximación al RAH elegida, con este conjunto de referencias. Mediante un mecanismo de decisión, que no necesariamente ha de estar implementado en un módulo independiente, se decide que referencia aproxima mejor la entrada, construyéndose el texto reconocido.

2.1.2. Clasificación de los Sistemas de Reconocimiento

La construcción de un sistema RAH cuya versatilidad y capacidad fuese equivalente a oyente humano no es posible actualmente. El reconocimiento de voz se ve dificultado en la práctica por una serie de problemas adicionales que imponen la necesidad de establecer una serie de restricciones para reducir la complejidad del reconocimiento. Según las restricciones que se impongan en la construcción de un sistema de reconocimiento, se pueden establecer las diferentes categorías que se describen a continuación [4].

Conjunto de usuarios

Según el conjunto de usuarios para los que esté preparado el sistema se puede realizar la siguiente clasificación:

- Monolocutor. El sistema es entrenado con un solo locutor, y las pruebas de test de fiabilidad del sistema son realizadas con el mismo locutor. El sistema está preparado únicamente para reconocer voz emitida por éste.
- Multilocutor. El sistema es entrenado con varios locutores, y son los mismos los que realizan el test del sistema. El sistema está preparado únicamente para reconocer la voz emitida por éstos.
- Independiente del Locutor. Los conjuntos de locutores de entrenamiento y test son distintos. El sistema puede reconocer voz de cualquier locutor, independientemente de si ha intervenido o no en el entrenamiento de éste.

Obviamente la complejidad de estas categorías es creciente, debido a la mayor variabilidad de la voz. De esta forma, conforme aumenta el número de usuarios finales del reconocedor, mayor ha de ser el volumen de datos de entrenamiento, a fin de recoger el mayor espectro posible de variaciones.

Tipo de Secuencia y Unidad

El tipo de secuencias de voz que se pretenda reconocer, junto con la unidad sonora básica empleada, da lugar a una segunda clasificación de los sistemas:

- Palabras Aisladas. Las palabras se pronuncian aisladamente dejando suficiente silencio tanto al inicio como al final, de forma que es posible aislar estas palabras y reconocerlas una a una, usando las propias palabras como unidades.

- Palabras Conectadas. Las palabras se emiten sin silencios intermedios, pero la unidad es nuevamente la palabra [5]. La frase reconocida no tiene que responder a ninguna gramática definida.

- Voz Continua. En este caso se aborda el reconocimiento de frases completas sin pausas entre palabras [4, 6]. Aunque en principio no existe ninguna restricción respecto al tipo de unidad a usar, es usual el uso de unidades inferiores a la palabra. La frase reconocida debe responder a un conjunto de reglas que forman una *gramática*.

- Habla Espontánea. Es el reconocimiento de voz más complejo, en donde se aborda el reconocimiento de frases completas, sin pausas entre palabras y que pueden contener titubeos, repeticiones de palabras, falsos comienzos, etc [7, 8]. La frase reconocida no sólo debe responder a una gramática sino también a una *semántica*.

- Word-Spotting. Estos sistemas tratan de reconocer únicamente las palabras más importantes de la frase, que conforman un conjunto de palabras clave. Se usan técnicas similares a las propuestas en palabras conectadas, pero permitiendo el rechazo de las palabras extrañas [9, 10].

Robustez frente al Ruido

Los sistemas RAH aplican distintas técnicas de robustecimiento frente al ruido, siendo optimizados para las condiciones en las que han de operar. Las técnicas de robustecimiento son variadas y van desde la utilización de representaciones poco sensibles al ruido hasta la adaptación de la señal de voz a las condiciones de referencia, limpiando la señal contaminada. Otra posibilidad es entrenar o adaptar el sistema con voz adquirida en entornos similares a aquellos en los que debe operar. Por la forma de afrontar el ruido, los sistemas se pueden clasificar de forma genérica en robustos y no robustos dependiendo de que apliquen o no alguna técnica de compensación o robustecimiento frente al ruido [11].

Tamaño del vocabulario y unidades de reconocimiento

Las frases reconocidas mediante el RAH se obtienen a partir de concatenaciones posibles de las unidades de reconocimiento. Por tener significado completo, la unidad más natural para el ser humano es la palabra. El problema de usar la palabra como unidad es que el reconocimiento puede volverse extraordinariamente complejo cuándo se usan vocabularios medianamente grandes (más de 100 palabras). Por ello es usual utilizar unidades inferiores a la palabra para vocabularios con más de 100 palabras (vocabularios medianos y grandes). Los *fonemas* son las unidades lingüísticas mínimas en las que el mensaje acústico se puede dividir [12]. Sin embargo los fonemas son unidades ideales, la ejecución de un fonema puede dar lugar a sonidos diferentes denominados *alófonos*. Por ello, se emplean unidades alternativas inferiores a la palabra pero superiores al fonema como difonemas, sílabas, demisílabas o trifonemas [6, 13].

2.1.3. Aproximaciones al reconocimiento automático del habla

Como mencionábamos en la sección 2.1.1, los vectores de características obtenidos en la etapa de análisis son comparados, mediante alguna técnica, con el conjunto de referencias. Siguiendo un orden aproximadamente histórico, se pueden agrupar las diferentes aproximaciones al RAH más ampliamente utilizadas en *Comparación de Objetos*, *Aproximación Estadística* y *Aproximación Conexionista*.

Comparación de Objetos

La aproximación basada en comparación de objetos constituye una de las aproximaciones más simples al problema del RAH. En la etapa de entrenamiento se obtiene una serie de objetos de referencia o plantillas que representan distintas clases. Cada clase identifica una secuencia, como por ejemplo la palabra. En la fase de test, las entradas son comparadas con todos los objetos de referencia de forma que éstas son reconocidas como el objeto con el que se minimiza la distancia.

El primer problema que presenta esta aproximación surge del hecho de que incluso una misma palabra emitida por el mismo locutor presentará normalmente diferentes longitudes. Ésto da lugar a la necesidad de comparar objetos de distintas longitudes, siendo necesario alinear el objeto de entrada con el de referencia. Este alineamiento se puede conseguir con un método de programación dinámica denominado en la bibliografia DTW (Dynamic Time Warping). El método fue inicialmente introducido por Bellman [14] para

el alineamiento óptimo de dos objetos, y aplicado por primera vez a reconocimiento de palabras por Vintsjuk [15].

Como puede suponerse, la aproximación por comparación de objetos requiere tanta memoria como secuencias se vayan a reconocer. En el caso de reconocimiento de palabras aisladas se necesitan tantas secuencias como palabras existen en el vocabulario, mientras que para el reconocimiento continuo tantas como frases en la gramática. En el caso del reconocimiento de voz continuo existe una extensión del DTW a palabras conectadas. Ésta extensión presenta algunos problemas en la detección de las palabras contenidas en una frase emitida de forma continua, donde no hay separación explicita entre ellas y donde las palabras son influenciadas por sus inmediatas adyacentes. Estos problemas han sido resueltos con éxito por algunos métodos como el "Two-Level DP Matching" propuesto por Sakoe [16], el "Level-Building" propuesto por Myers et al. [17] o el "One-Pass DP" de Bridle [2].

Modelado estadístico

A diferencia de la comparación de objetos, en donde se representa la referencia por un objeto, en la aproximación estadística la referencia se representa mediante un modelo estadístico. De forma abreviada, el proceso de reconocimiento puede ser descrito como el cálculo de la probabilidad $P(W|X)$ de que un texto W corresponda a una señal acústica X, sobre todo el conjunto de posibles textos, a fin de encontrar aquel que proporciona el valor máximo. De esta forma, el reconocimiento de voz consistirá en la selección de aquel texto $\hat{W}$ que verifique,

$$\hat{W} = arg \max_{W} P(W|X) \tag{2.1}$$

Obtener directamente la probabilidad mencionada puede ser una tarea complicada, pero si disponemos de las probabilidades *a priori* de cada texto posible $P(W)$, utilizando la regla de Bayes, $P(W|X)$ puede ser obtenida según la expresión

$$P(W|X) = \frac{P(X|W)P(W)}{P(X)} \tag{2.2}$$

donde $P(X|W)$ es la probabilidad de la señal acústica X dado el texto W y $P(X)$ es la probabilidad a priori de la señal acústica. Puesto que $P(X)$ es común a todos los textos posibles (no depende de W), no es necesario determinarla para la llevar a cabo la selección. El reconocimiento de voz queda así dividido en dos fases: evaluación de la evidencia acústica ($P(X|W)$), y evaluación de la probabilidad de emisión del texto ($P(W)$). De

esta forma, es necesario considerar dos modelos de producción, *el modelo del lenguaje*, que determina las probabilidades a priori de cada texto posible y el *modelo acústico*, que determina la probabilidad de que el texto W genere la señal acústica X. Ambos modelos pueden ser representados mediante Modelos Ocultos de Markov (*HMM–Hidden Markov Models*) extensivamente empleados en la bibliografía. Éstos serán descritos más adelante.

Aproximación Conexionista

En la aproximación conexionista se emplean redes de unidades simples de proceso, llamadas neuronas, en donde las referencias están representadas por patrones de actividad. Por analogía con el sistema nervioso humano reciben el nombre de Redes Neuronales Artificiales (*ANN–Artificial Neural Network*). El procesamiento que realiza cada neurona es muy sencillo, consistiendo en una función de activación f (normalmente de tipo sigmoide) aplicada sobre la suma ponderada de sus entradas e_i menos un umbral T,

$$s = f\left[\sum_{i=1}^{N} w_i e_i - T\right] \tag{2.3}$$

donde w_i son los pesos de ponderación y s la salida de la neurona. Mediante un sencillo mecanismo de aprendizaje es posible adaptar los pesos w_i y el umbral T para que la neurona reproduzca ciertas funciones binarias. Sin embargo, un conjunto de neuronas aisladas es incapaz de reproducir funciones no separables linealmente (como por ejemplo, la función XOR) por lo que se agrupan formando redes. Según la topología de estas redes se puede distinguir entre Redes *Multicapa*, *Máquinas de Boltzmann* y *Redes Neuronales recurrentes*.

En las redes multicapa (*MLP–MultiLayer Perceptron*) las neuronas están organizadas por capas, de forma que la salida de cada capa de neuronas conforma la entrada de la siguiente capa. La primera y última capa se denomina, respectivamente, de entrada y de salida, mientras que las capas intermedias son llamadas capas ocultas. Durante el entrenamiento o aprendizaje, los estímulos son propagados a través de la red hacia la capa de salida. En esta capa, la respuesta obtenida es comparada con la deseada. El error cometido se propaga hacia las capas inferiores (*retropropagación*) con el objetivo de reajustar los valores de pesos y umbrales. El proceso es iterado hasta alcanzar una situación estable. En el reconocimiento, los estímulos son propagados por la red hasta la capa de salida. La neurona de esta capa que presente el valor de salida más alto corresponderá al objeto reconocido. Uno de los inconvenientes de las MLPs es que el

mecanismo de aprendizaje sólo asegura alcanzar un óptimo local, que posiblemente no sea global.

Las máquinas de Boltzmann son redes en las que las neuronas no se organizan en capas, sino que es posible la conexión de un nodo con otro cualquiera. En este caso, también se hace una distinción entre nodos ocultos y nodos visibles (de entrada y salida). El método de entrenamiento empleado es el "Enfriamiento Simulado" que conduce a un óptimo global, pero con un coste computacional 10 veces mayor a la retropropagación.

Por último, las redes neuronales recurrentes (*RNN–Recurrent Neural Network*) solucionan el problema del alineamiento temporal. Las dos redes anteriores poseen una estructura estática, es decir, un número de entradas fijo. Esto contrasta con la variabilidad temporal característica de las señales de voz. En las RNN las neuronas de la capa de salida están conectadas con las de la capa de entrada de forma que, además de una entrada para un estímulo externo, disponen de la excitación proveniente de la capa de salida. Las redes RNN se activan al recibir los estímulos externos y los propagan cíclicamente por la red, a través de las conexiones recurrentes, de forma que la salida correspondiente a una secuencia aparece en la capa de salida tras haber sido procesados todos los estímulos. En este sentido, las RNN presentan grandes analogías en su funcionamiento con los modelos HMM [18].

Aunque las ANN resultan muy atractivas para el reconocimiento de voz, presentan serios problemas de cara a su diseño. La elección de la arquitectura de la red, el número de capas o el número de neuronas no es una tarea trivial. Por otro lado, la experiencia no ha dejado clara su superioridad frente a la potencia del modelado estadístico, además de que los ANN aprenden con menor rapidez modelos temporales que los HMM [19]. Aunque algunos autores han propuesto y evaluado sistemas híbridos que combinen la potencia de las ANN y los HMM [20, 21], se prefiere el uso de estos últimos frente a los primeros.

2.2. Representación de la señal de voz

La forma de onda de la señal no es una representación adecuada para el reconocimiento de voz. Además de codificar el texto del mensaje oral, la señal de voz aporta mucha otra información que no es relevante para el proceso de reconocimiento, como la información relativa al locutor: sexo, edad, estado de ánimo, lugar de procedencia, etc. Por otro lado, la señal que se obtiene para el reconocimiento contiene información de diversas fuentes distintas de la voz del locutor, como ecos, reverberaciones o ruido ambiental, formado por ruido de aparatos, ruido del micrófono e incluso voz de otros locutores.

Por esta razón, es necesario aplicar diversas operaciones a la señal para obtener una representación apropiada. El objetivo del bloque de representación de voz es la construcción de una representación de la señal que sea compacta y exenta de redundancias, que preserve la información relativa al mensaje y que deseche el resto de informaciones. Es decir, el objetivo del bloque de representación no es otro que la eliminación de la mayor parte posible de fuentes de variabilidad, perjudiciales para el reconocimiento. El bloque de representación juega así un papel importante en los sistemas de RAH ya que, al eliminar la redundancia y al reducir la variabilidad, el proceso de reconocimiento se facilita [22].

Para obtener una representación compacta y exenta de redundancias, debemos establecer qué características de la señal de voz permiten distinguir a unos fonemas de otros y por tanto son relevantes para el reconocimiento:

- Información espectral. Un perfil espectral suavizado permite identificar los *formantes*. Dependiendo de la configuración del tracto vocal (posición de la lengua, los labios, los dientes, el velo del paladar, etc.) aparecen en la señal de voz determinadas frecuencias de resonancia o formantes. Debido a esta estrecha relación con la configuración del tracto, los formantes son el rasgo que mejor caracteriza a los fonemas. Para caracterizarlos, hay que ignorar el rizado espectral debido a los armónicos, que no son debidos a las resonancias del tracto, sino a la vibración de las cuerdas vocales.

- Frecuencia fundamental. La presencia de pulsos glotales periódicos que exciten el tracto vocal al producir la señal de voz indican que el fonema es sonoro. Así, el análisis de la frecuencia fundamental o *pitch* puede ayudar a distinguir entre vocales y consonantes sordas. Sin embargo, la frecuencia fundamental suele descartarse como característica para el reconocimiento puesto que es una importante fuente de variabilidad interlocutor (puede llegar a 80 Hz en locutores masculinos y a 500 Hz en femeninos) e intralocutor (ya que puede cambiar según el estado de animo, prosodia, contexto, etc.).

- Energía. La energía en vocales y consonantes sonoras es sensiblemente superior a la de las consonantes sordas. Correctamente normalizado, este parámetro suele presentar menor variabilidad, siendo útil para la caracterización de fonemas y frecuentemente utilizado en la representación de la voz.

La representación de la voz para reconocimiento está basada generalmente en un análisis espectral de tiempo corto. Para ello, la señal de voz es segmentada en tramas. Previamente a la segmentación en tramas, es usual aplicar un filtro de pre-énfasis a la señal muestreada para realzar las altas frecuencias. El objetivo de este pre-énfasis es compensar el efecto de los pulsos glotales y de la impedancia de radiación para centrar el análisis en las propiedades del tracto vocal. Por lo general este filtro tiene la forma $H(z) = 1 - \mu z^{-1}$ con valores de μ comprendidos entre 0.95 y 0.98.

Aunque la voz no se puede considerar globalmente como estacionaria, si puede asumirse cierta estacionariedad en segmentos cortos que, para el reconocimiento del habla, suelen establecerse teniendo en cuenta la duración típica de un fonema (entre 20ms y 50ms). Para cada segmento se emplea una ventana distinta de la rectangular con el objetivo de reducir el efecto de longitud finita o error de *leakage*. Una ventana muy común, pues supone un buen compromiso entre la resolución espectral y el rizado lateral, es la ventana de Hamming [23]. Además, las tramas están solapadas para mejorar la resolución temporal, situándose el periodo de la trama en torno a los 10 ms.

Cada trama es analizada y representada por un vector de parámetros que caracteriza el tracto vocal. Existen varias técnicas para el análisis espectral de las tramas, entre las cuales destacan el *análisis LPC*, el análisis basado en *banco de filtros* y las técnicas basadas en *modelos auditivos*.

2.2.1. Representación basada en el modelo LPC

La representación basada en el modelo de predicción lineal (*LPC–Linear Prediction Coding*) deriva de la representación de la voz empleada originalmente en los codificadores paramétricos [24], en donde la caracterización del espectro se realiza mediante un modelo digital de producción de voz. Básicamente, lo que se pretende con este modelo es caracterizar de forma independiente el tracto vocal y la señal que lo excita. Esta señal de excitación representa a la señal generada por las cuerdas vocales, que se puede simplificar como un tren de pulsos con un cierto pitch, cuando el sonido es sonoro, y ruido blanco aleatorio cuando el sonido es sordo.

En el modelo LPC el tracto vocal queda representado por un filtro lineal digital con una función de transferencia $H(z)$ cuyos parámetros son variables en el tiempo. El filtro necesario para representar el tracto vocal debería ser todo-polos para los fonemas orales, y tener ceros y polos para los fonemas nasales. Sin embargo, por simplicidad, usualmente se

suele considerar un filtro todo-polos con un orden incrementado para todo tipo de sonidos, de la forma:

$$H(z) = \frac{G}{A(z)} \tag{2.4}$$

donde G es la ganancia del filtro y $A(z)$ un polinomio en z^{-1},

$$A(z) = 1 + \sum_{k=1}^{p} a_k z^{-k} \tag{2.5}$$

El filtro queda descrito a partir de sus coeficientes, a_k, con $k = 1, \ldots, p$, que caracterizan la configuración del tracto vocal. Para estimarlos, la representación basada en el modelo LPC recurre a un *predictor lineal* de orden p, identificando los *coeficientes de predicción lineal* obtenidos con los coeficientes del filtro. El filtro $H(z)$ suele denominarse filtro LPC (así como el modelo). Información más detallada acerca de este filtro así como la justificación de porqué se emplea un predictor lineal para estimar sus coeficientes puede encontrarse en la sección 4.2.1.

Los coeficientes LPC resultan inadecuados para el reconocimiento, por lo que se suelen transformar a representación más adecuada. El cepstrum LPC, $c(n)$, es la señal temporal correspondiente a la transformada inversa de Fourier del espectro LPC logarítmico [25],[2]:

$$\hat{H}(e^{j\omega}) = \log(H(e^{j\omega})) = \sum_{n=-\infty}^{+\infty} c(n) e^{j\omega n} \tag{2.6}$$

donde $H(e^{j\omega})$ es la respuesta en frecuencia del filtro digital o espectro LPC. Este tratamiento homomórfico presenta ciertas ventajas. Por un lado, las señales mezcladas por convolución en el dominio del tiempo aparecen mezcladas de forma aditiva en el dominio del cepstrum (llamado *cuefrencia* y que corresponde al dominio del tiempo). Por otro lado, la transformación al dominio de la cuefrencia permite representar las características del tracto vocal con un reducido número de parámetros que no presentan correlaciones importantes.

Los coeficientes cepstrales que describen un filtro $H(z)$ pueden obtenerse mediante una relación recursiva que los relaciona con su respuesta impulsiva $h(n)$, supuesto causal y estable [25]. Si se consideran espectros normalizados en ganancia, $H(z) = 1/A(z)$, los coeficientes cepstrales se puede calcular mediante la siguiente recursión a partir de los

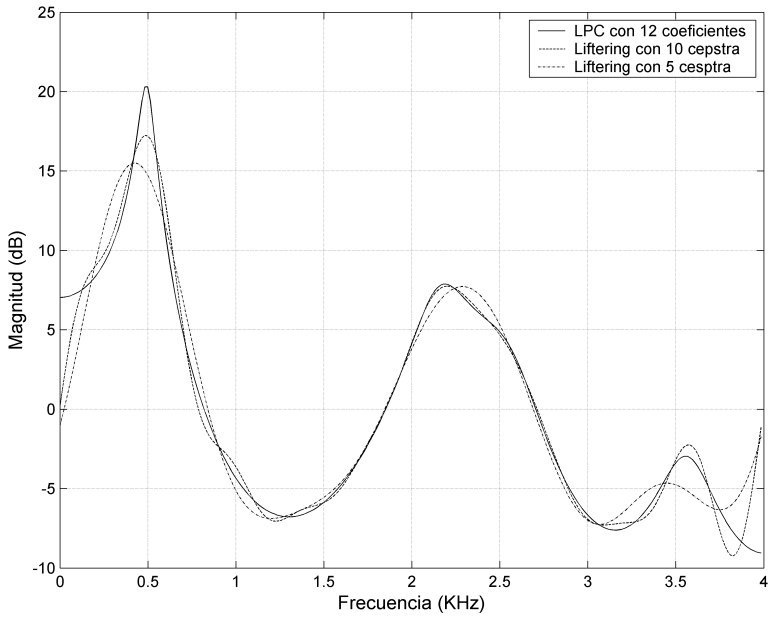


Figura 2.2: Efecto de la aplicación de ventanas rectangulares de distinto tamaño en el dominio de la cuefrencia.

coeficientes LPC,

$$c(n) = \begin{cases} 0 & ; n \leq 0 \\ -a_1 & ; n = 1 \\ -a_n - \sum_{k=1}^{n-1} \frac{k}{n} c(k) a_{n-k} & ; n > 1 \end{cases} \tag{2.7}$$

De este modo, cada trama de la señal de voz queda representada por un vector de coeficientes cepstrales. Debe tenerse en cuenta que, aunque el orden de predicción lineal sea finito (se calculan p coeficientes LPC), se pueden calcular infinitos coeficientes cepstrales. Por esta razón, hay que limitar el número de coeficientes cepstrales haciendo uso de una ventana usualmente denominada *ventana de liftering*. La limitación del número de coeficientes cepstrales conduce además a un suavizado del espectro LPC (figura 2.2) que ayuda a mejorar el rendimiento del sistema de reconocimiento [2].

2.2.2. Representación basada en banco de filtros

En las representaciones basadas en banco de filtros, la caracterización del espectro se obtiene a partir de las salidas de un conjunto de filtros pasa-banda. Los coeficientes

cepstrales se calculan a partir de un espectro suavizado, ya que cada filtro realiza un promedio pesado de las componentes espectrales presentes en su banda. De este modo, al igual que en el análisis LPC, el análisis basado en banco de filtros se centra en la obtención de la envolvente espectral. Sin embargo, la potencia de este análisis frente al análisis LPC radica en la posibilidad de definir la estructura del banco de filtros. Éste puede construirse teniendo en cuenta consideraciones perceptuales que recreen la forma en que la voz es percibida.

Debido a la anatomía del aparato auditivo humano, la respuesta subjetiva del oído no es uniforme en frecuencia. Estudios psicofisiológicos han demostrado que la habilidad humana para distinguir frecuencias sigue una función logarítmica de la frecuencia, y no lineal (como hasta ahora se ha considerado). Esta respuesta humana puede ser tenida en cuenta mediante una transformación de la escala de frecuencias a otra conocida escala *mel* o frecuencia subjetiva [2]. Por medio de experimentos psicoacústicos se puede realizar un mapeo entre la frecuencia (en Hz) y la escala mel, cuya forma se puede aproximar a partir de diversas expresiones paramétricas, por ejemplo la propuesta por Deller [26]:

$$Mel(f) = 2595 \log_{10}\left(1 + \frac{f}{700}\right) \tag{2.8}$$

donde f es la frecuencia expresada en Hertzios. La figura 2.3 muestra una gráfica de esta expresión.

En la representación basada en filtros, M filtros son distribuidos uniformemente en el espectro con respecto a la escala mel. Para construir cada uno de los filtros $H_m[k]$, se toma una serie de frecuencias centrales $f[k]$ distribuidas en escala mel de manera uniforme (figura 2.4). Por la simplicidad de cálculo que ello conlleva, generalmente se emplean filtros triangulares. La energía de salida de cada uno de estos filtros, al ser aplicados sobre el espectro de potencia obtenido a partir de una transformada discreta de Fourier, da lugar a un coeficiente por filtro. Estos coeficientes se conocen como coeficientes espectrales de frecuencia (*MFSC–Mel Frequency Spectral Coefficients*). Los coeficientes cepstrales en escala mel, comúnmente denominados *MFCC* (*Mel Frequency Cepstral Coefficients*), se obtienen aplicando una transformada coseno sobre los coeficientes MFSC:

$$c_k = \sum_{b=1}^{B} X_b \cos\left(k\left(b - \frac{1}{2}\right)\frac{\pi}{B}\right) \quad k = 1, \ldots, K \tag{2.9}$$

en donde X_b es el logaritmo de la energía de salida para el filtro pasa-banda b, c_k el coeficiente cepstral de orden k, B es el número de filtros y K el número de coeficientes

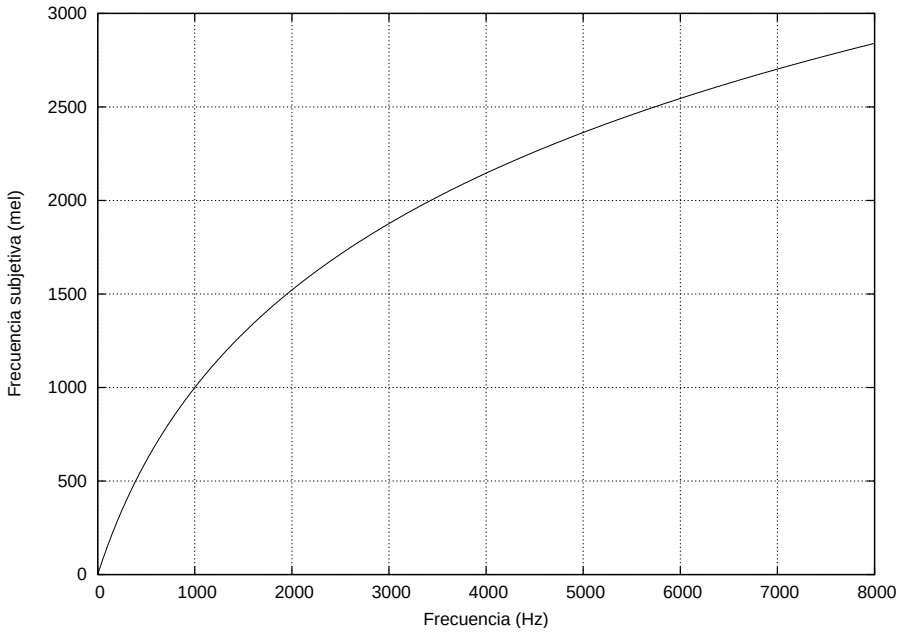


Figura 2.3: Escala Mel.

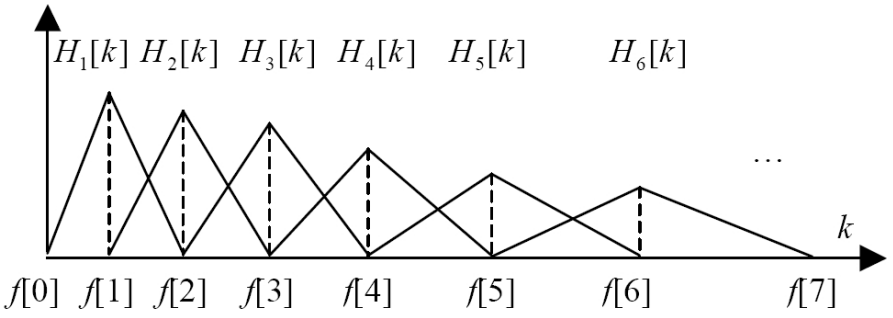


Figura 2.4: Banco de filtros triangulares en escala mel normalizado.

cepstrales calculados.

2.2.3. Representaciones basadas en Modelos Auditivos

Algunos autores han propuesto representaciones que tienen en cuenta aspectos bien conocidos del proceso de audición. Entre las representaciones basadas en el modelo de audición se encuentran el análisis PLP (*Perceptually-based Linear Prediction*), el modelo EIH (*Ensemble-Interval Histogram*) [27, 28], o los modelos auditivos síncronos, como el modelo auditivo de Seneff [29] o el modelo de predicción lineal síncrona (*SLP–Synchronous Linear Prediction*) propuesto por Junqua [30].

En condiciones acústicamente desfavorables, en las que los mecanismos de enmascaramiento o inhibición en el proceso auditivo son de gran relevancia, los modelos auditivos proporcionan buenos resultados de reconocimiento frente al cepstrum LPC [29, 30]. Sin embargo, los modelos auditivos requieren una gran cantidad de tiempo de parametrización. Como ejemplos, en el modelo auditivo de Seneff se requiere un tiempo de computación unas 120 veces superior al de los coeficientes MFCC, y de 360 veces para el modelo EIH. Este hecho, añadido a que los coeficientes MFCC proporcionan resultados tan sólo ligeramente peores a los obtenidos con modelos auditivos en condiciones de ruido, ha provocado que los modelos auditivos de alta resolución no hayan sido incorporados en sistemas de reconocimiento de voz que deben operar en tiempo real. En la actualidad está muy difundido el uso de parametrizaciones derivadas de los coeficientes MFCC, que también incorporan características perceptuales.

2.2.4. El vector de características

Para cada una de las tramas, el bloque de representación construye un vector de características, de forma que la señal quede representada por una secuencia de vectores [31]. Además de los coeficientes cepstrales, que representan el tracto vocal, resulta deseable la inclusión de otras informaciones que permitan enriquecer la representación de la voz. Por esta razón, el vector de características suele incluir tres tipos de características:

1. *Información espectral*, expresada normalmente en el dominio del cepstrum.

2. *Energía del segmento*, que contribuye a la discriminación entre vocales, consonantes sonoras y consonantes sordas. Suele incorporarse en escala logarítmica a fin de evitar la variabilidad.

3. *Características dinámicas*, que tratan de representar la evolución temporal de los parámetros. Generalmente se incluye la primera y segunda derivada (velocidad y aceleración) de los parámetros anteriores.

La energía del segmento suele calcularse a partir de la señal $s(n)$ como,

$$E = 10 \log_{10} \left(\frac{1}{N} \sum_{n=0}^{N-1} s^2(n) \right) \tag{2.10}$$

Sin embargo, según la representación elegida, existen alternativas más eficientes para la obtención de la energía. Cuando se emplea la representación LPC, la energía del segmento se puede obtener a partir del coeficiente de autocorrelación de orden cero $R(0)$ de la forma:

$$E = 10 \log_{10}(R(0)) \tag{2.11}$$

Por otro lado, en la parametrización MFCC, la energía de la trama se puede obtener sumando la energía de los filtros:

$$E = 10 \log_{10} \left(\sum_{b=1}^{B} E_b \right) \tag{2.12}$$

donde E_b es la energía de salida del filtro b. Adicionalmente, en parametrizaciones MFCC se puede obtener como coeficiente para la energía, el coeficiente cepstral de orden cero [32], extendiendo la ecuación (2.9) para $k = 0$. Conviene destacar que en ese caso el coeficiente es la suma de los logaritmos de las energías, en vez del logaritmo de la suma de energías.

Los coeficientes cepstrales junto con la energía contienen información estática o *instantánea* de la voz. Para mejorar la representación de la voz, es posible incluir características dinámicas que informen sobre la evolución temporal de la señal, representando la evolución en el tiempo de los parámetros. Para ello, cada característica instantánea x_k es considerada una función del tiempo $x_k(t)$. En la aproximación de Furui [33], se considera un entorno de $2W$ tramas centrado en la trama correspondiente al instante de tiempo t, de forma que la característica $x_k(t)$ se pueda desarrollar en una serie de polinomios ortogonales. Considerando únicamente hasta el polinomio de primer grado, los coeficientes dinámicos Δx_k^0 se pueden obtener como,

$$\Delta x_k^0 = \frac{\sum_{w=-W}^{+W} w \cdot x_k(t+w)}{\sum_{w=-W}^{+W} w^2} \tag{2.13}$$

en donde Δx_k^0 se sustituye por $\Delta x_k(t)$ para indicar que el entorno esta centrado en el instante de tiempo t. Los coeficientes $\Delta x_k(t)$ pueden considerarse como la pendiente de la tangente a $x_k(t)$.

Otros trabajos [34] proponen el uso de coeficientes $\Delta x_k(t)$ más simples, obtenidos como la diferencia de los coeficientes estáticos entre intervalos cortos de tiempo,

$$\Delta x_k^0 = x_k(t+2) - x_k(t-2) \tag{2.14}$$

El cálculo de las características dinámicas, empleando cualquiera de las aproximaciones anteriores, puede interpretarse como la aplicación de una ventana deslizante. Los pesos de dicha ventana dependerán de la expresión (2.13 o 2.14) elegida.

Las características dinámicas se obtienen tanto para los coeficientes cepstrales, denominándose coeficientes *delta-cepstrales*, como también para las energía, conociéndose como coeficientes *delta-energía*. Adicionalmente, es posible incluir en el vector de características la segunda derivada de los coeficientes. Ésta se obtiene, o bien considerando hasta un polinomio de segundo grado en la aproximación de Furui, o bien aplicando cualquiera de las ecuaciones (2.13) o (2.14) sobre los coeficientes delta-ceptrales o la delta-energía. Se habla entonces de coeficientes *delta-delta-cepstrales* y *delta-delta-energía*, respectivamente.

2.3. Reconocimiento de voz mediante modelos ocultos de Markov

Los modelos ocultos de Markov (*HMM–Hidden Markov Models*) constituyen una de las herramientas más utilizadas en el reconocimiento del habla en las últimos años. Sus orígenes se remontan a principios de los años 60, en los que sus fundamentos fueron establecidos por Baum y sus colaboradores. Sin embargo, los primeros trabajos de aplicación de HMMs en RAH no aparecen hasta mediados de los 70, aplicándose de forma generalizada a partir de los años 80 [35, 36, 37, 38].

La teoría sobre HMMs se desarrolla como una generalización de los *Procesos de Markov*. Un proceso de Markov puede ser descrito mediante un conjunto de estados y un conjunto de probabilidades de transición desde un estado a otro. Así, los procesos de Markov se caracterizan por la dependencia que presenta el estado actual respecto a los anteriores, esto es, poseen "memoria". Los procesos de Markov son muy útiles a la hora de modelar procesos con memoria en los que los estados son directamente observables.

Sin embargo, en ciertos procesos, los estados no son directamente observables, sino que un mismo estado puede generar diferentes observaciones con distinta probabilidad. Este es el caso del proceso de producción de voz.

2.3.1. Formulación de los HMMs

Un HMM es un modelo estadístico que describe la producción de una secuencia de observaciones $\{o_1, o_2, ..., o_T\}$, generada por una secuencia "oculta" de estados $\{q_1, q_2, ..., q_T\}$. De esta forma, un HMM modela dos procesos estocásticos superpuestos, uno de ellos oculto, la secuencia de estados, y otro observable, la secuencia de observaciones. Un HMM se define mediante un conjunto de estados $\{s_1, s_2, ..., s_N\}$ interconectados entre si. Entre dichos estados se establecen probabilidades de transición por medio de una matriz $A = \{a_{ij}\}$ donde

$$a_{i,j} = P\left(q_{t+1} = s_j | q_t = s_i\right) \quad (i, j = 1, ..., N) \tag{2.15}$$

que verifican la siguiente condición de normalización,

$$\sum_{j=1}^{N} a_{i,j} = 1 \tag{2.16}$$

La probabilidad del estado inicial viene dada por una matriz, $\Pi = \{\pi_i\}$, de probabilidades a priori, donde:

$$\pi_i = P\left(q_1 = i\right) \quad (i = 1, ..., N) \tag{2.17}$$

con la condición de normalización,

$$\sum_{i=1}^{N} \pi_i = 1 \tag{2.18}$$

La distribución de probabilidad asociada a la generación de observaciones en cada estado depende de si el conjunto de observaciones es discreto o continuo, esto define, a su vez, el tipo de modelado. Si el conjunto de posibles observaciones es discreto, $V = \{v_1, v_2, ..., v_M\}$, se habla de modelos Discretos (*DHMM–Discrete HMM*) y la probabilidad

de producción de cada observación viene dada por:

$$b_i(v_k) = P(o_t = v_k | q_t = s_i) \qquad (2.19)$$
$$\sum_{k=1}^{M} b_i(v_k) = 1$$

Si el conjunto de observaciones es continuo, hablaremos de modelos continuos (*CHMM–Continuous HMM*) [37], sustituyendo la probabilidad de producción de observaciones por una función densidad de probabilidad (*pdf*):

$$b_i(x) = p(x|s_i, \lambda) = \sum_{v_k \in V(s_i,\lambda)} p(x|v_k, s_i, \lambda) P(v_k|s_i, \lambda) \qquad (2.20)$$
$$\int_{\mathbb{R}^p} b_i(x) d^p x = 1$$

en donde $x \in \mathbb{R}^p$ y cada función de probabilidad es etiquetada con un índice v_k que varía en un conjunto $V(s_i, \lambda)$ específico para el estado s_i del modelo λ. La pdf $p(x|v_k, s_i, \lambda)$ es logarítmicamente cóncava o elípticamente simétrica, muy frecuentemente una gaussiana multivariada, dado que existe un método de estimación adecuado. Generalmente la expresión (2.20) es una mezcla de gaussianas [39], donde los factores $p(x|v_k, s_i, \lambda)$ son los coeficientes de la mezcla (la suma extendida a $V(s_i, \lambda)$ debe ser la unidad).

El modelado DMMH supone que la secuencia de vectores, procedente del análisis de la señal de voz, llega al núcleo del sistema de reconocimiento como una secuencia de observaciones pertenecientes a un conjunto discreto. Esto implica que es necesario un paso previo de cuantización que irremediablemente supone una pérdida de información. Por esta razón, suele hacerse deseable el modelado continuo.

Sin embargo, el modelado CHMM también presenta algunos inconvenientes, el modelo de probabilidad de las observaciones puede ser incorrecto si se emplea un número insuficiente de gaussianas en la mezcla, además de ser más complejos y requerir más cómputo. Debido a esto, se han propuesto simplificaciones del modelo continuo que conducen a nuevas aproximaciones HMM. Entre estas simplificaciones destacan el modelado semicontinuo o *SCHMM* [40], el modelado con cuantización vectorial múltiple o *MVQHMM* [41, 42] y el modelado semicontinuo con cuantización vectorial múltiple o *SCMVQHMM* [43, 44].

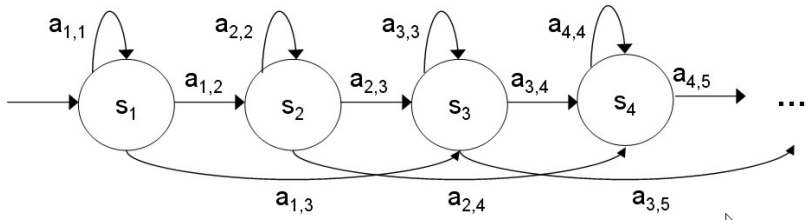


Figura 2.5: HMM con topología de izquierda a derecha con salto máximo de dos estados.

2.3.2. Topología de los HMMs en reconocimiento de voz

La topología más general para un modelo HMM corresponde al modelo *ergódico* que asegura la existencia de una transición desde un estado dado a otro cualquiera. Sin embargo, debido a la naturaleza secuencial de la señal de voz, la topología más extendida para el reconocimiento de voz es la *izquierda-derecha* o de *Bakis* [38, 45]. En esta topología los estados están ordenados y sólo se permite la transición desde un estado s_i a uno posterior $s_{i+\delta}$, donde δ puede tomar valores desde 0 hasta un valor de salto máximo Δ. La figura 2.5 muestra un ejemplo de esta topología con $\Delta = 2$. En la práctica, esta topología se puede obtener anulando las probabilidades de transición a_{ij} tales que $j \neq i + \delta$ $(0 \geq \delta \geq \Delta)$.

2.3.3. Modelado de voz con HMMs

Como vimos en la sección 2.1.3, en la aproximación estadística al RAH es necesario considerar dos modelos de producción, el modelo del lenguaje y el modelo acústico. Los HMMs son empleados generalmente en el modelado acústico, asociando a cada unidad de reconocimiento λ_l, su propio HMM.

El reconocimiento de palabras aisladas se puede llevar a cabo mediante un procedimiento como el mostrado en la figura 2.6. En éste se asocia un modelo HMM por cada palabra, de forma que el modelo que arroje la mayor probabilidad indica la palabra reconocida $P(X|\lambda_l)$. Aunque en el reconocimiento de palabras aisladas, con un vocabulario pequeño, es viable esta asociación, para el caso de reconocimiento del habla continua, resulta impracticable. Esto se debe no sólo a los enormes requerimientos de memoria y procesamiento, sino también a la ingente cantidad de secuencias de entrenamiento.

Por esta razón, el reconocimiento de habla continua se aborda de forma distinta. Primero se define un *macromodelo* que incluye todas las posibles frases permitidas por la gramática. En este macromodelo, las palabras están asociadas a estados, mientras que

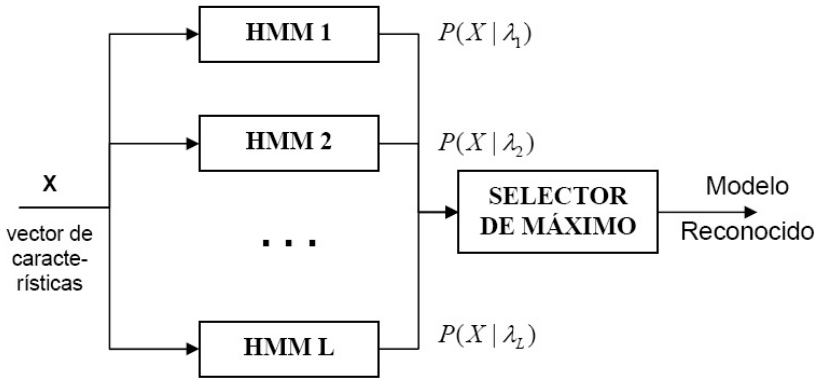


Figura 2.6: Sistema de reconocimiento de palabras aisladas basado en modelos HMM.

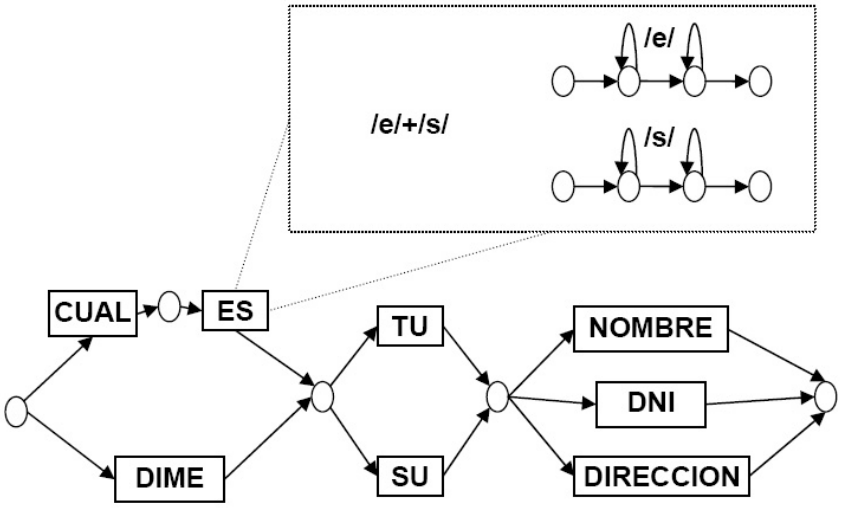


Figura 2.7: Sistema de reconocimiento de voz continua basado en modelos HMM.

las probabilidades de transición entre ellos vienen dadas por la gramática. Alternativamente, si se emplean unidades inferiores a la palabra como fonemas, silabas o trifonemas, se pueden formar palabras concatenando los modelos de estas unidades, y frases a partir de macromodelos que concatenen palabras. La figura 2.7 muestra un ejemplo de un macromodelo. En este caso, no importa tanto evaluar la probabilidad de que el modelo HMM genere la observación, sino conocer qué camino (esto es, secuencia de estados, fonemas, palabras, ...) maximiza la probabilidad $P(Q|X, \lambda)$ en todo el macromodelo para una observación dada.

El reconocimiento del habla por medio de modelos HMMs requiere, por tanto, la resolución de los siguientes problemas [4, 46]:

1. Problema de *evaluación*, que consiste en determinar, dada la observación acústica X y el modelo λ, la probabilidad de que el modelo genere esa observación, es decir, la probabilidad acustica $p(X|\lambda)$.

2. Problema de *decodificación*. Dado el modelo λ, consiste en hallar la secuencia de estados $Q = q_1, q_2, q_3, \ldots, q_T$ más probable o camino óptimo dada la observación acústica X,
$$Q^* = arg \max_Q P(Q|X, \lambda) \tag{2.21}$$

3. Problema de *estimación*, que consiste en hallar el conjunto de parámetros de cada HMM que mejor se ajusta a un conjunto de observaciones acústicas. Estos parámetros son los descritos en la ecuación (2.15), (2.17), y ecuaciones (2.19) o (2.20), dependiendo de si se usa un modelado discreto o continuo, respectivamente. Cuando se utiliza la topología izquierda a derecha, normalmente se impone que s_1 sea el primer estado del camino, por lo que no es necesaria la estimación de la matriz Π (ecuación (2.17)).

El primer problema es el problema más básico de reconocimiento. Éste se plantea en los sistemas de reconocimiento de palabras aisladas y para resolverlo de forma eficiente se utiliza el algoritmo adelante-atrás (o *algoritmo forward-backward*) [38, 47]. El segundo problema aborda la recuperación de la parte oculta del proceso, es decir, la secuencia de estados. Tiene especial interés en el reconocimiento continuo del habla, donde se debe encontrar el camino óptimo sobre el macromodelo. Para resolverlo se recurre al *algoritmo de Viterbi* [48]. Finalmente, el tercer problema trata el entrenamiento de los modelos HMMs, esto es, como obtener los parámetros que los definen únicamente a partir de observaciones acústicas y sus transcripciones. Este problema se resuelve a través del algoritmo de *Baum-Welch* [47, 49].

2.4. Criterios de evaluación

En este trabajo se proponen y analizan distintas técnicas para mejorar el reconocimiento remoto de voz en canales adversos. Como en muchas otras situaciones, la elección de una técnica u otra en el diseño de un sistema RAH requiere del uso de alguna métrica que permita comparar el rendimiento del sistema de reconocimiento con cada una de ellas. El rendimiento de un sistema de reconocimiento se debe evaluar considerando: 1) la probabilidad de cometer errores de reconocimiento, 2) los requerimientos computacionales,

como la complejidad de los algoritmos implicados y los requerimientos de memoria y 3) el tiempo de respuesta. La opción más apropiada será aquella que reduzca la probabilidad de error manteniendo los requerimientos computacionales y el tiempo de respuesta bajo unos limites tolerables.

2.4.1. Tasa de error y precisión en el reconocimiento

La precisión del reconocedor se evalúa por medio de pruebas de reconocimiento en las que se cuentan el número de errores producidos. El cociente entre el número de errores y el número de elementos reconocidos constituye la *tasa de error* y representa la probabilidad de cometer errores de reconocimiento.

La tasa de error se obtiene de distinta forma según se reconozcan palabras aisladas o voz continua. En tareas de reconocimiento de palabras aisladas, se define la *tasa de error de palabra* (*WER-Word Error Rate*), obtenida como,

$$WER = \frac{n_e}{n_t} \tag{2.22}$$

donde n_e es el número de errores o palabras clasificadas incorrectamente y n_t es el número total de palabras del test.

En tareas de voz continua, el reconocimiento se realiza frase a frase, en donde pueden aparecer errores de reconocimiento de tres tipos: 1) *inserciones*, o palabras adicionales insertadas en la frase reconocida, 2) *sustituciones*, o palabras sustituidas por otras palabras, y 3) *borrados*, o palabras que no aparecen en la frase reconocida. Para contabilizar el número de inserciones, sustituciones y borrados, la frase reconocida y su transcripción correcta son alineadas mediante un procedimiento de alineamiento de cadenas basado en DTW. El WER en tareas de voz continua se obtiene entonces como,

$$WER = \frac{n_i + n_s + n_d}{n_t} \tag{2.23}$$

donde n_i, n_s y n_d son, respectivamente, el número de inserciones, sustituciones y borrados. Igualmente, el rendimiento de un reconocedor puede expresarse en términos de la tasa de acierto o precisión del reconocimiento de palabra (*WAcc–Word Accuracy*),

$$W_{Acc} = 1 - WER = \frac{n_t - (n_i + n_s + n_d)}{n_t} \tag{2.24}$$

Tanto el WER como el WAcc se ofrecen en porcentajes, pudiéndose obtener tasas de error superiores al 100 % o, equivalentemente, WAccs negativos, debido a las inserciones.

2.4.2. Medidas de Confianza

La finalidad de las pruebas de reconocimiento es estimar la *probabilidad de acierto* (p), o alternativamente, *la probabilidad de error* ($1-p$), de un sistema RAH. Esta medida permite comparar sistemas de reconocimiento a fin de elegir el mejor de ellos. La tasa de acierto obtenida en una prueba de reconocimiento es una estimación de la probabilidad de acierto, pero no la probabilidad de acierto. Esto implica que las tasas de acierto deben interpretarse cuidadosamente, ya que las mejoras obtenidas durante las pruebas pueden no resultar estadísticamente significativas.

El valor exacto de la probabilidad de acierto no se puede conocer a través del test de reconocimiento. Sin embargo, sí se puede definir un intervalo de confianza en torno a la tasa de acierto de forma que podamos establecer cómo de fiables son las conclusiones que extraigamos. Este intervalo se denomina *intervalo de confianza* y se define con respecto a un cierto porcentaje, por ejemplo el 95 %, para una determinada tasa de acierto $\hat{p}$. Si se asume que el número de elementos correctamente reconocidos es una variable aleatoria de distribución binomial $B(n,p)$, caracterizada por la probabilidad de acierto p, siendo n el número total de ensayos, entonces podemos definir un intervalo de confianza centrado en $\hat{p}$ (la tasa de acierto), que contiene, con probabilidad $(1-\alpha)$, la probabilidad de acierto. Mediante el teorema del límite central, se puede demostrar que la distribución de probabilidad tiende a la distribución normal $N(0,1)$, de forma que el intervalo de confianza puede obtenerse como,

$$\left[\hat{p} - z_{1-\frac{\alpha}{2}}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}, \hat{p} + z_{1-\frac{\alpha}{2}}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}\right] \tag{2.25}$$

donde $z_{1-\alpha/2}$ es el cuantil $1-\alpha/2$ de la distribución normal estándar (la tabla 2.1 muestra algunos de ellos). Como puede observarse, cuanto mayor es el número de ensayos, esto es, el número de elementos reconocidos, más estrecho es el intervalo de confianza, es decir, se conoce con mayor precisión la probabilidad de acierto del sistema de reconocimiento.

2.4.3. Aspectos computacionales y tiempo de respuesta

En aplicaciones prácticas, los requerimientos de computación de los sistemas de reconocimiento son de gran importancia. Aunque los computadores evolucionan constantemente,

Intervalo de confianza	$z_{1-\alpha/2}$
90.00 %	1.65
95.00 %	1.96
95.44 %	2.00
99.00 %	2.57
99.74 %	3.00
99.90 %	3.32

Tabla 2.1: Cuantiles para una distrubición normal estándar.

siendo cada día más potentes y baratos, y menores las limitaciones computacionales, el reconocimiento del habla, especialmente, el de voz continua, requiere tal cantidad de cómputo que resulta inabarcable si no se realizan aproximaciones.

Un ejemplo de esto era descrito anteriormente cuando se comentaban las técnicas incluidas en el bloque de representación. Aquellas basadas en modelos auditivos no eran generalmente empleadas debido a su alta carga computacional, prefieriéndose las basadas en banco de filtros que, aunque ofrecían resultados ligeramente inferiores, presentaban una carga computacional significativamente inferior.

El proceso de reconocimiento de voz continua es considerablemente más complejo que el caso de palabras aisladas. Para tener una idea de la ingente cantidad de operaciones requeridas, simplemente el algoritmo de Viterbi (sin tener en cuenta el cálculo de probabilidades para cada estado) requiere, por cada iteración, M^2 sumas, una por cada transición posible, y M comparaciones entre los M^2 resultados. El uso de topologías izquierda-derecha, con un salto máximo permitido Δ pequeño, puede reducir el número de transiciones a calcular, agilizando el reconocimiento de palabras aisladas. Sin embargo, en el reconocimiento de voz continua, si la gramática permite un alto número de frases (lo cual es deseable) el número de transiciones se dispara.

Para habilitar el reconocimiento, se procede a una *estrategia de poda* en donde el espacio de búsqueda se reduce en cada iteración mediante alguna heurística. Por ejemplo, ciertos caminos que tengan acumulada una probabilidad inferior a cierto umbral, se eliminan bajo la suposición de que es improbable que lleguen a ser óptimos. Esta estrategia puede conducir a soluciones subóptimas si el umbral es demasiado restrictivo, aunque cuanto más restrictivo sea el umbral, mayores son las mejoras computacionales que se obtienen. Así, la elección del umbral de poda supone un compromiso entre la tasa de error y los requerimientos computaciones.

Estos altos requerimientos suponen un problema para los usuarios domésticos y limitan

el uso generalizado del RAH. Por esta razón, como veremos más adelante, el reconocimiento remoto de la voz se alza como una prometedora solución que no sólo generalizaría el uso del RAH, sino que, dado el amplio despliegue de comunicaciones existente, lo habilitaría prácticamente en cualquier lugar.

El tiempo de respuesta suele considerarse como el lapso de tiempo desde que se termina de pronunciar una frase y se obtiene una respuesta (aunque en ciertas aplicaciones, como en los dictáfonos, puede ser deseable que se produzca una respuesta por cada palabra). Esto implica no sólo a la parte de decodificación sino a todo el proceso de comunicación con la máquina. El tiempo de respuesta es relevante debido a que comunicación hombre-máquina es poco confortable si existe una latencia superior a 500ms [50]. En los sistemas de reconocimiento remoto de la voz, el tiempo de respuesta se ve afectado no sólo por la complejidad del reconocedor, sino también por el canal de comunicaciones, como veremos en posteriores capítulos.

2.5. Descripción del sistema RAH de referencia

En esta sección se describe el sistema de reconocimiento del habla que se empleará para evaluar las técnicas presentadas a lo largo de este trabajo. Este sistema de reconocimiento está basado en el marco experimental propuesto por el grupo de trabajo de la ETSI STQ-AURORA DSR Working Group [51], válido para la evaluación de sistemas de reconocimiento remoto en general.

La tarea de reconocimiento propuesta consiste en el reconocimiento de voz continua de cadenas de dígitos sin limitación en la longitud de éstas. La lengua reconocida es la inglesa con acento americano. El vocabulario empleado consta de 11 palabras, una para cada dígito excepto el cero, que admite dos pronunciaciones ("zero" y "o"). A continuación describiremos el proceso de extracción de características, el reconocedor y la base de datos empleados en este trabajo.

Extracción de Características

La extracción de características está basada en la representación mediante bancos de filtros. La señal de voz es muestreada a 8kHz, sobre ésta se aplica una compensación de la componente continua además de un preénfasis con un factor $\mu = 0{,}97$. Posteriormente, la señal es segmentada en tramas de 25ms tomadas cada 10ms, usando una ventana de Hamming [23] que supone un buen compromiso entre la resolución espectral y el rizado

lateral. Una DFT de longitud 256 se aplica a cada ventana para calcular la magnitud del espectro de la señal. Para ello se extiende con ceros cada trama hasta las 256 muestras.

Aplicando a la DFT un banco de filtros triangulares distribuidos uniformemente en escala mel desde 64Hz hasta 4kHz, se obtienen 23 MFSCs. Usando logaritmos naturales y mediante una DCT se obtienen los 12 coeficientes cepstrales en escala mel (desde el de orden 1 al de orden 12).

El vector de características con información instantánea de la voz consta de los 12 MFCCs anteriores más el logaritmo de la energía de la señal (obtenido tras eliminar la componente continua y aplicar el filtro de pre-enfasis). Este vector se amplia mediante coeficientes dinámicos (delta y delta-delta-cepstrales, y delta y delta-delta-energía) hasta alcanzar 39 componentes.

Reconocedor de Voz

El reconocedor de voz está basado en modelos ocultos de Markov. Para su construcción se ha empleado el paquete de software HTK versión 3.3 de Entropic. Este conjunto de herramientas fue ideado para la construcción y manipulación de HMMs, empleándose principalmente para el reconocimiento de voz.

Cada dígito es modelado como una palabra completa por un HMM continuo, con los siguientes parámetros:

- Se establecen 16 estados por palabra (18 estados si se consideran los dos nodos de enlace al principio y final de cada HMM).
- La topología empleada es la de izquierda a derecha con un salto máximo permitido de un estado ($\Delta = 1$).
- Cada estado cuenta con una mezcla de 3 gaussinas multivaluadas con 39 componentes.
- En las gaussianas multivaluadas no se consideran matrices de covarianza completas, sino sólo la diagonal principal (varianzas).

Adicionalmente, se definen dos modelos de pausa, uno para el comienzo y final de la frase y otro para los silencios entre las palabras. El primero consta de 3 estados con una mezcla de 6 gaussianas por estado. El segundo consiste simplemente en un estado ligado (compartiendo sus parámetros) al estado intermedio del modelo de silencio para comienzo y fin de frase. El entrenamiento se realiza aplicando el algoritmo de reestimación de Baum-Welch en varias iteraciones.

Base de datos

En este trabajo se ha adoptado un subconjunto de la base de datos del proyecto AURORA (ETSI STQ-AURORA Proyect Database 2.0 [52]). Esta base de datos es una versión revisada de la base de datos Noisy TIDigits construida, tal y como se indica en su documentación, aplicando las siguientes operaciones:

- Submuestreo. La base de datos original TIDigits está muestreada a 20kHz, por lo que fue submuestreada a la frecuencia usual empleada comunicaciones por voz, esto es, a 8kHz. El filtro de submuestreo empleado fue un filtro paso-bajo ideal.

- Filtrado. Un filtrado adicional fue aplicado para considerar de forma realista las características en frecuencia de los terminales y equipos en el área de comunicaciones. Este filtrado adicional se realizó con el estándar G.712 propuesto por la ITU [53].

- Adición de ruido acústico. A fin de evaluar las prestaciones en condiciones de ruido de fondo, además del conjunto en condiciones acústicas favorables se crearon, mediante la adición controlada de ruido, varios conjuntos de condiciones acústicas desfavorables incluyendo 8 condiciones distintas de ruido una relación señal ruido (SNR) de 20, 15, 10, 5, 0 y -5dB.

La base de datos consta de secuencias de dígitos conectados en inglés pronunciados por oradores americanos. Para la fase de entrenamiento se dispone de dos conjuntos diferenciados para entrenamiento sobre voz limpia, y para entrenamiento sobre voz limpia y ruidosa, denominados, *clean* y *multi-condition*, respectivamente. El primero consta de 8440 frases conteniendo un total de 55 voces masculinas y 55 femeninas, todas de oradores adultos, sobre las que se ha aplicado el filtrado G.712 pero no ha sido añadido ningún ruido acústico. El segundo conjunto consta de las mismas frases pero organizadas en 20 subconjuntos con distintas condiciones de ruido acústico (4 tipos de ruido con 5 SNRs).

Originalmente se definen 3 conjuntos de prueba en esta base de datos. El primer conjunto denominado *set A* contiene ruidos utilizados en las secuencias de entrenamiento multicondición, el segundo conjunto, *set B*, contiene ruidos que no han sido vistos durante la fase de entrenamiento, y el tercer conjunto (*set C*) emplea un filtrado M-IRS junto con adición de ruido que permiten testear la combinación de distorsión y ruido convolucional.

Ya que el tratamiento del ruido acústico queda fuera del interés de este trabajo, únicamente se han empleado los subconjuntos que incluyen voz limpia, sin ruido acústico. Esto es, para la fase de entrenamiento se ha empleado el conjunto "clean", con sólo voz

limpia, mientras que para la fase de entrenamiento se ha empleado el subconjunto del "set A" que incluye igualmente sólo voz limpia. Este subconjunto consta de 4004 frases de 52 locutores masculinos y 52 femeninos, divididos en 4 subconjuntos cada uno con 1001 frases.

El número total de elementos a reconocer durante las pruebas, que alcanza las 13159 palabras (realizaciones de éstas), asegura cierta significatividad estadística de los resultados obtenidos. La figura 2.8 muestra la amplitud de los intervalos de confianza al 90 %, 95 % y 99 % ofrecidos por el sistema de referencia para distintos porcentajes de precisión en el reconocimiento. Estos porcentajes van desde el 100 % al 60 %, ya que ninguna de las pruebas que se realizarán a lo largo de este trabajo obtienen precisiones de reconocimiento inferiores a este procentaje. Como puede observarse, gracias al elevado número de palabras empleadas en las pruebas, se obtiene intervalos de confianza reducidos. De forma grosera, puede afirmarse que, suponiendo un intervalo de confianza del 95 %, mejoras superiores al cuarto de punto resultan signficativas para precisiones en torno al 95 %, de medio punto para precisiones superiores al 90 %, de tres cuartos de punto para precisiones superiores al 75 % y de un punto para el resto. Debido a la sobrecarga de las tablas que ello supondría, se evitará ofrecer los intervalos de confianza junto con la precisión en el reconocimiento, recomendándose la consulta de esta gráfica para más detalles.

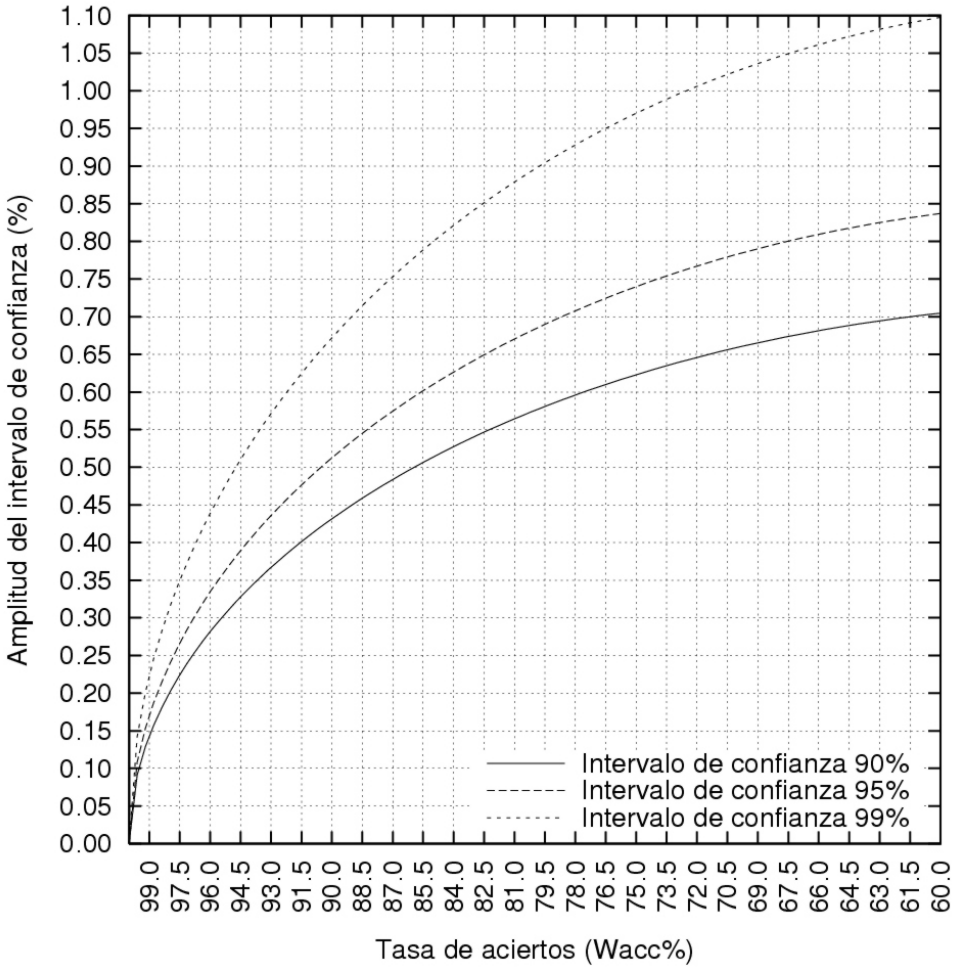


Figura 2.8: Amplitud del intervalo de confianza al 90 %, 95 % y 99 % según el porcentaje de precisión en el reconocimiento para la base de datos de referencia.

Capítulo 3

RECONOCIMIENTO REMOTO DE LA VOZ EN CANALES DIGITALES

3.1. Introducción

El acceso a la información en cualquier momento y en cualquier lugar se ha convertido en algo no tan sólo deseable sino casi necesario. Actualmente, un importante esfuerzo de investigación e ingeniería se está invirtiendo en este propósito. Continuamente están apareciendo nuevos dispositivos portátiles destinados a dicho acceso (como PDAs, teléfonos móviles, etc.) con más funcionalidades, una mayor autonomía y cada vez más pequeños. Sin embargo, uno de los problemas derivados esta miniaturización es la presencia de teclados y pantallas de reducidas dimensiones que dificultan la interacción con los dispositivos. Se hace patente la necesidad de interfaces de usuario mejoradas, y el acceso a la información por medio de algo tan cotidiano como la voz se plantea como solución.

Desafortunadamente, el reconocimiento de voz en un dispositivo móvil, aún siendo muy deseable por la ingente cantidad de servicios que posibilitaría, presenta serias dificultades. Actualmente la complejidad del reconocimiento con un vocabulario mediano o grande está más allá de las limitaciones de memoria, recursos computacionales y consumo de la mayoría de dispositivos portátiles. Esta dificultad a impulsado la aparición de un nuevo paradigma en el RAH denominado Reconocimiento Remoto de la Voz (*RSR–Remote Speech Recognition*). En él, el reconocimiento no se realiza en el propio dispositivo, sino en un servidor ubicado en otro lugar, cuyos recursos además pueden compartirse entre

Figura 3.1: Estructura general del reconocimiento remoto del habla.

múltiples usuarios. De esta forma, se eliminan las restricciones de tamaño y consumo, habilitando el uso de grandes y potentes ordenadores que contengan sistemas de reconocimiento más sofisticados y elaborados con sistemas de dialogo, módulos de lenguaje natural, síntesis de voz y acceso a bases de datos. En la figura 3.1 se muestra la estructura general de un sistema RSR. En ella, varios tipos de clientes, como PCs multimedia, PDAs y teléfonos móviles actúan como dispositivos activados por voz. En el otro extremo, las tareas de reconocimiento de los múltiples usuarios son gestionadas por un servidor de aplicaciones que controla un cluster compartido de servidores de reconocimiento [54]. Se distingue por tanto entre un cliente que requiere servicios de reconocimiento de voz y un servidor que los realiza, ambos conectados mediante alguna red de comunicaciones.

Por su alcance prácticamente local, dos redes de comunicación digitales se posicionan como candidatas a esta tarea, las redes móviles GSM y las redes de paquetes basadas en IP. Mediante una sencilla llamada telefónica, las redes GSM posibilitan la transmisión de la voz desde casi cualquier localización (en el mundo desarrollado), gracias a su naturaleza inalámbrica y enorme despliegue geográfico. Por su parte, las redes IP están logrando imponer su presencia en detrimento de cualquier otro tipo de red fija gracias, sin duda, a su capacidad (aún limitada) para transmitir cualquier tipo de información multimedia. De hecho, la amplia aceptación de las redes IP, así como el acceso masivo a la Red de redes, ha forzado un creciente interés por las posibilidades de interconexión de ambos tipos de redes, en una convergencia hacia una única red *"todo IP"*.

Sin embargo, este nuevo paradigma no está exento de problemas. La voz debe ser transmitida desde la ubicación del emisor hasta el reconocedor remoto. Esto no es una ta-

rea trivial ya que ahora la voz, o al menos la información necesaria para el reconocimiento contenida en ella, debe atravesar un canal que es propenso a errores. Como veremos, en GSM se dispone de un canal de radio altamente expuesto al ruido, mientras que en las redes IP dispondremos de un canal basado en conmutación de paquetes que, al no estar diseñado para la transmisión de datos en tiempo real, será propenso a la pérdida de paquetes. En este capítulo revisaremos, atendiendo principalmente a aquellos aspectos que resulten relevantes para el reconocimiento remoto, ambas redes. También, describiremos las arquitecturas propuestas para habilitar el reconocimiento remoto, así como las soluciones disponibles actualmente. Por último, dedicaremos las últimas secciones a evaluar y analizar la influencia que los problemas de estos canales tienen sobre la precisión en el reconocimiento.

3.2. Redes GSM

La radio móvil ha sido usada durante al menos durante 75 años. Aunque los conceptos de estructura celular, técnicas de dispersión del espectro, modulación digital y otras tecnologías empleadas en las comunicaciones móviles actuales eran conocidos desde hace más de 50 años, los servicios de telefonía móvil no aparecieron hasta 1960. Estos primeros sistemas estaban muy limitados y su capacidad era ínfima en comparación a los estándares actuales. Los sistemas celulares analógicos aparecerán más tarde, en la década de los 80. Tras su éxito y crecimiento comenzaron los problemas de escalamiento, no pudiéndose cubrir la demanda de servicios. Además, los diferentes sistemas eran incompatibles entre sí, era prácticamente imposible reducir el precio del terminal móvil y la calidad de comunicación era bastante ajustada. A principios de los 80 se hizo obvia la necesidad de un nuevo sistema de telefonía celular.

Sin embargo, el diseño de un nuevo sistema de telefonía celular requiere tal cantidad de esfuerzo e investigación que ningún país europeo podía afrontarlo de forma individual. Por esta razón se apostó por un diseño común hecho entre varios países. El CEPT (*"Conférence Européene des Postes et Télécommunications"*), una organización para la estandarización presente en más de 20 países europeos, creó en 1982 un nuevo cuerpo de estandarización cuya tarea era especificar un único sistema de radiotelecomunicaciones para Europa en la banda de los 900 MHz. El recién nacido Groupe Spécial Mobile (GSM) tuvo su primer encuentro en Diciembre de 1982 en Estocolmo, bajo la presidencia de Thomas Haug de la administración sueca. Treinta y una personas de once países estuvieron presentes en este primer encuentro. En 1990, por requerimiento del Reino Unido, se

añadió al grupo de estandarización la especificación de una versión de GSM a la banda de frecuencia de 1800 ± 75 MHz. A esta variante se le llamó DCS1800 (*Digital Cellular System 1800*). El significado actual de las siglas GSM se ha cambiado y en la actualidad se hacen corresponder con "Global System for Mobile Communications". La elaboración del estándar GSM necesitó casi una década.

Hoy en día, el estándar GSM no sólo ha sido adoptado en toda Europa sino también en diferentes países de prácticamente todos los continentes, lo que permite hablar de un verdadero sistema celular global, en competencia con los sistemas norteamericanos NADS (D-AMPS) y CDMA (N-CDMA).

3.2.1. Estructura Celular

El problema más grave de las comunicaciones móviles es la disponibilidad de espacio en el espectro para realizar la transmisión. El concepto de sistema celular [55] supuso un gran avance en la resolución del problema de la congestión espectral y de la capacidad de usuarios. Éste ofrece una gran capacidad en una localización limitada del espectro sin grandes cambios tecnológicos. La idea de un sistema celular consiste en un sistema basado en celdas, cada una de ellas proporcionando cobertura a sólo una pequeña porción del área de servicio. A cada estación base se le asigna una porción del número total de canales disponibles en el sistema completo, y a las estaciones base cercanas se les asignan diferentes grupos de canales de forma que las interferencias entre las estaciones base (y entre los usuarios móviles bajo su control) se reduzcan. Distribuyendo de forma sistemática las estaciones base y sus grupos de canales, los canales disponibles se distribuyen a través de una región y pueden ser reutilizados tantas veces como sea necesario, siempre que la interferencia entre estaciones con el mismo canal se mantenga por debajo de unos niveles aceptables. Conforme crece la demanda de servicios, se puede incrementar el número de estaciones base, proporcionando una capacidad de radio adicional sin incrementar las necesidades espectrales. Este principio es el fundamento de todos los sistemas modernos de comunicaciones inalámbricos, y en particular de GSM.

La Figura 3.2 ilustra el concepto de reutilización de frecuencias, en donde las celdas con el mismo número utilizan el mismo grupo de canales. En este ejemplo se emplea un patrón de 7 de celdas que se distribuye sistemáticamente por toda la región haciendo un uso eficiente del espectro de frecuencias. La forma hexagonal de la celda mostrada en la figura es conceptual y es un modelo simple de la cobertura de radio para cada estación base que ha sido universalmente adoptado, dado que el hexágono permite un análisis fácil

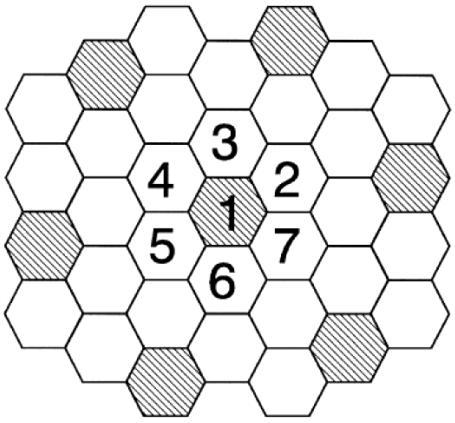


Figura 3.2: Ejemplo de un esquema de reutilización de grupos de canales en una estructura celular. Las celdas sombreadas emplean el grupo 1. Las celdas con diferentes números utilizan grupos diferentes.

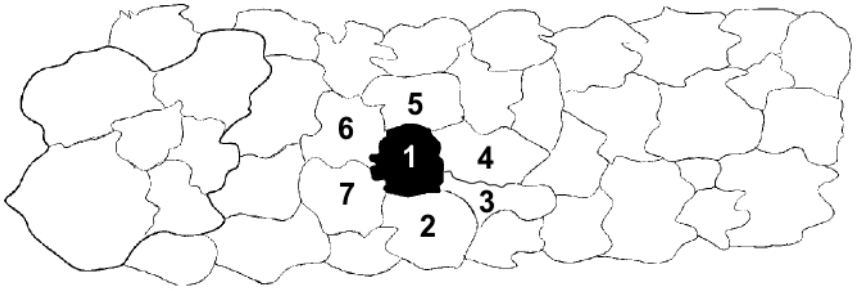


Figura 3.3: Huellas o "footprint" de las celdas en un sistema celular.

y manejable de un sistema celular. La cobertura real de una celda se conoce como *huella* (*footprint*) y se determina a partir de los modelos de campo o de los modelos de predicción de la propagación [56]. La figura 3.3 muestra una posible configuración de huellas en el espacio.

Para la utilización eficiente del espectro de radio, se requiere un sistema de reutilización de frecuencias que aumente la capacidad y minimice las interferencias. Se han desarrollado una gran variedad de estrategias de asignación de canales para llevar a cabo estos objetivos. Las estrategias de asignación de canales se pueden clasificar en fijas o dinámicas, definiendo las características del sistema. En una estrategia de asignación de canales fija, a cada celda se le asigna un conjunto predeterminado de canales. Cualquier llamada producida dentro de la celda, sólo puede ser servida por los canales no utilizados dentro de esa celda en particular. Si todos los canales de esa celda están ocupados, la llamada se bloquea y el usuario no recibe servicio, aunque en otra celda se dispongan

de canales libres. En una estrategia de asignación de canales dinámica, los canales no se asignan de forma permanente a las diferentes celdas. En su lugar, cada vez que se produce un requerimiento de llamada, la estación base servidora realiza una petición de un canal. En base a algún algoritmo de gestión que tiene en cuenta diversos factores como la frecuencia del canal a pasar, su distancia de reutilización, y otras funciones de coste, se asigna el canal requerido. Aunque las estrategias de asignación dinámicas aumentan las prestaciones del sistema, requieren una gran cantidad de cómputo en tiempo real.

Cuando una zona, debido al incremento de la demanda de servicios, se satura, se aplica un proceso de división de la celda o *splitting*. El splitting es una estrategia empleada para subdividir una celda congestionada en celdas más pequeñas, cada una con su propia estación base, a la que se le reduce la altura de la antena y la potencia de transmisión. Definiendo nuevas celdas que tengan un radio más pequeño que las celdas originales se incrementa la capacidad debido al incremento de canales por unidad de área. Así, el splitting incrementa la capacidad de un sistema celular al aumentar el número de veces que se reutilizan los canales, solucionando los problemas de escalamiento.

3.2.2. Arquitectura GSM

La infraestructura básica de un sistema GSM no difiere en mucho de la estructura de cualquier red celular. Cada celda tiene una estación base que opera con un conjunto de canales diferente de los utilizados por las células adyacentes. Las operaciones de control, como la gestión de canales, de un determinado conjunto de estaciones base son realizadas por estaciones de control. Una zona extensa, que puede constar de varias estaciones de control, es gestionada por un centro de conmutación que enruta llamadas entre la propia red, y hacia y desde redes externas públicas o privadas.

La figura 3.4 muestra un gráfico simplificado de los elementos que componen la arquitectura GSM. Estos elementos son la estación móvil (o terminal móvil), la estación base, el controlador de estación base y el subsistema de red. A continuación será descrito cada uno de ellos.

Estación Móvil

La estación móvil (*MS–Mobile Station*) representa normalmente la única parte del sistema GSM que el usuario ve. Existen estaciones móviles de muchos tipos como las montadas en coche, y los equipos portátiles, pero quizás las más famosas sean los terminales de mano. Una estación móvil además de permitir, mediante funciones de procesado de señal

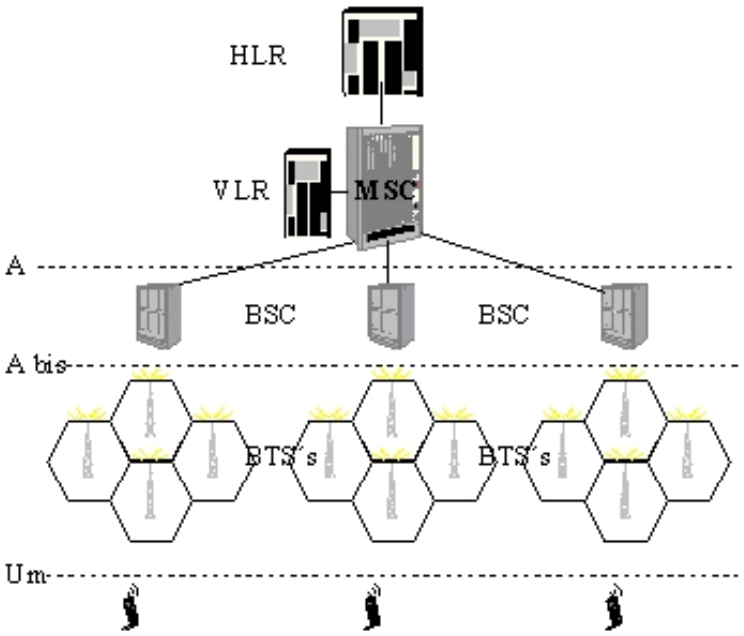


Figura 3.4: Esquema simplificado de la arquitectura GSM.

y de radiofrecuencia, el acceso a la red a través de la interfaz de radio, debe ofrecer también una interfaz al usuario (un micrófono, altavoz, display y tarjeta) para la gestión de las llamadas de voz, y puede incluir una interfaz para otro tipo de equipos (ordenador personal, fax, etc.).

Un elemento que se encuentra dentro de la estación móvil es el Módulo de Identificación del Abonado (*SIM–Subscriber Identity Module*), que es un nombre muy restrictivo para las diversas funciones que éste permite. El SIM se trata básicamente de una tarjeta con capacidad de procesamiento y memoria, su función es la de proveer al terminal móvil de una identidad dentro de la red. Además de ésto, la tarjeta puede contener una agenda con los números del usuario, espacio para almacenar mensajes cortos y otro tipo de información. Así, es la tarjeta SIM la que personaliza el móvil, pudiéndose emplear en cualquier terminal. Sin ella, el terminal móvil sólo puede realizar llamadas de emergencia. Debido a la sensibilidad de la SIM, ésta incorpora un código de seguridad de cuatro dígitos denominado PIN (*Personal Identification Number*).

Estación Base

La función principal de una estación base (*BTS–Base Transceiver Station*) es la de proporcionar un número de canales de radio a la zona a la que da servicio. La BTS dispone de los dispositivos de transmisión y recepción por radio, incluyendo las antenas, y también de los equipos encargados de todo el procesado de señal específico de la interfaz de radio (conocido como *Um*). Generalmente, las BTS se encuentran en el centro de la celda, determinando, según su potencia, el tamaño de ésta. Las BTS se pueden considerar como complejos modems de radio, con algunas otras funciones. Las BTS iniciales eran capaces de mantener simultáneamente 3 ó 5 portadoras de radio, permitiendo entre 20 y 40 comunicaciones simultáneas. Actualmente el volumen de los BTS se ha reducido mucho, esperándose un gran avance en este campo dentro de GSM.

Controlador de Estación Base

Los controladores de estación base (*BSC–Base Station Controller*) están conectados por un lado a varias estaciones base y por otro al centro de conmutación de servicios móviles (*MSC–Mobile Services Switching Center*). La función primaria de una BSC es la del mantenimiento de la llamada, así como la adaptación de la velocidad del enlace radio al estándar de 64 Kbits utilizado por la red (denominada transcodificación, sobre la que incidiremos más adelante). La BSC controla a su vez la potencia de trabajo de la estación móvil para minimizar la interferencia producida a otros usuarios y aumentar la duración de la batería. De esta forma, el BSC se encarga de toda la gestión de la interfaz de radio a través de comandos remotos sobre la BTS y la MS, principalmente referentes a la gestión de los canales de tráfico entre las celdas y de la conmutación de usuarios entre ellas. En GSM, cada estación móvil está continuamente comprobando la potencia de la señal con la que recibe no sólo a la BTS de la celda en la que se encuentra, sino también las de las celdas adyacentes. Mediante un informe acerca de estas medidas la MS comunica a la BSC la energía y calidad de la señal que recibe de la celda en la que está registrada. Esto permite a la BSC tomar la decisión de cuando iniciar una conmutación del usuario entre celdas, también conocido como *handover*. Durante este procedimiento, de forma totalmente transparente y sin cortes, el usuario es desconectado de la celda actual y conectado a otra que le ofrezca mejor calidad de servicio. Esto permite una verdadera movilidad para el usuario, que puede desplazarse libremente a través del conjunto de celdas. Una de las ventajas de GSM es que proporciona unos tiempos de conmutación mucho más bajos que otros sistemas celulares.

El hardware que requiere el BSC puede localizarse en el mismo sitio que una de las BTS que gestiona o en una localización aparte. Al conjunto formado por una BSC y las BTS que controla se denomina Subsistema de Estación Base (*BSS - Base Station Subsystem*).

Subsistema de Red

El subsistema de red (*NSS - Network SubSystem*) está compuesto por el *MSC*, el *HLR* (*Home Location Register*) y el *VLR* (*Visitor Location Register*), e incluye las principales funciones de conmutación en GSM, así como las bases de datos necesarias para los datos de los abonados y para la gestión de la movilidad. La función principal del NSS es gestionar las comunicaciones entre los usuarios GSM y los usuarios de otras redes de telecomunicaciones (RTB, RDSI, etc.). Dentro del NSS, las funciones básicas de conmutación están realizadas por el MSC, cuya función principal es coordinar el establecimiento de llamadas hacia, desde y entre los usuarios GSM. El MSC mantiene enlaces con los distintos BSS de la zona que gestiona. A través de ellos permanece en contacto con los usuarios GSM. Además, por estar en un nivel superior en la jerarquía, los MSC se encargan de coordinar las operaciones de handover realizadas entre distintas BSS. También, el MSC mantiene otros enlaces que permiten el intercambio de datos y voz con otras redes externas. En estos casos, la interfaz con las redes externas, ya sean GSM o no, requiere una puerta de enlace (*GMSC - Gateway Mobile Services Switching Center*). En comunicaciones hacia un usuario GSM, el GMSC se encarga de buscar la información sobre la posición de éste y encaminar la llamada hacia el MSC correspondiente. En las comunicaciones establecidas con una red externa diferente, como por ejemplo la de telefonía fija, el GMSC realiza además la adaptación de los datos y del protocolo de la red a la que se accede.

Para el control y gestión de la localización de los usuarios, en el NSS se disponen de diferentes bases de datos. El Registro de Localización Base (HLR) contiene información sobre los abonados pertenecientes al área gestionada por el NSS. Esta información contiene los datos referentes al suministro de servicios de telecomunicación contratado por el abonado. Una subdivisión funcional del HLR es el Centro de Autenticación (*AuC - Authentication Center*), cuya función consiste, como su propio nombre indica, en la autenticación de los usuarios antes del registro y acceso a los servicios. Por otro lado, el Registro de Localización para Visitantes (VLR) asociado a uno o más MSCs, está encargado del almacenamiento temporal de los datos de aquellos abonados situados en el área de servicio del correspondiente MSC. El VLR mantiene la información necesaria para localizar a cualquier usuario que se encuentre en ese momento dentro del área de acción

del NSS, aunque dicho usuario contenga sus datos de autenticación y contratación de servicios en otro NSS.

3.2.3. Canal de radio

En GSM generalmente el número de canales disponibles en una celda es menor que el número de posibles usuarios dentro de ella. Para ofrecer servicio a más usuarios se emplean técnicas de multiplexación, en las que estos comparten el medio de transmisión de forma ordenada. Aunque existen muchos tipos de multiplexación, el estándar actual de GSM emplea sólo dos de ellos de forma conjunta: multiplexación por división de frecuencias (*FDMA–Frecuency Division Multiple Access*), y multiplexación por división temporal (*TDMA–Time Division Multiple Access*).

Mediante FDMA se hace una división del espectro de frecuencias disponible para GSM. Mediante esta división se obtienen 125 bandas de frecuencia o canales de 200kHz cada uno, numerados de 0 a 124, no utilizándose el canal 0 para evitar interferencias con otros servicios a más baja frecuencia. Estos canales o bandas se agrupan y se reparten entre las estaciones base conforme al sistema celular, cubriendo toda la zona a la que se desea dar servicio.

Aplicando sólo FDMA, el número de canales disponibles en GSM es equivalente a los de un sistema analógico típico, no suponiendo ventaja alguna. Por ello, cada uno de los canales se multiplexa en el tiempo mediante TDMA, lo que permite un uso más eficiente de estos. Así, cada canal se divide en 8 fragmentos de tiempo o *slots*. Cada usuario del sistema debe transmitir de forma discontinua, ocupando uno de estos slots temporales. El conjunto de 8 slots se denomina trama TDMA y su duración total es de 4.615 milisegundos, es decir, cada trama tiene una duración de sólo 577 μs. Debido a la transmisión discontinua, es necesario dejar cierto margen para conmutar entre el encendido y apagado del transmisor. Así, los 577μs de los que dispone un slot, se ven reducidos a 546.12 μs. En este reducido periodo de tiempo, únicamente 148 bits son transmitidos. Estos 148 bits por slot no sólo transmiten la voz, sino también información de control y otros datos. El resultado es que el ancho de banda útil para un canal de tráfico de información es de 22.8 Kbps. Ya que esta tasa resulta insuficiente para la transmisión de voz como en telefonía fija (64 Kbps), se hace necesaria una mayor compresión de la señal de voz.

Por otro lado, la información enviada desde el emisor está contaminada y distorsionada cuando llega al receptor, básicamente por cinco tipos de influencias destructivas: 1) la

modulación, 2) el medio de transmisión, 3) fuentes de ruido, 4) fenómenos de desvanecimiento, y 5) la demodulación.

Cuando se utiliza un teléfono celular analógico en los límites de cobertura de una celda, es posible oír las influencias destructivas del canal, en forma de siseos, desvanecimiento de la señal, interferencias, cortes, clics, etc. Estos efectos temporales que resultan incómodos en las transmisiones analógicas, pueden destruir completamente las comunicaciones digitales por radio. Una inversión en un bit, por ejemplo, puede desde distorsionar levemente la percepción de un fonema a ocultar completamente el significado de una trama completa o cortar la conexión si se produce sobre los datos de control de la llamada. En las aplicaciones realizadas sobre enlaces terrestres por cable se considera confortable una conexión de 40dB de SNR. Si la SNR cayera por debajo de los 10dB en un enlace por cable, éste se consideraría defectuoso y sería reemplazado. En las comunicaciones móviles, desvanecimientos de 20dB ocurriendo del orden de 100 veces por segundo es la mejor condición esperada, siendo el caso típico mucho peor. De hecho, la única característica favorable es que el canal de radio a 900-MHz se comporta de forma lineal. Las propiedades destructivas del canal se clasifican en dos grandes grupos: aquellas que aparecen en condiciones estáticas, y aquellas que suceden en condiciones dinámicas.

Condiciones Estáticas

En este caso se supone que ni el móvil se está moviendo, ni hay nada más moviéndose cerca. En este caso hipotético e inusual, el canal está sujeto a la existencia de ruido aditivo blanco gaussiano (*AGWN–Aditive Gaussian White Noise*) y gobernado por la propagación atmosférica. El AGWN generalmente es producido por el propio terminal móvil y aparatos electrónicos del entorno. La propagación de la señal por la atmósfera da lugar a la propagación por múltiples caminos, zonas con sombras, y retardos que pueden ser de incluso varios microsegundos. Pérdidas completas de la señal pueden perderse debido a desvanecimientos planos en el espectro superiores a 40dB. La ecualización del canal mediante filtros adaptativos se usa para eliminar la interferencia intersimbólica.

Condiciones Dinámicas

Si suponemos que el terminal móvil se mueve (como es evidente), se añaden los efectos de la propagación terrestre, predominando la influencia más destructiva de todas: los desvanecimientos *Rayleigh*. Dado que las ondas de radio pueden seguir una variedad de caminos reflexivos hasta el receptor móvil, pueden ocurrir cambios de fase, dependientes

de la frecuencia, que se superpongan de forma sustractiva en el receptor. Al sustraerse, aparecen desvanecimientos importantes que ocurren de forma momentánea. En estos desvanecimientos se producen interferencias con los canales de otros terminales y estaciones base distantes. Para paliar estos efectos se introducen saltos pseudo-aleatorios en la frecuencia de transmisión. Estos saltos, además de añadir privacidad en las comunicaciones, diversifica el uso de frecuencias, reduciendo el efecto de los desvanecimientos Rayleigh.

Estos hechos ofrecen una idea de lo hostil que resulta el canal en las comunicaciones móviles. Cualquiera de las alteraciones en amplitud, frecuencia o fase, produce una pérdida de información que se puede medir en bits por segundo. La tasa de bits incorrectos (*BER–Bit Error Rate*) en comunicaciones móviles es millones de veces mayor que en un enlace por cable [57]. Por esta razón, los bits resultantes de la codificación de la voz además se codifican con bits redundantes de control y recuperación de errores, durante lo que se conoce como *codificación del canal*.

3.2.4. Codificación del canal

El objetivo de la codificación del canal es asegurar la integridad de la información que se transmite por el canal de comunicación. En GSM una buena parte del ancho de banda disponible se dedica a esta codificación, ya que el canal, como hemos descrito, está altamente expuesto al ruido.

La codificación del canal en GSM se realiza teniendo en cuenta la naturaleza de la información que se quiere transmitir, distinguiéndose diferentes tipos de canales. Estos canales se pueden agrupar en los siguientes grupos: 1) los canales de tráfico (*Traffic Channels*), 2) los canales de control (*Control Channels*), y 3) los canales de conmutación de paquetes (*Packet Switched Channels*). Todos estos canales se encuentran completamente detallados en el estándar ETSI GSM 5.03 [58].

En los canales de tráfico se agrupan los canales de datos y los canales de voz. Los primeros están dedicados a la transmisión de datos del usuario, sin hacer ninguna suposición acerca de estos, y dependiendo de la velocidad a la que se quiere transmitir (14.4, 9.6, 4.8 y 2.4 Kbps) se definen diferentes codificaciones de canal. Los segundos están dedicados a la transmisión de voz. En ellos se hacen suposiciones acerca de la importancia que tiene

cada bit para la síntesis posterior de la voz. La importancia de cada bit depende del codificador de voz que se emplee para la transmisión. Por esta razón, se definen diferentes canales de voz según el *codec* (*cod*ificador-*dec*odificador) empleado.

GSM tiene estandarizados cuatro codecs para la transmisión de voz, junto a los que se define su propia codificación de canal:

- *Full Rate* (*FR*). Fue el primer codificador de voz, introducido en la fase 1 de la estandarización GSM. El canal de tráfico de voz asociado se denomina TCH/FS (*Traffic Channel - Full rate Speech*). Este canal tiene una tasa de bits de 22.8 Kbps, de los que 13 Kbps se dedican a la codificación de voz y 9.8 Kbps a la codificación de canal.

- *Enhanced Full Rate* (*EFR*). Introducido en la fase 2 de la estandarización, este codec aporta una mejora significativa en la calidad de voz a la vez que se reduce ligeramente su tasa de bits. El canal de tráfico de voz asociado se denomina TCH/EFS (*Traffic Channel - Enhanced Full rate Speech*). Este canal tiene una tasa de bits igual al TCH/FS, pero 12.2 Kbps se dedican a la codificación de voz y 10.6 Kbps a la codificación de canal.

- *Half Rate* (*HR*). Fue introducido junto con EFR en la fase 2 de la estandarización. Este codec reduce a la mitad la tasa de bits requerida para la codificación de la voz. El objetivo inicial era permitir llamadas con una peor calidad pero de coste reducido, sin embargo actualmente ha caído en desuso. El canal de tráfico de voz asociado, TCH/HS (*Traffic Channel - Half rate Speech*), requiere sólo 11.4 Kbps, en el que 5.6 Kbps se dedican a la codificación de voz y 5.8 Kbps a la codificación de canal.

- *Adaptive MultiRate* (*AMR*). Éste es el más avanzado codificador de voz de GSM. Se caracteriza por adaptar la tasa de bits a las exigencias del canal en cada momento distribuyendo los bits dedicados a la codificación de voz y de canal de forma dinámica, consiguiendo una mayor calidad de voz. Realmente, esta asignación no es totalmente dinámica, sino que el codec distingue entre varios modos de operación, cada uno con sucesivas reducciones en la tasa de bits. Así, conforme el canal se degrada, se cambia de modo, y más bits están disponibles para una codificación más robusta del canal. Salvando las diferencias, los modos pueden considerarse como distintas adaptaciones del codec EFR, en donde los parámetros que representan la

voz se codifican con menos bits. Así, el modo de mayor calidad es casi idéntico a EFR.

A pesar de las diferencias entre los distintos codecs, así como de las diferentes tasas de bits que generan, todas las codificaciones de canal de tráfico para voz tienen en común la mayor protección de aquellos bits de mayor importancia para la síntesis posterior de la voz. En el siguiente capítulo se describirán en detalle los codificadores EFR (el codec de mayor calidad con tasa de bits fija) y AMR (el único codec con tasa adaptativa). Por el momento, baste decir que se pueden distinguir ciertos parámetros (y por tanto, bits que los codifican) de mayor relevancia perceptual. La alteración de estos bits puede ocasionar efectos desastrosos durante en la síntesis de voz, siendo su audición muy desagradable para el receptor.

Las codificaciones de canales de voz distribuyen los bits procedentes del codificador de voz en tres grupos, dependiendo de su importancia subjetiva en la reconstrucción de la señal:

- *Clase Ia.* Bits muy sensibles a errores. La síntesis de tramas con bits de este tipo alterados pueden ser muy desagradables para el receptor, de hecho, es preferible no reproducirlas.
- *Clase Ib.* Bits moderadamente sensibles a errores. La síntesis de bits erróneos de este tipo pueden degradar la calidad de voz, pero sin llegar a provocar una seria perturbación.
- *Clase IIa.* Bits poco sensibles a errores. La modificación de estos bits tiene un escaso impacto en la calidad de voz.

Los datos de voz son codificados en dos pasos. Primero, a los bits de la clase Ia se le añade un *código de redundancia cíclica* (*CRC*). Este CRC simplemente se usa para detección de errores, indicando al receptor, en el otro extremo del canal de radio, si han ocurrido errores que no han podido ser recuperados. Ésta es la primera etapa del proceso de codificación del canal y, su inversa, la última del de decodificación.

El segundo paso de la codificación de canal lo constituye una *codificación convolucional*. Esta codificación añade bits de redundacia de tal forma que el decodificador pueda, dentro de unos límites, detectar errores y corregirlos. El código convolucional se aplica tanto a los bits de la clase Ib como a los de la clase Ia, incluyendo estos últimos, su CRC. Antes de la codificación convolucional se añaden a estos bits una secuencia de bits puestos a cero

llamada *secuencia de cola*. El objetivo de esta secuencia de cola es inicializar el codificador convolucional.

Tras la codificación convolucional, se añaden a los bits resultantes, los bits de la clase IIa sin ningún tipo de protección añadida, generando un total de 456 bits (FR, EFR y AMR) o 228 bits (HR). Aunque este número de bits encaja perfectamente en 4 slots de tiempo (2 en el caso de HR), no se introducen de forma consecutiva en ellos. Esto se debe a que, en caso de que se produzcan ráfagas, la trama puede quedar completamente destruida. Previamente, se aplica un entrelazado de los datos. Este entrelazado dispersa también los posibles errores consecutivos, lo cual mejora la capacidad de corrección de los códigos convolucionales.

3.2.5. Mitigación de errores

A pesar de la protección incluida, las tramas pueden contener errores cuando alcanzan el receptor. En éste, las tramas son clasificadas como correctas o incorrectas, principalmente en base al resultado del código cíclico. Una marca, denominada BFI (*Bad Frame Indicator*), indica si la trama se ha recibido correctamente o no. Cuando esta marca está activa indica que las protecciones han fallado y que bits importantes para la decodificación de la voz han sido alterados. De forma general, se suele hablar de tramas incorrectas o erróneas cuando la marca BFI está activa (BFI=1) y tramas correctas cuando la marca no lo está (BFI=0). Sin embargo, lo único que puede afirmarse formalmente es que el decodificador del canal ha determinado que bits críticos para la síntesis de voz han sido alterados cuando la marca está activa, y que no los ha detectado cuando la marca está inactiva. Durante el análisis de los errores y sus consecuencias en el reconocimiento que se realizará más adelante en este mismo capítulo comentaremos esto con más detalle.

La decodificación de las tramas erróneas o con bits significativos alterados genera efectos en la síntesis muy desagradables. Para mejorar la calidad subjetiva de la señal, las tramas erróneas son sustituidas con una repetición o una extrapolación de la última o últimas tramas correctas. Esta sustitución se realiza de forma que, gradualmente, se reduzca el nivel de la salida. El efecto que se pretende conseguir es el de un apagado gradual de la señal que si se prolonga da lugar a ruido de confort. Este ruido tiene como objetivo producir la sensación subjetiva de que el terminal móvil está funcionado.

GSM propone distintos métodos de sustitución y apagado en función de los parámetros disponibles para cada codec (GSM 6.11 [59] para FR, GSM 6.21 [60] para HF, GSM 6.61 [61] para EFR, y GSM 6.91 [62] para AMR). En estos estándares no se impone ningún

algoritmo concreto de sustitución y apagado, tan sólo propone una posible implementación (usualmente la propuesta por los autores del codificador). Ésta suele consistir en el reemplazo parcial de los parámetros recibidos en tramas incorrectas con valores extrapolados de las tramas correctas anteriores. Junto con esta extrapolación, se limitan de forma sucesiva los valores de las ganancias, de forma que la recepción de tramas incorrectas conduzca a un apagado progresivo de la señal que, si se prolonga, degenera en *ruido de confort*.

Aunque tan sólo es obligatoria la condición de que sea una sustitución y apagado, dejando la implementación en manos de los fabricantes, la posible solución ofrecida en el estándar es la que comúnmente se emplea en la mitigación de tramas incorrectas.

3.3. Redes IP

El inicio de Internet y las redes IP, tal y como lo conocemos hoy en día suele situarse a mediados de los años setenta, cuando se estableció la red de conmutación de paquetes ARPANET. El objetivo de esta red era proporcionar un sistema para realizar pruebas de investigación sobre conmutación de paquetes en redes de área amplia. La conmutación de paquetes era considerada un enfoque prometedor para poder compartir recursos entre computadores.

Un *paquete* es un bloque que consta de una cabecera, que contiene la dirección destino, junto a la información que desea transmitir el usuario, y que se desplaza como una unidad a través de la red, de forma similar a como lo haría un telegrama en la red de telégrafos. ARPANET estaba formada por conmutadores de paquetes conectados mediante líneas de transmisión que proporcionan varios caminos por los que interconectar computadoras separadas por grandes distancias. Los conmutadores de paquetes se diseñaron usando minicomputadoras de propósito específico, que se conectaban por lo menos a otros dos conmutadores, teniendo así un camino alternativo en caso de fallo de un enlace. La transmisión en ARPANET no estaba orientada a conexión, en el sentido de que no se establece conexión antes de la transmisión de un paquete. Por tanto, los paquetes se podían transmitir sin sufrir el retardo necesario para el establecimiento de la conexión. Cada paquete o *datagrama* contenía la dirección destino, utilizada por los conmutadores para realizar el encaminamiento, mientras que cada conmutador de paquetes mantenía una *tabla de encaminamiento* en la que se especificaba la línea de salida que había que usar para cada destino. Los paquetes se almacenaban temporalmente en memoria hasta que eran retransmitidos por la correspondiente línea de salida. Así, los paquetes generados por diferentes

usuarios se multiplexaban en los enlaces que unían a los distintos conmutadores. Debido a que no era necesario el establecimiento de la conexión, no se realizaba a priori ninguna reserva ni de ancho de banda ni de memoria de almacenamiento temporal.

Tras el éxito de ARPANET, ARPA empezó a explorar las comunicaciones de datos usando satélites y redes de paquetes de radio móviles. Dada la naturaleza diferente de estas redes, era evidente la necesidad de desarrollar protocolos que abstrajeran la red física, permitiendo la compatibilidad entre todas las variantes de redes. La figura 3.5 muestra un ejemplo de una interconexión de dos redes. Para que la unión sea posible se hace indispensable un elemento llamado *dispositivo de encaminamiento*. Este dispositivo se conecta a ambas redes, conociendo los detalles específicos de la transmisión por cada una de ellas, y sirve de pasarela entre ambas. Este dispositivo también se encarga de acomodar la información de una red a la otra, solventando los problemas que esto pudiera ocasionar. Así, cuando un computador de la red A desea enviar información a otro de la red B, se la envía al dispositivo de encaminamiento. El dispositivo de encaminamiento acomoda la información de una red a otra y la retransmite a la máquina correspondiente. Este mecanismo parece fácil en un principio, pero no está exento de dificultades. Así, las máquinas de las redes A y B deben conocer la localización del dispositivo de encaminamiento y cuando deben usarlo para comunicarse con las máquinas de otra red. Igualmente, el dispositivo debe conocer como realizar las retransmisiones para que la información llegue a su destino. En el caso en que únicamente dos redes se interconectan puede resultar relativamente sencillo. Sin embargo, a medida que aumenta la complejidad del entramado de redes interconectadas, más complejo resulta el reenvío de información. Así pues, el problema consistía en automatizar todas estas tareas, de forma que el intercambio de información entre las redes fuera transparente, y el usuario viera la unión de todas éstas como una única red virtual.

El conjunto final de protocolos conformó lo que se conoce como *TCP/IP*. Estas siglas hacen referencia a los dos protocolos más importantes de comunicación en Internet (aunque el nombre suele emplearse para todo el conjunto de protocolos): el protocolo Internet (*IP - Internet Protocol*) y el protocolo de control de transmisiones (*TCP - Transmission Control Protocol*).

3.3.1. Características de las redes IP

El objetivo durante el diseño del TCP/IP fue permitir la interconexión de redes heterogéneas, mediante unos servicios de comunicación universales, proporcionando al usua-

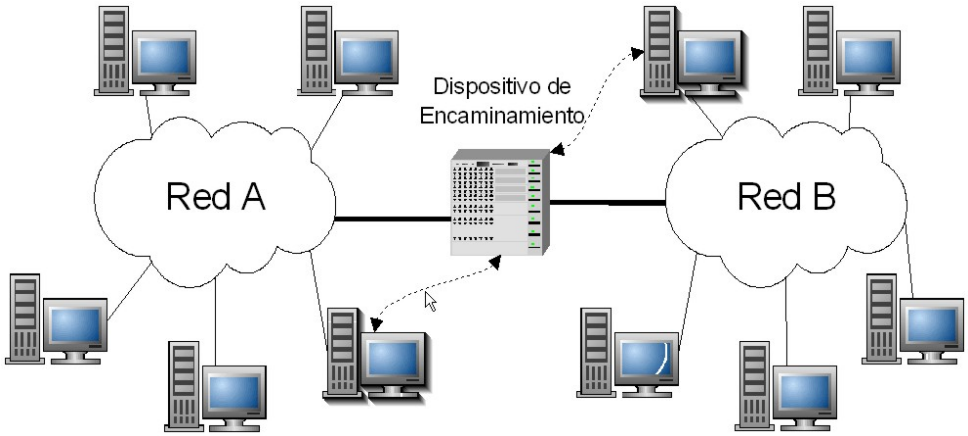


Figura 3.5: Interconexión de dos redes a través de un dispositivo de encaminamiento. El computador de la red A se comunica con el de la red B mediante el dispositivo de encaminamiento.

rio la ilusión de una única red virtual. Además de la independencia sobre la red que se implementa, otro aspecto relevante de una red IP es que ofrece un servicio no orientado a conexión y de mínimo esfuerzo. Un servicio no orientado a conexión implica que no se establece, previamente a la transmisión, una reserva de recursos en la red, como ocurre, por ejemplo, en una llamada telefónica. Para ello, se emplea la misma aproximación que la red originaria ARPANET, en donde se transmiten datagramas. En este sentido, Internet se parece a una red de correo postal. Las redes basadas en envío de datagramas permiten un uso más eficiente de los recursos que las redes orientadas a conexión. Mientras que en estas últimas los recursos sólo pueden emplearse por el usuario que los ha reservado, esté haciendo uso de ellos o no, en las primeras los recursos son empleados para la transmisión del paquete, quedando después libres para el resto de usuarios. Además, al no estar orientado a conexión, los dispositivos de encaminamiento no guardan información sobre el estado de los usuarios o de sus flujos de información, dando lugar a una reducción de la complejidad. Esta aproximación permite que las redes IP sean escalables a gran tamaño.

Como servicio de mejor esfuerzo, la red IP sólo intenta hacer llegar los paquetes hacia su destino, pero no evita que se pierdan, corrompan o desordenen, o incluso que se entreguen equivocadamente. Un servicio tan poco fiable puede resultar extraño, pero la exigencia de que los protocolos TCP/IP funcionen a través de cualquier tipo de red implica que debe primar la simplicidad. En este sentido, el servicio que ofrece una red IP no es fiable, puesto que proporcionar fiabilidad dentro de redes interconectadas conlleva un aumento considerable en la complejidad de los dispositivos de encaminamiento.

3.3.2. Pila de Protocolos TCP/IP

Las transmisiones en redes IP se realizan siguiendo una serie de protocolos. La familia de protocolos TCP/IP constituye todo un mundo, existiendo un gran número de protocolos para realizar multitud de tareas. Un número que además se va ampliando conforme nuevos servicios se van añadiendo a la red. Como la mayoría del software de red, los distintos protocolos se van agrupando por capas o niveles de abstracción. Cada una de estas capas se apoya en la anterior y ofrece nuevas funcionalidades a la capa siguiente, conformando así la pila TCP/IP. La figura 3.6 muestra la pila de protocolos TCP/IP junto con algunos protocolos importantes de cada una de las capas. Esta pila se compone básicamente de 4 capas:

- Capa de Aplicación. Esta capa la provee el programa que usa TCP/IP para realizar comunicaciones e interactuar con otros programas en otros computadores.
- Capa de Transporte. Esta capa proporciona la transmisión de datos de extremo a extremo a las capas superiores. La capa de transporte permite que una aplicación de una máquina se comunique con otra aplicación ejecutándose en otra máquina, pudiendo coexistir varias aplicaciones que utilicen concurrentemente esta capa. En ella se sitúan los conocidos protocolos UDP y TCP, usándose cada uno de ellos según las necesidades de seguridad en la transmisión.
- Capa de Interconexión. El objetivo de la capa de interconexión es hacer transparente la existencia de una comunicación que implica a múltiples redes. Así, esta capa ofrece a la capa de transporte una única red virtual sobre la que se realiza la comunicación. En esta capa se sitúa el protocolo IP, considerado la base de Internet. Con él se asocian protocolos para la asignación de identificadores virtuales dentro de esta única red, llamadas direcciones IP, a las máquinas reales, así como protocolos para el mantenimiento y gestión de la red.
- Capa de interfaz de red. Es la capa que gestiona la red real sobre la que se realiza comunicación. Como es evidente, la implementación de esta capa diferirá según el tipo de red y será especifica para cada una. Por ésta razón TCP/IP no especifica ningún protocolo, solamente estandariza la manera en que se accede desde la capa de interconexión, dejando el resto a los propios fabricantes.

La división en capas del proceso de comunicación permite la división de las tareas, facilitando la implementación y las pruebas y correcciones en el código. Gracias a que sólo

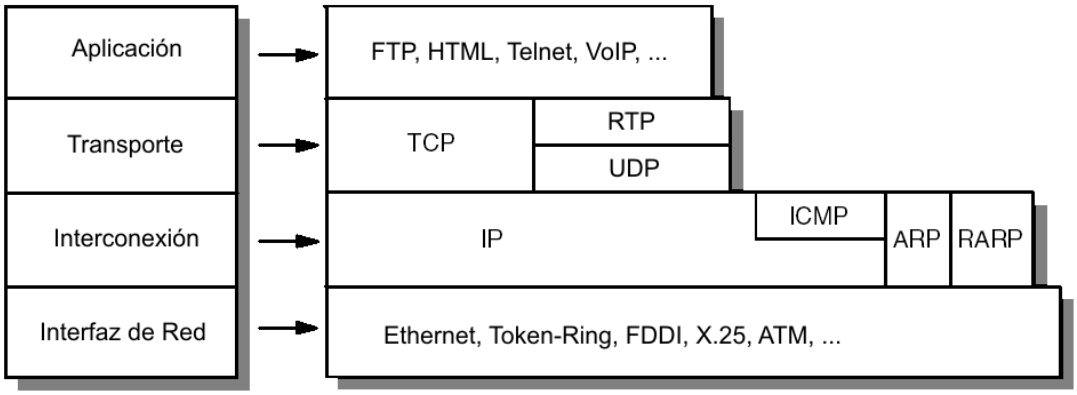


Figura 3.6: Estructura en capas y protocolos más comunes de la pila TCP/IP.

está especificada de forma concisa la manera en la que las capas se comunican, se permite la creación de capas con implementaciones alternativas. Esto es, a una capa no le interesan los detalles de implementación de las capas inferiores, sólo cómo debe comunicarse con ellas.

3.3.3. Transmisiones de tiempo real en redes IP

Las redes de conmutación de circuitos se han venido usando en las comunicaciones tradicionales con requerimientos de tiempo real (*RT - Real Time*). En estas redes se transmiten flujos constantes y estables de información con retardos muy cortos. Las llamadas en telefonía, así como la videoconferencia con baja tasa de bits, son ejemplos típicos de este tipo de comunicaciones. Por el contrario, las redes de paquetes se desarrollaron fundamentalmente para el transporte de datos que no tenían especiales requisitos de temporización.

Los avances en algoritmos de compresión, las mejoras en la potencia de cálculo de los computadores y el incremento del ancho de banda disponible, además de la capacidad de conmutación de paquetes de los dispositivos, están posibilitando las comunicaciones en tiempo real a través de redes de paquetes. Las cuales, además, presentan funcionalidades adicionales como, por ejemplo, la transmisión a múltiples destinatarios, que no es fácilmente implementable en las redes de circuitos conmutados. La transmisión de paquetes en tiempo real, no obstante, no está exenta de dificultades, ya que no fue ideada con ese objetivo. En este sentido se deben solventar dificultades inherentes a la propia conmutación de paquetes, entre las que cabe citar: el retardo de los paquetes, la dispersión temporal, la pérdida de paquetes, y la entrega desordenada o la aparición de paquetes duplicados.

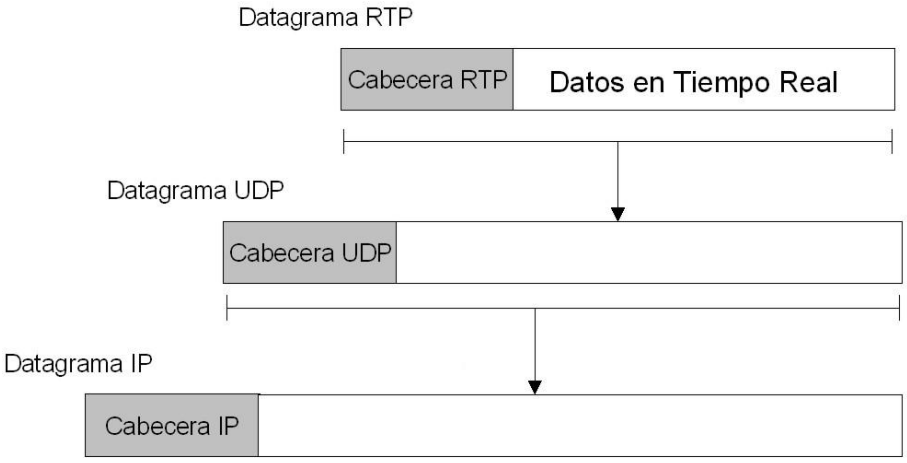


Figura 3.7: Encapsulado de los datos en comunicaciones IP con restricciones de tiempo real.

Las transmisiones con requerimientos de tiempo real, como el reconocimiento remoto, se realizan en las redes IP a través del protocolo de transferencia de tiempo real (*RTP–Real Time Protocol*). Este protocolo está orientado a las transmisiones multimedia y añade funcionalidades a las capas inferiores de forma que sea posible este tipo de comunicación. De acuerdo con la figura 3.6 vista anteriormente, el protocolo RTP se apoya en el protocolo *UDP* (*User Datagram Protocol*). El mecanismo por el cual los protocolos de una capa hacen uso de los de la capa subyacente se denomina encapsulado. Mediante el encapsulado los paquetes de un protocolo se integran dentro de los paquetes de otro. La figura 3.7 muestra el encapsulado de los paquetes RTP hasta la capa de interconexión de red. Empezando de menor a mayor funcionalidad, describiremos cada uno de estos protocolos, necesarios para habilitar la transmisiones de tiempo real en las redes IP.

Protocolo IP

El Protocolo Internet (*IP–Internet Protocol*) [63] se desarrolló para la transferencia de paquetes a través de redes interconectadas y conforma la base de las redes IP. IP es independiente de cualquier red o plataforma, pudiendo ser éstas de cualquier tipo. Como comentamos, en IP las redes constituyentes se interconectan mediante conmutadores de paquetes especiales denominados pasarelas, dispositivos de encaminamiento, o simplemente *encaminadores*. Estos dispositivos pueden conectarse a dos o más redes, siendo los encargados de dirigir la transferencia de paquetes IP en Internet. Las redes subyacentes son las encargadas de transmitir los paquetes entre los dispositivos de encaminamiento o

distribuirlos en su propia red, en el caso de que la máquina destino estuviera en ella.

Puesto que es independiente de la red sobre la que implementa, IP utiliza su propio espacio de direcciones, definiendo así una única red virtual. Este espacio está formado por direcciones IP, que consisten en cuatro bytes expresados normalmente en notación decimal separados por un punto; por ejemplo, 150.214.60.156. Este espacio de direcciones se organiza de forma jerárquica incluyéndose información sobre la localización de las máquinas dentro de toda la estructura. Así, las direcciones IP tienen dos partes: un identificativo de red y un identificativo de la máquina. Las máquinas situadas en la misma subred compartirán parte de la dirección, lo cual permite a los dispositivos de encaminamiento procesar de igual manera a todas las direcciones que tengan el mismo prefijo. Debido a la dificultad que conlleva memorizar una dirección IP, en Internet se ofrece también un espacio de nombres que identifica a las máquinas conectadas a la Internet. El sistema de nombres de dominio (*DNS–Domain Name System*) [64, 65] se encarga de traducir automáticamente los nombres a direcciones. El espacio de nombres tiene una estructura jerárquica, pero es puramente administrativa y la red no la usa para encaminar.

La transferencia de bloques independientes de información mediante datagramas puede servir para muchas aplicaciones. Sin embargo, hay otras aplicaciones que requieren la transferencia segura de cadenas de información secuenciadas u ordenadas correctamente, o transmisiones en tiempo real para aplicaciones multimedia. Así pues, sobre IP se implementan dos protocolos que permiten llevar a cabo un gran rango de aplicaciones: TCP [66] y UDP.

Protocolo UDP

El protocolo de datagrama de usuario (*UDP–User Datagram Protocol*) [67] es un protocolo de la capa de transporte no orientado a conexión y no fiable. Es un protocolo muy simple que proporciona solamente dos servicios más que los que proporciona IP: demultiplexación y comprobación de errores en los datos.

Aunque IP sabe cómo entregar paquetes a una computadora, no sabe a que aplicación específica de la computadora debe entregarlos. UDP añade un mecanismo que distingue entre las diversas aplicaciones en la computadora. Por otro lado, IP sólo comprueba la integridad de su cabecera. UDP puede opcionalmente comprobar la integridad del datagrama UDP completo. Generalmente, las aplicaciones que requieren tratamiento en tiempo real, no emplean el protocolo TCP para la transmisión, sino que trabajan sobre el protocolo UDP. Aunque el protocolo TCP asegura la recepción de todos los paquetes emitidos por

el emisor, no asegura que dicha recepción se obtenga en un tiempo razonable [66]. Esto se debe a que TCP asegura la recepción mediante la retransmisión de los paquetes perdidos. En muchas ocasiones, sobre todo en las aplicaciones de transmisión de audio o video, no importa tanto la recepción de todos los paquetes como el retraso con el que llegan. Por ejemplo, en una transmisión de voz, si llega un paquete con información acerca de una trama que debería haberse reproducido antes de la trama que se está sintetizando ahora, ese paquete es prácticamente inútil. UDP tampoco asegura la recepción dentro de un tiempo razonable (ni siquiera asegura la recepción), pero no presenta los problemas de retransmisión de TCP. Si un paquete es transmitido por UDP y no alcanza el destino se da por perdido, mientras que si un paquete se transmite por TCP y no se alcanza el destino, tras un cierto tiempo, se vuelve a retransmitir. Seguramente, para cuando el paquete se haya recibido ya no será necesario, consumiendo unos recursos que se podrían aprovechar para enviar paquetes más recientes.

Un campo en la cabecera, llamado puerto destino, permite a la capa UDP demultiplexar los datagramas para cada aplicación en la computadora destino, mientras que un puerto origen identificará la aplicación particular en la computadora origen que recibirá las respuestas. La protección de errores se realiza a través del campo suma de comprobación, mediante el que UDP detecta errores en el datagrama. El uso de este campo es opcional. Si una computadora origen no desea que se calcule la suma de comprobación, este campo debe contener sólo ceros, de forma que la computadora destino sepa que la suma de comprobación no se ha calculado. Si el resultado de la comprobación es precisamente todo ceros, entonces se empleará la otra representación para el cero en complemento a 1, es decir, todo el campo se rellena de unos.

Protocolo RTP

UDP, tal cual, no ofrece los servicios necesarios para la transmisión en tiempo real, ya que no provee de los mecanismos necesarios para solventar dificultades tales como la dispersión temporal o posibles pérdidas, así como la recuperación de la señal de reloj y la sincronización entre los distintos tipos de medios enviados (por ejemplo, sincronizar la imagen con el sonido en una videoconferencia). Por esta razón, sobre UDP se implementa el protocolo de transporte para tiempo real (*RTP–Real-Time Transport Protocol*) [68]. RTP proporciona transporte de datos de extremo a extremo entre aplicaciones que requieran transmisión en tiempo real. Desgraciadamente, RTP no proporciona ningún medio para asegurar la

temporización en la entrega de la información, ni tampoco proporciona garantías de calidad de servicio, dependiendo éstos de las garantías que ofrezca la red subyacente. RTP ofrece los siguientes servicios:

- Identificación del tipo de información transportada (denominada carga útil). Los paquetes RTP pueden transportar fragmentos de diferentes medios, como audio y video, codificados de diferente manera. Para diferenciar cada uno de esos flujos, la aplicación emisora incluye un identificador en la cabecera del paquete RTP. Este identificador indica el esquema especifico de codificación empleado para generar la secuencia de bits, de forma que el receptor sea capaz de seleccionar el decodificador adecuado a través de él.

- Numeración secuencial. Durante la transmisión, el orden de recepción de los paquetes no tiene porqué coincidir con el orden de emisión. Esta numeración es empleada en el receptor para obtener la secuencia original de los paquetes durante la reconstrucción.

- Inclusión de marcas de tiempo. Las marcas de tiempo permiten la sincronización entre paquetes de diferentes fuentes. Esta marca representa el momento en el que la trama fue creada.

Adicionalmente a RTP, se implementa el protocolo de control RTP, (*RTCP–RTP Control Protocol*). Este protocolo asociado sirve para monitorizar la calidad del servicio observado por el receptor, así como para informar al emisor de esto y de otras cuestiones relativas a los participantes. Esta funcionalidad es especialmente útil en aquellas situaciones en las que el emisor pueda adaptar su algoritmo a las condiciones imperantes en la red; por ejemplo, al ancho de banda disponible, el retardo o la dispersión temporal en la red.

El RTP es un protocolo que, intencionadamente, no se ha especificado de forma completa, y está ideado para ser lo suficientemente flexible como para poder ser incorporado en las aplicaciones sin necesidad de implementarse en una capa separada. El uso del RTP en una aplicación en particular exige documentación complementaria. Para un tipo determinado de aplicación, como, por ejemplo, de audio o vídeo, existirá un documento denominado *perfil* que ha de definir los atributos y/o modificaciones y extensiones RTP. Adicionalmente, debe también existir un documento denominado *formato de la carga útil* que define cómo se transporta con RTP los datos con requerimientos de tiempo real.

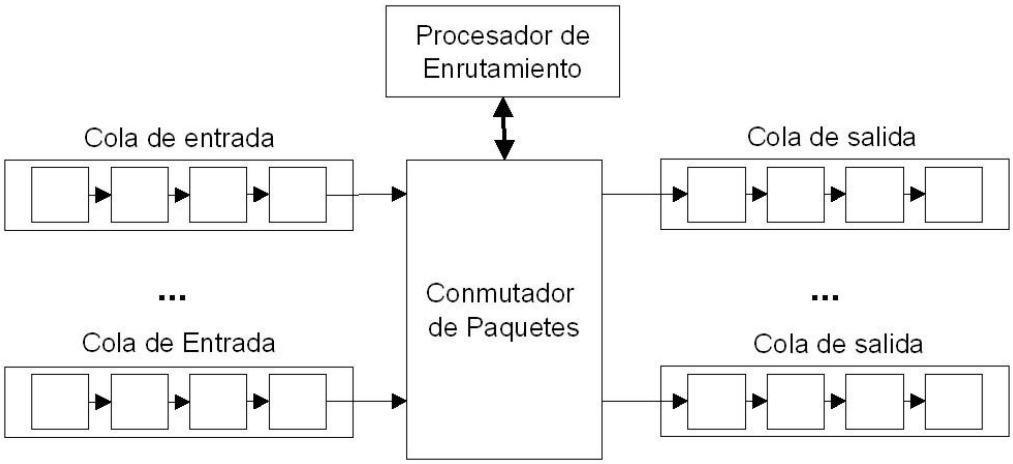


Figura 3.8: Estructura interna de un dispositivo de encaminamiento.

3.3.4. Pérdida de paquetes en redes IP

Como comentamos anteriormente, las redes IP no fueron diseñadas para ofrecer un servicio confiable de transmisión de paquetes, por lo que no se asegura que todos los paquetes lleguen a su destino. Ignorando problemas eventuales en la red y suponiendo que el sistema funcione correctamente, la raíz del problema de la pérdida de paquetes radica en los dispositivos de encaminamiento.

Como ya dijimos, los dispositivos de encaminamiento son los encargados de interconectar las redes, acomodando y reenviando sucesivamente los paquetes a lo largo de toda la red IP. En la figura 3.8 se muestra un esquema básico de un dispositivo de encaminamiento o *router*. Como puede apreciarse, un router consta de una serie de colas de entrada y de salida, además de una lógica interna que se encarga de decidir hacia que cola de salida deben ser reenviados los paquetes de las colas de entrada.

Las colas implementadas en los routers son del tipo *FIFO* (*First Input First Output*). El objetivo de estas colas es acomodar las distintas velocidades de las redes de entrada y de salida y de la lógica de conmutación de paquetes. Cuando un paquete llega a un router, éste se almacena en una cola de entrada a la espera de ser procesado. Tras tomar la decisión de encaminamiento, el paquete se coloca en su correspondiente cola de salida a la espera de que la red pertinente quede libre para que pueda ser transmitido. La decisión de encaminamiento se realiza en base a la dirección de los paquetes IP (cómo sabe el router a partir de la dirección hacia que red debe enviar el paquete depende de múltiples protocolos y algoritmos que quedan fuera del interés de este trabajo). En este entorno existen tres puntos problemáticos:

- Si la velocidad conjunta de los flujos de entrada es superior a la capacidad de proceso de la lógica de conmutación, los paquetes se acumularán en las colas de entrada. Si la situación se prolonga lo suficiente, la memoria reservada para las colas se desbordará.

- Si la velocidad de un flujo de salida es inferior a la capacidad de proceso de la lógica de conmutación, los paquetes se acumularán en las colas de salida. Como en el caso anterior, si la situación se prolonga, la cola de salida se desbordará.

- Debido a la estructura FIFO de las colas, si un paquete no puede ser conmutado hacia una cola de salida, no sólo se produce un retraso en este paquete, sino sobre todos los paquetes posteriores a él. Generalmente las lógicas de conmutación no permiten conmutar paralelamente dos paquetes de flujos de entrada hacia un mismo flujo de salida, por lo que uno de los paquetes sufre un retraso, junto con todos los posteriores en la cola.

Puesto que IP mantiene una filosofía de simplicidad y mejor esfuerzo, el paquete que llega una cola saturada es descartado, siendo responsabilidad de los extremos afrontar este problema. Para simular un envío confiable de información en donde no haya pérdida de paquetes, el protocolo TCP implementa algoritmos que, pasado un cierto tiempo sin que se reciba el paquete en el receptor, reenvían el paquete perdido. Como ya comentamos, el envío confiable proporcionado por TCP resulta útil para aplicaciones que no tengan restricciones temporales y que necesiten de todos los paquetes para realizar el servicio, como pueden ser las aplicaciones Web o FTP. Sin embargo, para aplicaciones con restricciones de tiempo, TCP no aporta ventajas, ya que un paquete recibido fuera de tiempo es equivalente a uno perdido, y sí inconvenientes, puesto que la retransmisión de paquetes consume recursos que bien podrían emplearse para enviar paquetes útiles. Por esta razón, el protocolo RTP se utiliza por encima de UDP, en donde no hay retransmisión de paquetes perdidos.

La posible saturación en las colas FIFO es un problema inherente al propio proceso de conmutación de paquetes. Si suponemos que todos los flujos de entrada alcanzan una velocidad máxima de V paquetes por segundo, existiendo N flujos de entradas, entonces la lógica de conmutación debe ser capaz de conmutar a NV paquetes por segundo. De esta forma, se conseguiría eliminar la saturación en las colas de entrada. Sin embargo, si todos los paquetes deben ser conmutados hacia el mismo flujo de salida, la velocidad de los flujos de salida deberían ser igualmente de NV paquetes por segundo para evitar la congestión

en las colas de salida. Aún en este caso, se estaría diseñando una red basándonos en situaciones de máximos, suponiendo un grave desperdicio del ancho de banda disponible.

Para poder ofrecer servicios de tiempo real garantizados se proponen diferentes cambios en el funcionamiento y dispositivos de la red para hacer frente a los retrasos en la recepción y a la pérdida de paquetes. La solución más simple es incrementar la capacidad de las redes que conforman Internet, de forma que sea improbable la saturación en las colas. Sin embargo, la experiencia demuestra que al aumentar las prestaciones ya sean de computación, almacenamiento o transmisión, nuevas aplicaciones surgen capaces de coparlas. Así pues, se prevén otro tipo de soluciones:

- Los servicios integrados [69] proponen introducir mecanismos de reserva dinámica de ancho de banda. Esto hace necesario la existencia de protocolos para realizar la reserva de recursos así como la modificación del funcionamiento de los routers para proporcionar las reservas. Básicamente, los routers deben incluir un clasificador de paquetes que identifique los distintos flujos y mecanismos para controlar la admisión (decidir si se dispone o no de suficientes recursos) y supervisión del tráfico generado por las aplicaciones.

- Los servicios diferenciados [70], por otro lado, no proponen cambios significativos en la red, basándose en esquemas sencillos de supervisión en los extremos. Los paquetes son clasificados y marcados (a través de los campos ya existentes en el protocolo IP) para recibir un tratamiento diferenciado a lo largo de los nodos por los que se retransmite. Se espera que este tratamiento diferenciado permita acomodar la transmisión realizada por la red a las expectativas del usuario.

- Finalmente, las redes privadas virtuales [71, 72] proponen una reserva permanente de porciones de ancho de banda. Los routers diferenciarían el tráfico de la red privada a través de las direcciones IP, proporcionando el ancho de banda reservado mediante estrategias de control y planificación.

Todas estas alternativas suponen un gasto adicional para acomodar la red IP a aplicaciones de tiempo real. Debido a esto, suelen emplearse en entornos corporativos, de forma reducida, no siendo el caso general de la Internet actual.

3.4. Arquitecturas para el reconocimiento remoto del habla

Como se planteó anteriormente en la sección 2.1.1, el reconocimiento del habla puede verse como un problema de reconocimiento de formas en el que se pueden distinguir tres tareas: adquisición de la voz, análisis o parametrización de la voz, y reconocimiento de patrones. Estas tres tareas no tienen porqué necesariamente coexistir en un mismo lugar, pudiendo realizarse en localizaciones distintas siempre que exista alguna red de comunicación entre ellas.

Dependiendo de donde situemos la red de comunicaciones en la cadena de procesamiento y reconocimiento de voz, podemos distinguir dos alternativas o arquitecturas para el reconocimiento remoto de la voz. En la arquitectura *sólo-servidor* el cliente sólo ha de adquirir y transmitir voz, realizándose todo el procesamiento en el servidor. La arquitectura *cliente-servidor* en cambio, distribuye la extracción de parámetros o características relevantes de la voz al cliente y el proceso de clasificación, o reconocimiento propiamente dicho, al servidor.

Aunque no forme parte del paradigma del reconocimiento remoto, se puede considerar el reconocimiento tradicional de voz como una posible arquitectura *solo-cliente*. En ella, todo el proceso de reconocimiento se realiza en el dispositivo del cliente, mientras que el servidor, a partir del texto reconocido por el cliente, realiza el acceso y servicio de información requerida.

3.4.1. Arquitectura Sólo-Servidor

En la arquitectura sólo-servidor, la única tarea relacionada con el reconocimiento que realiza el cliente es la adquisición de la voz del usuario. El resto de tareas, incluida la parametrización de la voz, son realizadas por el servidor. Por esta razón, la arquitectura sólo-servidor es también conocida como reconocimiento remoto basado en red o *NSR* (*Network-Based Speech recognition*), ya que desde el punto de vista del usuario, es la red la que se encarga de realizar todo el proceso de reconocimiento. La figura 3.9 (arriba) muestra la estructura de una arquitectura de este tipo.

La principal virtud de esta arquitectura es que no requiere cambios en la tecnología existente. Si el dispositivo es un terminal móvil, la transmisión de voz es connatural a este, mientras que para redes IP se disponen de estándares y protocolos bien definidos para la

transmisión de voz como *VoIP* (*Voice over IP*), así como de productos comerciales que hacen uso de ellos.

Sin embargo, también plantea algunas desventajas. La voz debe ser codificada antes de ser transmitida, dando lugar a distorsiones, lo cual puede suponer una reducción en la precisión del reconocimiento. A esto debe añadirse la degradación que sufre la voz debido al canal de transmisión, que también repercute negativamente en la precisión de reconocimiento.

3.4.2. Arquitectura Cliente-Servidor

En la arquitectura cliente-servidor, las tareas de procesamiento y computación se hayan distribuidas entre el terminal y el servidor remoto de reconocimiento de voz. De ahí que esta arquitectura sea generalmente conocida como reconocimiento distribuido de la voz o *DSR* (*Distributed Speech Recognition*). En este tipo de arquitecturas se evita transmitir directamente la señal de voz. En vez de esto, en el dispositivo del cliente se lleva a cabo la adquisición de la voz y las tareas propias del bloque de representación, esto es, la extracción de las características. Estas características, y no la voz, son codificadas y transmitidas por un canal de datos (en GSM estos canales ofrecen una mayor protección), con algún tipo de protección adicional. De esta forma se evita la posible degradación por compresión de la voz, a la vez que se realiza una transmisión más robusta frente a ruidos del canal. En el receptor, las características de voz son decodificadas, alimentando al reconocedor de patrones (o reconocedor propiamente dicho). Se distingue así entre un *front-end* que realiza las tareas de representación, y un *back-end* que realiza el reconocimiento. Adicionalmente, esta arquitectura requiere un ancho de banda significativamente menor que la arquitectura NSR. Esto se debe a que es necesario transmitir sólo parámetros relevantes para el reconocimiento. La figura 3.9 (abajo) muestra la estructura de una arquitectura de reconocimiento remoto cliente-servidor.

Como indica Haavisto [73], el principal inconveniente de esta aproximación radica en que es necesario un *front-end* estandarizado común para todas las aplicaciones. Este front-end estándar debe permitir una alta precisión tanto para entornos limpios como ruidosos, así como considerar múltiples lenguajes. Igualmente, debe proporcionar independencia con respecto al equipamiento del terminal. En este sentido, el *Aurora DSR Working Group* ha realizado un importante esfuerzo, desarrollando cuatro front-ends que habilitan el reconocimiento distribuido en GSM, y que actualmente se encuentran estandarizados por ETSI. Estos cuatro estándares son el *Front-End* básico (*FE*) [74], el primero que se presentó;

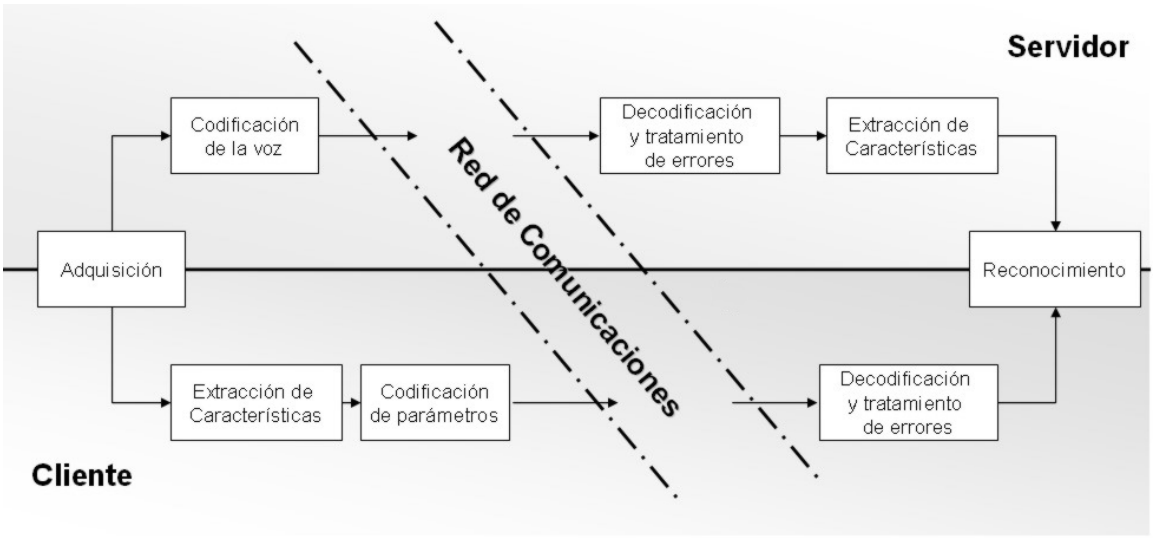


Figura 3.9: Arquitecturas para el reconocimiento remoto: Arquitectura sólo-servidor (arriba), y Arquitectura cliente-servidor (abajo).

el *Advanced Front-End* (*AFE*) [75], que ofrece una mayor precisión en entornos acústicos ruidosos [76]; el *eXtended Front-End* (*XFE*) [77], que incluye parámetros adicionales que permiten el reconocimiento de lenguas tonales y síntesis de voz; y el *eXtended Advanced Front-End* (*XAFE*) [78], que es una combinación de los dos anteriores.

Dado que el tratamiento del ruido acústico queda fuera del interés de este trabajo, así como el reconocimiento de lenguas tonales, o la síntesis posterior de la voz a partir de parámetros de reconocimiento, nos centraremos en el primero de ellos, el FE, al que usualmente nos referiremos como estándar DSR. A continuación, describiremos brevemente este estándar.

El estándar DSR

El estándar de ETSI fue concebido para posibilitar la arquitectura DSR sobre telefonía móvil, en donde ofrece muy buenos resultados con entornos acústicamente limpios [51]. Sin embargo, su uso no ha quedado limitado a este tipo de redes, extendiéndose posteriormente a IP. Este estándar está basado en la representación por Mel-Cepstrum (sección 2.2.2) dada su amplia utilización en la industria del reconocimiento de voz.

La extracción de características que realiza este estándar coincide con aquella descrita para nuestro sistema de referencia (consultar sección 2.5 para más detalles), con la salvedad de que el MFCC de orden 0 también se calcula. Para reducir la tasa de bits requerida durante la transmisión, sólo se transmiten las características estáticas (las dinámicas se

calculan a partir de éstas en el servidor) comprimidas mediante una cuantización *SVQ* (*Split Vector Quantisation*). Seis diccionarios de 64 centros (6 bits) se emplean para cuantizar cada par de coeficientes cepstrales desde el de orden 1 al de orden 12, mientras que el coeficiente cepstral de orden 0 y el logaritmo de la energía se cuantizan mediante un diccionario de 256 centros (8 bits).

Los vectores de características codificados son agrupados en parejas denominadas *frame pairs* (FP), formados por dos vectores a los que se les añade un código de redundancia cíclica de 4 bits. La tasa de bits requerida asciende a tan solo 4800 bits por segundo de voz. En el receptor se aplican dos test para la detección de errores: la comprobación del CRC y un *test de consistencia* de datos basado en la continuidad de estos. Mediante el CRC se verifica que los FP hayan sido recibidos sin errores. Adicionalmente, si el CRC del FP actual es erróneo, se aplica el test de consistencia sobre el FP anterior. Si éste falla, la trama anterior también se considera errónea (aunque su CRC sea correcto). A partir de este punto todas las tramas recibidas se marcan como erróneas hasta que se reciba un FP que supere tanto el CRC como el test de consistencia. De esta forma se obtiene un mecanismo efectivo para la detección de errores en canales de radio. Un algoritmo de mitigación se encarga de las tramas erróneas. Éste consiste en la sustitución de los vectores erróneos por aquellos correctamente recibidos más cercanos. Es decir, supuesta una ráfaga de $2B$ vectores, los B primeros vectores son reemplazados por el último vector correcto antes de la ráfaga, mientras que los B últimos son reemplazados por el primer vector correctamente recibido tras ella.

Este estándar se mantiene prácticamente idéntico en redes IP, en donde queda especificado mediante el formato de carga útil DSR para el protocolo RTP [79]. La figura 3.10 muestra un diagrama de la estructura de este formato. Al igual que antes, los vectores de características codificados son agrupados de dos en dos y se les añade también la protección por CRC. Este CRC se mantiene por razones de compatibilidad ya que, como vimos en la sección 3.3.3, desde capas más bajas ya se introducen protecciones que aseguran la integridad de los datos. Dado que la longitud original de un FP es de 92 bits y ésta cantidad no puede descomponerse de forma entera en octetos de 8 bits, se le añade un relleno de cuatro bits todos iguales a cero. Así los FP tienen una longitud igual a 12 octetos (96 bits). En el receptor se identifican los paquetes perdidos gracias al campo de numeración incluido en el protocolo RTP. Cuando se detecta una ráfaga de vectores perdidos se aplica el mismo algoritmo de mitigación descrito en el párrafo anterior.

El número de FPs por paquete IP constituye un punto conflictivo. Por un lado, enviar pocas parejas de vectores por paquete supone un desperdicio de ancho de banda, debido a

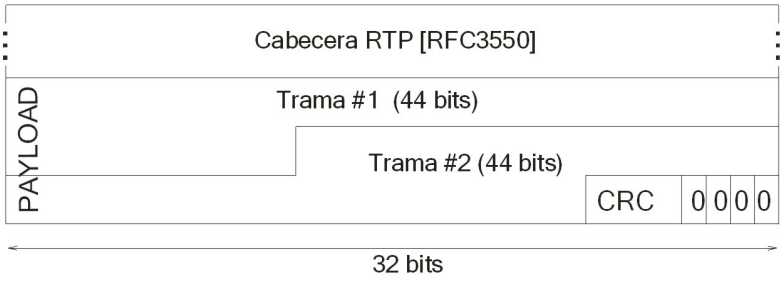


Figura 3.10: Formato de carga útil DSR para el protocolo RTP.

la sobrecarga originada por las cabeceras (RTP/UDP/IP). Por otro lado, usar un número alto de FPs por paquete incrementa el retardo de extremo a extremo, incrementando igualmente la latencia del reconocimiento. Además, los efectos provocados por la pérdida de un paquete serán mayores cuantos más FP transporte. Por esta razón, aunque el número de FP por paquete no está especificado, se recomienda que sea minimizado (generalmente un FP por paquete), debiendo ser determinado por la latencia y los requerimientos de ancho de banda de la aplicación. Asimismo, se recomienda el uso de técnicas de compresión de la cabecera para mejorar la eficiencia de uso del ancho de banda [80, 81].

3.5. Redes y Arquitecturas. Soluciones disponibles

Como hemos visto anteriormente, existen dos arquitecturas posibles para el reconocimiento remoto, la arquitectura NSR y la arquitectura DSR. Ambas arquitecturas son igualmente válidas, sin embargo, presentan tanto ventajas como inconvenientes que deciden positiva o negativamente su viabilidad y conveniencia en un tipo u otro de red. En este apartado, examinaremos las combinaciones red-arquitectura posibles destacando la conveniencia o no de cada una de ellas.

NSR sobre GSM

A corto plazo, una de las arquitecturas disponibles para habilitar el reconocimiento de voz sobre dispositivos móviles es aquella basada solo en servidor o NSR. La mayor dificultad de esta arquitectura estriba en la reducción del rendimiento del sistema de reconocimiento debido a la distorsión producida por los errores de transmisión y el proceso de codificación y decodificación de la voz.

El limitado ancho de banda disponible en GSM plantea un problema de cara al reconocimiento. Por un lado, una transmisión de voz sin codificar requiere una tasa de bits

inaceptable para un enlace móvil. Por otro, una codificación con una tasa de bits muy pequeña produce distorsiones en la voz que reducen la precisión en el reconocimiento. Diversos estudios cuantifican los efectos de la codificación de la voz sobre el reconocimiento de voz [73, 82, 83, 84] usando distintos codificadores. Para evitar esta degradación algunos trabajos proponen realizar el reconocimiento no a partir de la voz sintetizada, sino a partir de los parámetros con los que ésta es codificada [85, 86].

Sin embargo, la necesidad de codificar la voz no es la principal causa de la reducción de la precisión, se unen además los problemas inherentes a la transmisión. Más adelante en este mismo capítulo, se muestra un estudio de la influencia sobre la precisión en el reconocimiento que tienen los errores del canal al transmitir sobre GSM. Como se podrá ver, la mayor causa de distorsión y reducción de la precisión no es debida a la codificación de la voz, sino precisamente, a los errores del canal.

A pesar de estos problemas, el reconocimiento distribuido basado en red sigue siendo muy atractivo para la telefonía móvil ya que usa la tecnología disponible actualmente sin requerir ningún cambio en los terminales móviles.

DSR sobre GSM

Como alternativa, el reconocimiento distribuido basado en arquitecturas cliente-servidor sobre GSM se presenta prometedor. La mayor protección aplicada en los canales de tráfico para datos, hacen esta alternativa más robusta a los problemas del canal. Adicionalmente, el ancho de banda necesario para la transmisión de los parámetros de voz es menor.

Aunque el problema del diseño de un front-end común para todos los dispositivos está prácticamente resuelto, aún quedan por definir los protocolos de alto nivel que regulen la interacción entre las aplicaciones y el usuario. Por otra parte, existen problemas inherentes a esta arquitectura que dificultan su implantación en el mercado GSM. La necesidad de hardware nuevo que realice la extracción de características en los dispositivos móviles implica que ningún terminal actual esté preparado para hacer uso de esta tecnología.

Por otro lado, las implicaciones con respecto a la seguridad y verificación de las transacciones es un inconveniente de los estándares FE y AFE para DSR. Dado que los parámetros que se transmiten son aquellos relevantes de cara al reconocimiento, las características referentes a la identificación del locutor se eliminan (carecen de interés para el reconocimiento). Esto imposibilita la verificación de una transacción en el caso de que un usuario la repudiara, lo que supone un obstáculo en el uso de esta tecnología en entornos bancarios

o bursátiles. Igualmente, si se quiere diseñar un front-end global, éste no puede obviar la existencia de lenguas tonales, en donde el pitch juega un papel fundamental. En esta línea, los estándares XFE y XAFE proponen la inclusión de algunas características de identificación del locutor, entre las que se incluye el pitch. Esto permite reconstruir la voz con suficiente calidad como para ser identificada [87, 88, 89], así como el reconocimiento de lenguas tonales [77]. Pero en este caso, los parámetros adicionales deberán ser robustos frente a las degradaciones del canal, lo que nos remite a los diseños de codificadores de voz estándar. Es decir, degeneramos hacia un codec de voz de baja calidad y menor bitrate orientado, en vez de a transmisión, a reconocimiento de voz.

NSR sobre IP

Esta arquitectura presenta ventajas e inconvenientes similares a NSR sobre GSM. En este caso, la transmisión de la voz se podría realizar reutilizando la tecnología existente, como por ejemplo VoIP (*Voice over IP*). Tanto investigadores como desarrolladores han realizado múltiples esfuerzos por integrar el reconocimiento de voz en VoIP. Sin embargo, los trabajos realizados en este tema arrojan una reducción de la precisión del reconocimiento inadmisible si éste se realiza sobre voz transmitida con VoIP. Esta reducción se debe principalmente a la distorsión introducida por el codec y la pérdida de tramas, frecuente en las transmisiones reales a través de redes IP con restricciones de tiempo [90].

DSR sobre IP

Trabajos recientes proponen el uso del estándar DSR para telefonía móvil [74] en redes IP [91, 92]. El reconocimiento distribuido de voz a través de una arquitectura basada cliente-servidor se introduce de forma natural en las redes IP, pues dicha arquitectura es empleada por tantos otros servicios ya disponibles. A diferencia de GSM, el entorno de redes IP se presenta muy propicio para el uso de DSR por una importante característica diferenciadora: la flexibilidad.

Las redes IP fueron diseñadas para el transporte de cualquier tipo de medio, ya sea texto, imágenes, video, voz, parámetros de reconocimiento o una combinación de todos ellos. Una de las mayores ventajas es que, a diferencia de las redes GSM, no es necesario realizar cambios en el hardware de las máquinas conectadas a la red, ya que es posible realizar las tareas del front-end mediante la implementación del software correspondiente.

Sin embargo, si las tecnologías de transmisión de voz sobre redes IP están aún en desarrollo, conformando todo un campo de estudio, las referidas al reconocimiento de

voz están prácticamente en gestación. Actualmente, dos grupos en la IETF (*Internet Engineering Task Force*) trabajan sobre el reconocimiento distribuido sobre redes IP. El primero de ellos, el grupo de trabajo de control de servicios de voz, tiene como objetivo el desarrollo de protocolos para dar soporte al procesamiento distribuido de flujo de voz, concretamente al RAH, conversión de texto a voz y verificación del locutor. Relativamente nuevo, este grupo espera trabajar en coordinación con el grupo de trabajo del *W3C Multimodal Interaction* y con el ETSI Aurora STQ, entre otros. El propósito del grupo se centra únicamente en el control distribuido seguro de estos servicios, no en el transporte de los datos para el reconocimiento. A esto último se dedica el segundo grupo, el grupo de trabajo sobre transporte de vídeo y audio. En Octubre del 2002, este grupo introdujo el estándar en el que se especifica la transmisión con requerimientos de tiempo real para reconocimiento distribuido de la voz (RFC 3557 [79]). Como comentamos antes, este estándar está basado en el propuesto por ETSI, y constituye una apuesta clara por el desarrollo del reconocimiento basado en arquitectura cliente-servidor en redes IP.

Supuesta una red de comunicaciones GSM, la selección de la mejor arquitectura no es ni mucho menos sencilla. El punto conflictivo lo constituye el extractor de características situado en el terminal móvil. Al estar desarrollado específicamente para el reconocimiento, es obvio que la solución presenta mayor robustez frente a los problemas del canal, redundando en una mayor precisión del reconocimiento. Sin embargo, éste implica la necesidad de un hardware especifico que ningún terminal móvil posee en la actualidad. Ello explica que no se espere la implantación de DSR sobre GSM hasta que no se extienda definitivamente UMTS junto con los móviles de tercera generación, en los que, además, está prevista la inclusión por defecto del codec de voz AMR. Los estudios realizados sobre este codec, que también se incluye opcionalmente en los móviles de segunda generación, arrojan porcentajes de precisión en el reconocimiento de voz muy cercanos a DSR [93], con la ventaja añadida de que al tratarse de una arquitectura NSR, se dispone de la señal de voz completa (aunque codificada), siendo, por tanto, más versátil. Esto justifica el interés, tanto de la comunidad científica como de los proveedores de servicios móviles, por el estudio de las dificultades, y propuesta de soluciones para ellas, que conlleva el reconocimiento remoto en una arquitectura NSR a través de GSM.

En el caso de que se emplee una red IP para establecer el canal de comunicación, la superioridad de la arquitectura DSR sobre la NSR resulta patente. Al eliminar la

imposición de hardware especifico nuevo, ya que este puede implementarse por software sobre la pila de protocolos, junto con la flexibilidad de las redes IP, que permiten la transmisión conjunta con la voz codificada en caso de necesitarse, facilita la implantación inmediata de los servicios activados por reconocimiento remoto. A esto se le añade la flexibilidad a la hora de realizar cambios que mejoren el front-end, ya que la propia red puede usarse para realizar actualizaciones automáticas.

Debido a estas razones, este trabajo se ha centrado en las soluciones NSR sobre GSM y DSR sobre IP. Pese a que estas soluciones parezcan a primera vista muy diferentes, lo cierto es que, como se mostrará al final de este capítulo, presentan una problemática similar.

3.6. Problemas del Reconocimiento de Voz Codificada sobre redes GSM

Diversos estudios han sido llevados a cabo tanto para determinar la influencia de la codificación de la voz sobre la precisión en el RAH [73, 82], como para mejorar la precisión del reconocimiento en este tipo de situaciones [83, 94, 95]. En líneas generales, predomina la idea de que, puesto que la codificación implica una degradación en la señal de voz, cabe esperar una reducción en la precisión del RAH conforme se reduce la tasa de bits. Sin embargo, dado que la mayoría de codificadores de baja tasa de bits están optimizados en base a criterios perceptuales, no es posible predecir a priori sus efectos en el proceso de reconocimiento.

En este sentido, Hirsch [96] presentó un estudio acerca de la influencia de la codificación sobre el reconocimiento automático de la voz. En dicho estudio, la precisión en el reconocimiento de voz codificada con G.711, FR, HR, EFR y AMR fue comparada con la precisión obtenida con voz sin codificar. Los resultados obtenidos evidencian una reducción no despreciable de la precisión si el reconocedor estaba entrenado con voz sin codificar, sin embargo, un entrenamiento multi-condición, en donde se empleara voz codificada, permitía reducir significativamente estas diferencias. Los resultados de este estudio coinciden con los obtenidos por Kiss en [97], donde se compara el rendimiento de la aproximación NSR empleando el codificador EFR con la aproximación DSR empleando el estándar propuesto por la ETSI. En las conclusiones de este estudio se indica que, aunque la codificación con EFR induce cierta degradación en el reconocimiento de voz limpia, se mantiene un nivel aceptable de rendimiento, el cual puede ser mejorado mediante un

re-entrenamiento de los modelos con voz codificada con EFR. Además, para ruidos estacionarios se observó un mayor rendimiento cuando la voz estaba procesada con EFR.

La segunda causa de degradación del reconocimiento viene dada por la distorsión introducida por el canal GSM. Esta degradación suele ser la más significativa [51]. Así, en el mismo trabajo de Kiss se indica que el rendimiento de la solución DSR permanece prácticamente inalterable excepto en condiciones extremas de operación (fuera de la celda de servicio), mientras que EFR sufre una caída continua conforme se degrada el canal, manteniéndose siempre por debajo del rendimiento de DSR.

Para el presente trabajo, hemos realizado un estudio acerca de la influencia del canal GSM en el reconocimiento de voz. El objetivo de estudio consiste en comprobar la importancia de la degradación debido a los problemas del canal, así como identificar sus causas. Éste se centra en el codec EFR, justificándose por las siguientes razones:

- El codec EFR es el estándar implementado mayoritariamente para la segunda generación de telefonía móvil, principalmente porque proporciona la mayor calidad de voz de entre todos los disponibles para GSM. De hecho, el codec más avanzado, AMR, emplea EFR para el modo de mayor calidad.

- A diferencia de AMR, la codificación de la voz en EFR es independiente del estado del canal. De esta forma es más sencillo aislar y cuantificar la reducción de precisión debida únicamente a la degradación del canal.

- Salvo ciertas consideraciones con respecto a AMR, las conclusiones obtenidas para EFR pueden ser extrapolables al resto de codecs. Todos los codecs GSM (incluidos los distintos modos de AMR) se basan en los mismos principios de operación: son codificadores de análisis por síntesis (descritos en la sección 4.2) cuyos parámetros se protegen previamente a la transmisión en base a criterios perceptuales.

- Que la protección del canal sea más robusta en AMR, no implica que sea completamente inmune a los problemas que se describen en este estudio. Aunque en menor medida, estos pueden aparecer.

Los resultados obtenidos a partir de este estudio resultaran muy útiles para el desarrollo posterior de técnicas encaminadas a la mitigación de errores ocasionados por el canal GSM.

3.6.1. Identificación de las distorsiones del canal

Durante la decodificación de voz en GSM existen dos tipos de procesamiento bien diferenciados para las tramas de voz recibidas. En base a los mecanismos de protección incluidos durante la codificación del canal, las tramas se clasifican como correctas e incorrectas. Las tramas marcadas como correctas se decodifican tal cual, mientras que las tramas incorrectas se sustituyen a través de un algoritmo de mitigación que consiste, básicamente, en un apagado gradual de la señal.

El hecho de que una trama sea clasificada como correcta no implica que los bits que la codifican se correspondan exactamente con los bits enviados desde el emisor. Esto se debe a que no todos los bits están protegidos, ya que aquellos perceptualmente poco importantes quedan excluidos de las distintas protecciones. En este sentido, la señal sintetizada puede presentar anomalías que, aunque sean vagamente percibidas por un oyente humano, tengan relevancia durante el reconocimiento. Por lo que respecta a las tramas clasificadas como incorrectas, éstas son directamente excluidas del proceso de síntesis. Mediante una extrapolación de tramas anteriores se realiza un apagado gradual de la señal que resulta más confortable para el oyente que la síntesis de datos incorrectos, pero que puede inducir a errores durante la etapa de reconocimiento. Así pues, se pueden distinguir tres tipos de tramas recibidas en el receptor: 1) tramas estrictamente correctas, idénticas a las emitidas por el emisor, 2) tramas perceptualmente correctas, que difieren de las emitidas pero sin cambios en bits significativos, y 3) tramas incorrectas, que difieren de las emitidas con alteraciones en bits significativos.

Para el estudio de los errores de transmisión y de su influencia en el RAH se pueden utilizar los patrones de error para GSM, EP1, EP2 y EP3. Estos patrones, comúnmente empleados en el desarrollo y evaluación de técnicas de mitigación de errores en canales GSM, han sido generados a partir de transmisiones reales de radio GSM realizadas en diferentes condiciones sucesivamente más desfavorables. El patrón EP1 tiene una relación entre la portadora de señal y las interferencias (C/I) de 10dB y se corresponde con operaciones realizadas dentro de la celda GSM. El EP2 con una relación de 7dB C/I representa operaciones en los límites de la celda, mientras que el EP3 con 4dB C/I representa situaciones en las que se opera más allá de los límites de la celda GSM.

Aplicando estos patrones durante la transmisión y observando en el receptor los bits de los parámetros que codifican la voz en cada trama, se puede obtener, tanto para tramas clasificadas correctas como para tramas incorrectas, la tasa de errores de bit (*BER–Bit Error Rate*) para cada uno de los patrones de error. La tabla 3.1 muestra estos resultados

Trama	Canal	BER
Correcta (BFI=0)	EP1	1.31 %
	EP2	2.15 %
	EP3	3.36 %
Incorrecta (BFI=1)	EP1	14.25 %
	EP2	14.81 %
	EP3	19.06 %

Tabla 3.1: Tasa de errores de bit según la clasificación de la trama para cada condición de canal.

para cada patrón considerando sólo tramas correctas y sólo tramas incorrectas. Puede observarse un incremento del BER del orden de 6 a 10 veces en tramas incorrectas con respecto a tramas correctas.

Un aspecto relevante de las tramas incorrectas es que suelen aparecer de forma consecutiva. Si atendemos a las características del enlace de radio, se puede establecer cierta relación entre los desvanecimientos Rayleigh y la aparición de estas tramas. Estas interferencias tienen capacidad suficiente como para romper la protección del código convolucional y del entrelazado, afectando a bits relevantes de la trama. De ser así, generalmente se verán implicadas más de una trama, dando lugar a una ráfaga de tramas incorrectas.

A través de los patrones de error descritos anteriormente podemos establecer una estadística acerca de la frecuencia de estas ráfagas según su longitud (denominada longitud de ráfaga), es decir, según el número de tramas incorrectas consecutivas. En la figura 3.11 se muestra un histograma con la frecuencia relativa de aparición de ráfagas por longitud. La caída del número de ráfagas conforme se incrementa su longitud viene a ser aproximadamente exponencial.

Se puede concluir entonces que la degradación conjunta del canal se debe a dos tipos de errores: Errores de *background*, que alteran bits perceptualmente poco importantes y de forma aislada, y errores *en ráfagas* o *de ráfaga*, mucho más fuertes, que alteran bits importantes provocando la pérdida de una o varias tramas completas. Cada uno de ellos afectará en diferente medida la precisión en el reconocimiento.

3.6.2. Impacto de los errores del canal en el reconocimiento

Los dos errores descritos anteriormente, tanto por su naturaleza como por su frecuencia de aparición afectan de forma distinta al reconocimiento. Para cuantificar los efectos tanto

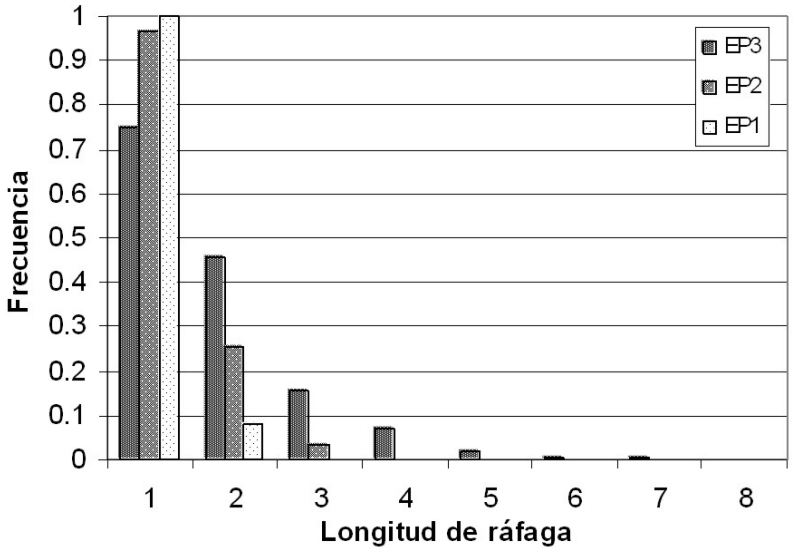


Figura 3.11: Distribución según longitud de ráfaga, en número de tramas consecutivas, para cada condición de canal.

del ruido de fondo como del ruido de ráfaga se ha recurrido a pruebas de reconocimiento en las que se aisla cada uno de ellos.

El marco experimental de estas pruebas incluye el sistema RAH de referencia propuesto en la sección 2.5. A este sistema se le han añadido los bloques funcionales necesarios para simular la transmisión de voz por una red GSM empleado el codec EFR. Para ello, la voz es codificada y decodificada mediante el estándar EFR y transmitida empleando el canal de tráfico TCH/EFS. Para caracterizar el comportamiento del canal, se han tomado los patrones de error EP1, EP2 y EP3. La figura 3.12 muestra un diagrama del sistema.

El bloque *"Test A"* será el bloque funcional que nos permita aislar los dos tipos de

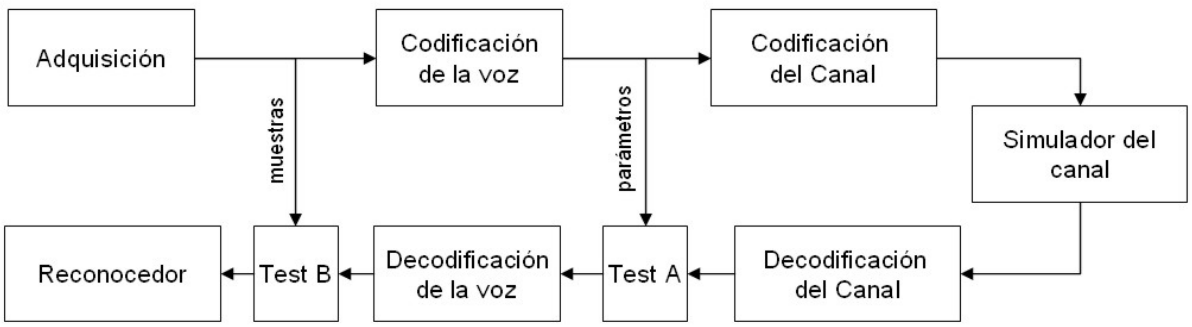


Figura 3.12: Diagrama funcional propuesto para evaluar el impacto de los errores del canal en el reconocimiento en una arquitectura NSR sobre EFR.

Canal	DSR	EFR	Errores Background	Errores de ráfaga	Ruido de Memoria	Ruido de sust. y apagado
Limpio	99.04	98.70	–	–	–	–
EP1	99.04	98.44	98.50	98.61	98.44	98.68
EP2	98.95	96.91	98.31	97.53	97.73	98.28
EP3	93.41	84.48	98.22	85.80	93.54	90.47

Tabla 3.2: Precisión del reconocimiento para DSR y EFR junto con diferentes efectos aislados, para distintas condiciones de canal.

ruidos (por el momento, el bloque *"Test B"* no hará nada), permitiendo los siguientes experimentos:

- Reconocimiento sobre EFR. La señal atraviesa un canal ruidoso modelado por los patrones de error EP1, EP2 y EP3. No se aplica ningún tipo de procesamiento adicional.

- Errores de Background. En este experimentos sólo consideramos el ruido generado por las tramas no marcadas ($BFI = 0$). Las tramas marcadas con $BFI = 1$ son reemplazadas en el bloque *"Test A"* por la correspondiente trama correcta (con los parámetros de codificación correspondientes a una transmisión limpia). De esta forma estamos únicamente considerando el efecto de los errores de background.

- Errores de ráfaga. En este caso sólo tenemos en cuenta el ruido debido a las tramas con BFI activo (tramas erróneas). Las tramas no marcadas ($BFI = 0$) son reemplazadas por la correspondiente trama correcta, lo que nos asegura que no hay ningún bit modificado. Así estamos evaluando únicamente el efecto de los errores de ráfaga.

La tabla 3.2 (3ª, 4ª y 5ª columna) muestra los resultados de estos experimentos bajo diferentes condiciones de canal. Puede observarse que los errores de ráfaga constituyen la principal fuente de reducción del rendimiento. Esto resulta coherente con el procesamiento que tienen, por parte del decodificador EFR, las tramas marcadas. Por un lado, la información que una trama marcada pudiera contener se deshecha al ser sustituida. Por otro, el apagado sucesivo puede derivar a ruido de confort. Estos silencios artificiales dan lugar, además de las correspondientes sustituciones, a inserciones durante el reconocimiento. Igualmente cabe destacar el efecto despreciable de los errores de background, lo que sugiere que el criterio basado en la calidad subjetiva empleado para la protección de bits podría ser también adecuado para el reconocimiento de voz.

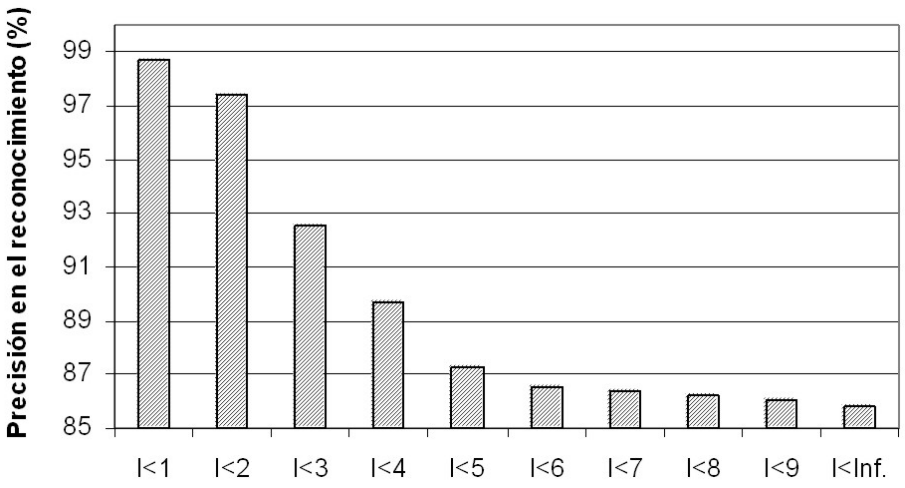


Figura 3.13: Influencia de la longitud de ráfaga (l) sobre el reconocimiento cuando se reconoce voz codificada con EFR con la condición de canal EP3.

Claramente los efectos de los errores de ráfaga serán dependientes de la longitud de las ráfagas en la transmisión. Sin embargo, no es posible predecir el comportamiento debido a dos factores: 1) la frecuencia de aparición de la ráfaga, y 2) la capacidad de destrucción de la ráfaga. Es de suponer que las ráfagas de menor longitud tiene un efecto menos importante en el reconocimiento, sin embargo, son más frecuentes. Por otro lado, las ráfagas de mayor longitud tendrán un efecto importante en el reconocimiento, pero su frecuencia disminuye exponencialmente con la longitud (figura 3.11). En la figura 3.13 se analiza el efecto de la longitud de la ráfaga l sobre el rendimiento del reconocedor (las ráfagas han sido extraídas de la condición de canal EP3). Cada etiqueta $l < n$ indica que sólo aquellas ráfagas con una longitud menor a n son considerados (las ráfagas más largas son reemplazadas por las correspondientes tramas correctas). Así, $l < 1$ coincide con una transmisión limpia y $l < \infty$ con la condición EP3. Puede observase que aunque las ráfagas largas son más dañinas para el reconocimiento, el efecto de ráfagas mayores a 5 tramas es casi despreciable, dado que éstas son muy escasas incluso en el peor condición (EP3).

Efecto de la Memoria del Codec

Analizando la señal de voz obtenida en una transmisión ruidosa, puede observarse una degradación de la señal correspondiente a tramas correctamente recibidas tras un error de ráfaga. La figura 3.14 muestra las formas de onda de una señal transmitida por un canal limpio y una señal transmitida con dos ráfagas de longitud 5 que comienzan en la trama 20 y 35 respectivamente, ambas codificadas con EFR. Puede observarse que aunque las

tramas 25 y 40 se reciben sin errores, la forma de onda sigue alterada durante algunas tramas posteriores a la ráfaga. En la figura también se muestra la evolución de la *relación señal-ruido* (*SNR–Signal to Noise Ratio*), medida entre la señal original y el ruido de codificación, de una señal recibida con errores y de otra correctamente transmitida. Estas SNRs se ha obtenido a intervalos no solapados de 160 muestras (el tamaño de una trama EFR). Puede observarse una divergencia entre ambas SNRs en el momento en el que comienzan las ráfagas de error. Sin embargo, la convergencia no se produce en el instante en el que termina la ráfaga (5 tramas después). En la primera ráfaga se necesitan unas 6 tramas tras la ráfaga para alcanzar la convergencia, mientras que en la segunda son necesarias unas 8. La figura 3.15 muestra los espectrogramas para ambas señales. Estos espectrogramas se han obtenido con ventanas de 160 muestras solapadas 80 muestras. Aunque de una forma mucho menos acusada, se puede apreciar cierta alteración de los formantes.

Los codificadores empleados en GSM incluyen, entre otros, un filtro de largo retardo que se aplica sobre la señal de excitación y que tiene por objeto modelar el pitch de la voz. El uso de este filtro implica que para la decodificación de la trama actual sea necesaria la trama anterior. Por ello, la influencia de un error durante una ráfaga puede prolongarse más allá de los límites de la propia ráfaga. Esto es, supongamos que existe una ráfaga de B tramas que va desde la trama F_{t+1} a la trama F_{t+B} (ambas incluidas), puesto que la trama F_{t+B+1} se sintetiza empleando muestras de la señal de la trama F_{t+B}, la señal sintetizada sufre cierta alteración aunque los parámetros que la representan se transmitan intactos. Igualmente, la siguiente trama F_{t+B+2}, al depender su síntesis de las muestras de la trama anterior F_{t+B+1}, también se puede ver alterada. Este esquema se repite sucesivamente hasta que finalmente los errores son absorbidos por el sistema. Esta degradación es inherente a la naturaleza predictiva del proceso de codificación y nos referiremos a ella como *ruido de memoria*.

Empleando el marco anterior de experimentación (figura 3.12), podemos aislar los efectos de la sustitución y apagado durante un error de ráfaga, y el ruido de memoria tras la ráfaga, estudiando el impacto de cada tipo de ruido sobre la precisión en el reconocimiento. Puesto que lo que se altera es la señal, no se reemplazaran los parámetros de las tramas, sino que se actúa directamente sobre las muestras de la señal. En este caso haremos uso del bloque funcional *"Test B"* en la figura 3.12. Las pruebas son las siguientes:

- Ruido de sustitución y apagado. Las muestras de voz pertenecientes a las tramas correctamente recibidas ($BFI = 0$) son reemplazadas por las correspondientes mues-

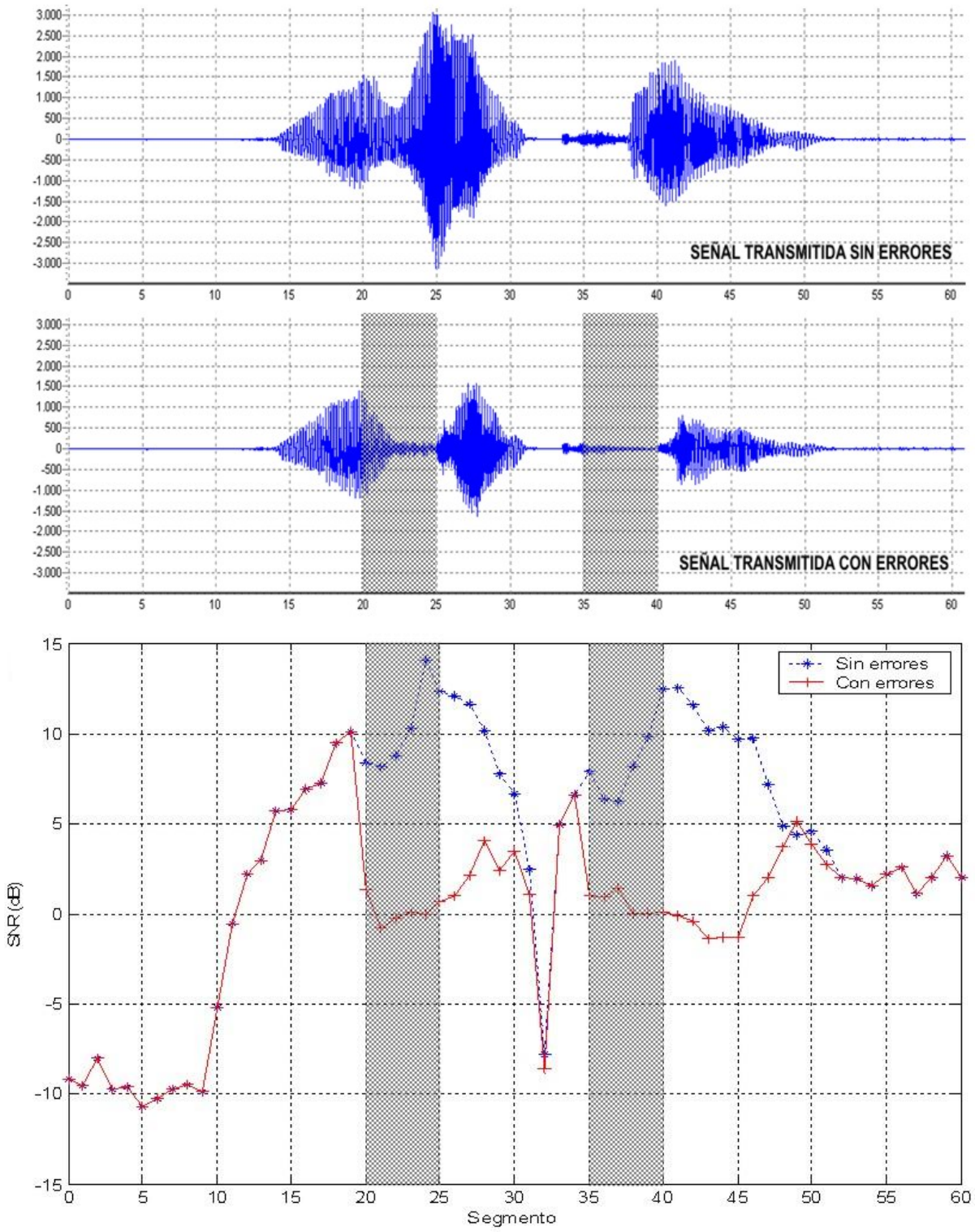


Figura 3.14: Forma de onda de una señal de voz transmitida por EFR sin y con errores de ráfaga (arriba) y SNR de codificación por tramas de ambas señales (abajo).

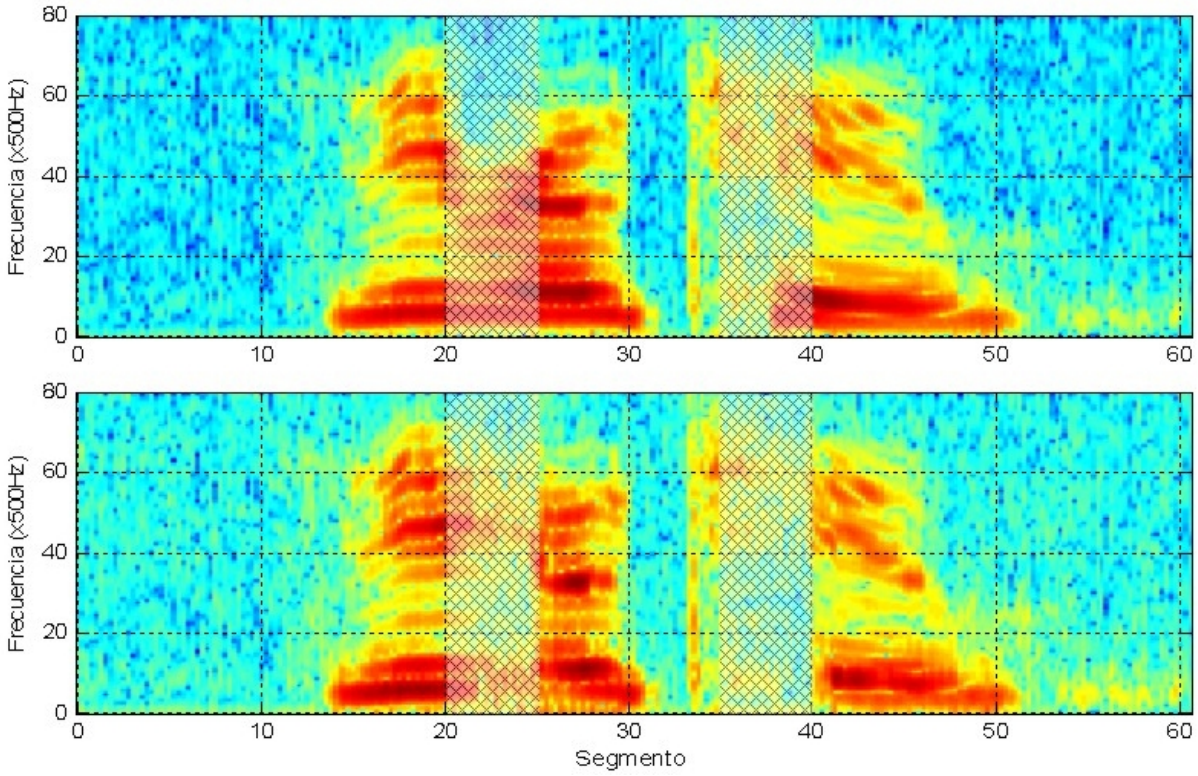


Figura 3.15: Espectrograma de una señal de una señal de voz transmitida por EFR sin y con errores de ráfaga.

tras correctas. De esta forma, las únicas muestras ruidosas que permanecen en la señal sintetizada son aquellas donde se aplica el algoritmo de sustitución y apagado.

- Ruido de memoria. Las muestras pertenecientes a las tramas erróneas ($BFI = 1$) son reemplazadas por sus correspondientes muestras correctas. Previamente los errores de background han sido eliminados, de tal forma que la única degradación presente es el ruido de memoria.

En la tabla 3.2 (columnas 6ª y 7ª) se muestran los resultados de estas pruebas. Se puede observar que el ruido de memoria que aparece en las tramas correctamente recibidas después de cada ráfaga es también una fuente importante de degradación. Igualmente, confirma que los errores de ráfaga extienden su influencia más allá de los límites de la ráfaga, degradando el reconocimiento.

3.7. Problemas del Reconocimiento Distribuido sobre redes IP

Al realizar un procesamiento distribuido de la voz, evitando así la distorsión debida la codificación, la única dificultad de la arquitectura DSR sobre redes IP viene dada por el propio canal de comunicaciones. En un principio, podría afirmase que las transmisiones realizadas mediante una red IP están exentas de ruido de canal, en el sentido de que los bits entregados en el receptor son idénticos a los bits enviados por el emisor. Esto se debe a que los paquetes IP incluyen protecciones CRC destinadas a identificar la presencia de bits modificados. Ya en el protocolo de la capa más baja (UDP) si un paquete se recibe con bits alterados éste se descarta y no se pasa a la capa superior para que sea procesado. De esta forma, si un paquete finalmente llega a la capa de aplicación en el receptor, puede asegurarse la integridad de la información que contiene. A esta protección ofrecida por los protocolos se les une la bondad de las redes subyacentes que componen Internet. Dado que múltiples redes colaboran en la transmisión de un paquete, si todas ellas se despreocuparan de la fiabilidad de los datos que transmiten, la calidad de la transmisión sería muy pobre, ya que un paquete puede atravesar fácilmente decenas de redes. En su mayoría, las redes subyacentes trabajan sobre enlaces por cable con una alta relación señal-ruido, donde la probabilidad de que un bit se altere es muy baja. Aquellas redes como las inalámbricas o los enlaces por satélite, donde se trabaja con SNRs más bajas, incluyen mecanismos de protección que aseguran la calidad de la transmisión. Esto es, ya en la red subyacente, gestionada por la capa de interconexión, se aplican las protecciones necesarias para que la transmisión de un paquete desde una máquina a otra sea lo más fiable posible.

En este sentido, cabría esperar unos excelentes resultados para el reconocimiento distribuido sobre IP, puesto que no hay distorsión por codificación de la voz y disponemos de un canal prácticamente sin ruido. Sin embargo, la propia naturaleza de las redes IP impone otra serie de obstáculos de cara a la transmisión de información con restricciones temporales en la entrega, mencionados anteriormente:

- Latencia o retardo extremo a extremo, esto es, la diferencia de tiempo entre el instante en que es recibido un paquete y el instante en el que fue emitido. La latencia debe quedar circunscrita a las restricciones de tiempo real del servicio, a lo que contribuye negativamente la latencia propia de cada red, así como los retardos que sufren los paquetes en cada router a la espera de ser encaminados.

- Dispersión temporal del retardo o *jitter*. Cada paquete puede seguir una ruta distinta y por lo tanto, su retardo puede ser diferente, planteando un serio problema ya que se hace muy complejo predecir la latencia.

- Pérdida de paquetes. Como se indicó en apartados anteriores, además de los fallos ocasionales que pudieran ocurrir en la red, los paquetes son descartados en situaciones de congestión. Igualmente, si un paquete presenta un retardo superior a las restricciones temporales en la entrega, pasa a considerarse perdido.

El primer obstáculo tiene una menor importancia en el reconocimiento distribuido de la voz, ya que en éste no se espera una respuesta inmediata del servidor, aunque sería deseable. Por ello, es posible emplear memorias intermedias de mayor tamaño que permitan absorber además la dispersión temporal. En última instancia, si debido al jitter un paquete llega fuera de la latencia máxima permitida, éste pasaría a considerarse perdido. Así, es el último problema, la pérdida de paquetes, el de mayor relevancia en el reconocimiento distribuido sobre IP, ya que se produce una pérdida de información que afecta al reconocimiento.

3.7.1. Modelado de las pérdidas de paquetes

Desde principios de los 70, se llevaron a cabo aproximaciones experimentales y mediciones sistemáticas de los retrasos y pérdidas de paquetes que se producían en ARPANET. En estas mediciones se examinaba las variaciones en el retraso de los paquetes para diferentes caminos, horas del día, días de la semana, etc. Los resultados obtenidos en aquellos trabajos se emplearon posteriormente para ajustar los parámetros de retransmisión de paquetes en el protocolo TCP. El uso de Internet para aplicaciones de tiempo real ha abierto de nuevo el interés por evaluar y cuantificar el desempeño de la red para este tipo de tareas, atendiendo principalmente a los retrasos y a la pérdida de paquetes.

En 1993, Bolot examinó una conexión desde Francia a los EE.UU. con el fin de analizar el comportamiento de las pérdidas y retrasos de paquetes de extremo a extremo, midiendo el tiempo de ida y vuelta (*RTT - Round Trip Time*) de paquetes UDP enviados a intervalos regulares [98]. Los resultados obtenidos a través de la medición resultaron consistentes con la hipótesis de que la carga de Internet viene dada por una mezcla de tráfico "pesado" formado por paquetes de gran tamaño y tráfico "interactivo" constituido por paquetes de reducido tamaño. Para modelar los resultados en cuanto a pérdidas y retrasos observados en el tráfico analizado, Bolot presentó un modelo sencillo que preservaba las

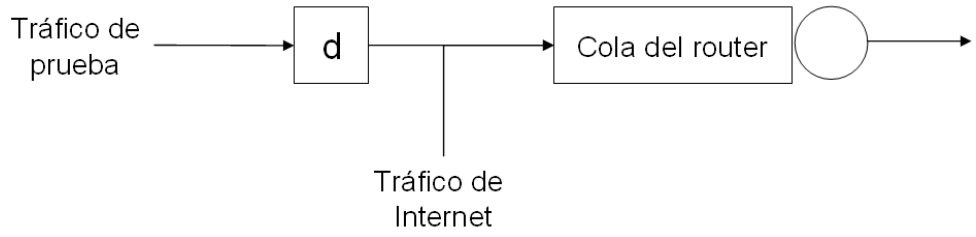


Figura 3.16: Modelo empleado por Bolot para simular retardos y pérdidas de paquetes.

características esenciales de sus experimentos (figura 3.16). En este modelo un servidor procesa a una determinada velocidad los paquetes almacenados en una cola que recibe dos flujos de entrada, considerándose fragmentos discretos de tiempo o *slots*. El primer flujo, denominado de prueba, genera paquetes de tamaño fijo a un ritmo constante, siendo d el tiempo que transcurre entre la emisión de paquetes. El segundo flujo representa el tráfico de Internet, y genera paquetes de acuerdo con una distribución que se puede obtener a través de los retrasos medidos de extremo a extremo en una conexión.

Los resultados obtenidos en el estudio de Bolot muestran que la probabilidad de pérdida de un determinado paquete aumenta conforme se incrementa el ritmo de generación de paquetes. Esta probabilidad se conoce en el ámbito de redes como probabilidad incondicional de pérdida o *ulp*, y se define como,

$$ulp = P(rtt_n = \infty) \tag{3.1}$$

en donde rtt_n es el tiempo de ida y vuelta del paquete correspondiente al slot temporal n. Cuando el ritmo de generación de paquetes es muy bajo (d tiene un valor alto), la contribución del flujo de prueba a la cola del servidor es despreciable, sin embargo, al incrementarse, aumenta la contribución, adquiriendo relevancia en los momentos de saturación. Por otro lado, si un paquete se pierde en un instante t debido a un desbordamiento en la cola, es probable que el siguiente paquete, recibido en el instante $t + d$ también se pierda. La probabilidad de pérdida condicional, *clp*, expresa la probabilidad de que un paquete se pierda dado que el anterior se haya perdido:

$$clp = P(rtt_{n+1} = \infty | rtt_n = \infty) \tag{3.2}$$

Esta probabilidad depende de la frecuencia con la que se generan paquetes de prueba. Si la frecuencia es alta, el tiempo que transcurre entre la emisión de paquetes (d) disminuye. Esto hace más probable que la cola siga congestionada cuando reciba el siguiente paquete,

pues ha pasado poco tiempo. Es decir, la probabilidad de pérdida condicional disminuye conforme crece d y viceversa.

De esta forma, el modelo propuesto por Bolot puso de relevancia la ocurrencia de pérdidas de paquetes de forma consecutiva o en ráfagas, cuyo efecto es más destructivo que la pérdida aislada de paquetes. Más adelante, diferentes autores han ido confirmando esta aparición de pérdidas en forma de ráfagas en las redes IP [99, 100, 101, 102].

Para modelar las pérdidas a través de un modelo matemático se define una *función de indicación de pérdida*. Esta función se describe como sigue:

$$l(n) = \begin{cases} 0 & \text{paquete } n \text{ recibido.} \\ 1 & \text{paquete } n \text{ perdido.} \end{cases} \tag{3.3}$$

A través de la función de indicación de pérdida se puede transformar el flujo emitido en una serie temporal binaria, según sean recibidos o no los paquetes en el emisor, que se suele conocer como *traza*. De esta forma, las pérdidas se definen como un proceso aleatorio caracterizado por medio de series temporales binarias $\{X_t\}_{t=1}^{\infty}$. Yajnik et al. [103] estudiaron la función de autocorrelación de éstas series, concluyendo en que la correlación de las pérdidas en la escala temporal alcanza aproximadamente un segundo, es decir, más allá de este límite las pérdidas de paquetes pueden considerarse independientes entre si. Desde entonces, distintos autores han propuesto diferentes modelos capaces de ajustarse y representar la distribución de paquetes perdidos. A continuación describiremos los modelos más extendidos.

Modelo de Bernoulli

En el modelo de pérdidas de Bernoulli, la secuencia de variables aleatorias se considera independiente e idénticamente distribuida. Esto es, la probabilidad de que una variable X_t sea 0 o 1 es independiente del resto de variables temporales y del índice t. El modelo se caracteriza simplemente por un parámetro r que determina la probabilidad de que X_t sea 1 (correspondiendo a un paquete perdido). Este parámetro se puede estimar a partir de una traza como,

$$\hat{r} = n_1/n \tag{3.4}$$

donde n_1 es el número de veces que el valor 1 aparece en la serie temporal observada y n es el número de muestras de la serie. Es decir, $\hat{r}$ es la tasa media estimada de paquetes perdidos. La distribución de las ráfagas tanto de paquetes recibidos $f(l)$, como de paquetes perdidos $\bar{f}(l)$ viene dada por:

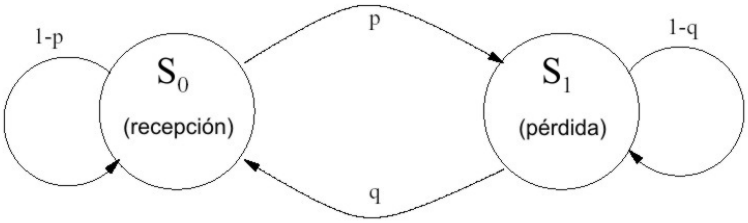


Figura 3.17: Modelo de Gilbert.

$$f(l) = \hat{r}(1-\hat{r})^{l-1} \qquad (3.5)$$
$$\bar{f}(l) = (1-\hat{r})\hat{r}^{l-1} \qquad \text{para } l = 1, 2, \ldots, \infty$$

Modelo de Gilbert

Debido a la correlación temporal existente entre los paquetes que son enviados en instantes de tiempo no muy distantes, se está de acuerdo en que el modelo de Bernoulli no aproxima bien las pérdidas para este tipo de envíos [103, 104, 105], que tienden a producirse en ráfagas. Por esta razón se propone el uso de procesos de Markov capaces de capturar la dependencia temporal de las pérdidas.

El modelo de Gilbert es el más simple de ellos, constando únicamente de dos estados (figura 3.17). Dos probabilidades p y q gobiernan las transiciones entre los estados. En el modelo de Gilbert, el estado actual X_t depende sólo del estado anterior, o valor previo X_{t-1}. De forma que,

$$p = P[X_i = 1 | X_{i-1} = 0] \qquad (3.6)$$
$$q = P[X_i = 0 | X_{i-1} = 1]$$

Estas probabilidades pueden estimarse a partir de una traza por medio de las expresiones siguientes,

$$\hat{p} = n_{01}/n_0 \qquad (3.7)$$
$$\hat{q} = n_{10}/n_1$$

en donde n_{01} es el número de veces observado en las series $\{X_t\}_{t=1}^{\infty}$ la secuencia '01' (esto es, un paquete recibido precede a una pérdida), mientras que n_{10} es el número de veces

que aparece la secuencia '10' (una pérdida precede la recepción de un paquete); n_0 es el número de paquetes recibidos (número de 0 en la serie) y n_1 el número de pérdidas (1 en las serie). La distribución de las ráfagas tanto de paquetes recibidos $f(l)$, como de paquetes perdidos $\bar{f}(l)$ viene dada por:

$$\begin{aligned} f(l) &= \hat{p}(1-\hat{p})^{l-1} \\ \bar{f}(l) &= (1-\hat{q})\hat{q}^{l-1} \qquad \text{para } l = 1, 2, \ldots, \infty \end{aligned} \tag{3.8}$$

Comparando las ecuaciones (3.5) y (3.8) se puede observar que este modelo añade un parámetro más que permite independizar la tasa global de pérdidas de la longitud de las ráfagas. Generalmente, el modelo de Bernoulli o bien sobrestima la tasa global, o bien subestima longitud de las ráfagas. Si $p = q$, el modelo de Gilbert se reduce a un modelo de Bernoulli.

Modelo general de Markov

Los procesos de estados finitos de Markov son suficientemente ricos como para representar una gran variedad de dependencias temporales. El modelo de Gilbert es un caso especial de este tipo de modelos. En un proceso de Markov la variable aleatoria actual depende de las n anteriores, siendo este valor n el que define el orden del proceso. De acuerdo con esto, la variable aleatoria X_t se genera dependiendo de en que estado $[x_{t-n}, x_{t-n+1}, x_{t-n+2}, \ldots, x_{t-1}]$ se esté en el proceso,

$$P[X_t = x_t | X_{t-1} = x_{t-1}, \ldots, X_{t-n} = x_{t-n}] = P_n[x_t | x_{t-1}, \ldots, x_{t-n}] \tag{3.9}$$

para cualquier combinación de $[x_{t-n}, \ldots, x_{t-1}]$. Debido a esto, un proceso de Markov de orden n requiere un total de 2^n estados. En el trabajo de Yajnik et al. [103] se muestra que la mayoría de las trazas obtenidas en sus observaciones podían ser modeladas correctamente con $n \leq 6$, sin embargo, en otras se requería un orden superior a 20. En estos casos, el número de estados llega a ser prohibitivo.

Modelo extendido de Gilbert

Sanneck y Carle proponen un modelo diferente, que permite un menor número de estados, llamado modelo extendido de Gilbert [105]. Este modelo sólo necesita $n + 1$ estados para registrar n eventos. A diferencia que los modelos generales de Markov, en donde se

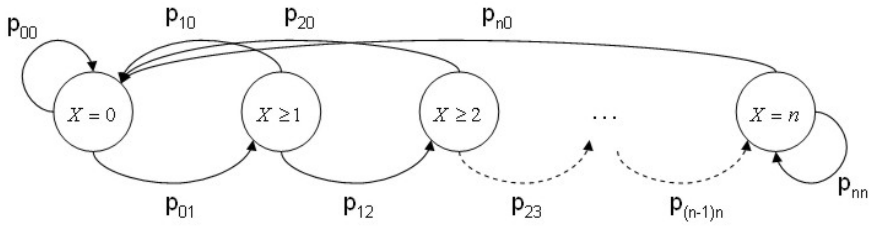


Figura 3.18: Modelo Extendido de Gilbert.

consideran todas las posibles combinaciones de n eventos pasados, la clave de este modelo reside en considerar únicamente aquellos que se corresponden con una ráfaga. En este modelo, el estado inicial indica la recepción de un paquete, mientras que la pérdida consecutiva de un paquete da lugar a una transición al siguiente estado. De esta manera, los estados representen la pérdida de un paquete adicional de forma consecutiva (figura 3.18). Cuando un paquete es recibido tras una ráfaga, se regresa al estado inicial. Así, la matriz de transiciones es de la forma,

$$A^T = \begin{bmatrix} a_{00} & a_{10} & ... & a_{(n-2)0} & a_{(n-1)0} \\ a_{01} & 0 & ... & 0 & 0 \\ 0 & a_{12} & ... & 0 & 0 \\ ... & ... & ... & ... & ... \\ 0 & 0 & 0 & a_{(n-2)(n-1)} & 0 \end{bmatrix} \tag{3.10}$$

en donde $a_{(k-1)k}$ es la probabilidad de que, tras $k-1$ paquetes consecutivos perdidos anteriores, el paquete actual se pierda. Formalmente, el modelo extendido de Gilbert se define a través de una variable aleatoria X, tal que $X = 0$ indica la recepción de un paquete y $X = k$ indica que "exactamente" k paquetes consecutivos se han perdido (en el sentido de que los paquetes precedente y posterior a los k perdidos se han recibido con probabilidad 1) y que "al menos" k paquetes consecutivos se hayan perdido (en el sentido de que el paquete precedente a los k paquetes se ha recibido con probabilidad 1, y el siguiente puede o no recibirse). Entonces, las probabilidades de la matriz de transición A se definen como,

$$a_{(k-1)k} = P(X \geq k | X \geq k-1) \tag{3.11}$$

que puede ser descompuesto de la forma,

$$a_{(k-1)k} = \frac{P(X \geq k \cap X \geq k-1)}{P(X \geq k-1)} = \frac{P(X \geq k)}{P(X \geq k-1)} \tag{3.12}$$

la cual permite que las estimaciones de los parámetros de la matriz A^T se pueden calcular como,

$$a_{01} = \frac{\sum_{i=1}^{n-1} m_i}{m_0} \tag{3.13}$$

$$a_{(k-1)k} = \frac{\sum_{i=k}^{n-1} m_i}{\sum_{i=k-1}^{n-1} m_i} \tag{3.14}$$

$$a_{k0} = 1 - a_{k(k+1)} \tag{3.15}$$

en donde m_i es el número de ráfagas de longitud i, mientras que m_0 es el número de paquetes recibidos. La expresión (3.13) estima la probabilidad de transición desde el estado 0, en el que se recibe el paquete, hasta el estado 1, en el que al menos un paquete se ha perdido. La expresión (3.14) estima la probabilidad de pasar hacia estados con una pérdida incremental de paquetes. Por último, la primera fila de la matriz de transiciones (a_{k0}), referente a la probabilidad de recibir un paquete (volviendo entonces al estado inicial), se estima mediante (3.15). El resto de elementos de la matriz A son nulos.

Cuando $n = 1$ entonces el modelo extendido se convierte en un modelo de Gilbert, mientras que si $n = 0$ obtenemos un modelo de Bernoulli. Por esta razón, estos dos modelos se consideran casos específicos del modelo extendido.

Modelado de las recepciones. Otras métricas

Los modelos anteriores describen el canal IP atendiendo principalmente a la longitud de las ráfagas. Esta métrica, también conocida como *longitud del periodo de pérdida* (*LPL–Loss Period Length*) [106], resulta especialmente útil a la hora de evaluar técnicas de recuperación aplicadas en el receptor. Como veremos en el capítulo 5, el rendimiento de estas técnicas tiene una fuerte dependencia con la longitud de las ráfagas.

Sin embargo, cuando se pretenden evaluar otras técnicas de recuperación, resulta conveniente no sólo modelar cómo se producen las pérdidas, sino también cómo se producen las recepciones. La distancia entre periodos de pérdidas (*ILPL - Inter-Loss-Period-Lengths*) [106] indica la distancia entre ráfagas sucesivas, es decir, las longitudes de las "rafagas" de paquetes recibidos. EL ILPL es una medida relevante cuando se trata de evaluar técnicas basadas en el emisor, ya que estas técnicas basan su funcionamiento en las recepciones. Este métrica suele servir para complementar los modelos anteriores, ya que estos son capaces de modelar bien las longitudes de las ráfagas, pero no las distancias entre ellas. Así, Milner [107] propone una modificación al modelo de Gilbert mediante la

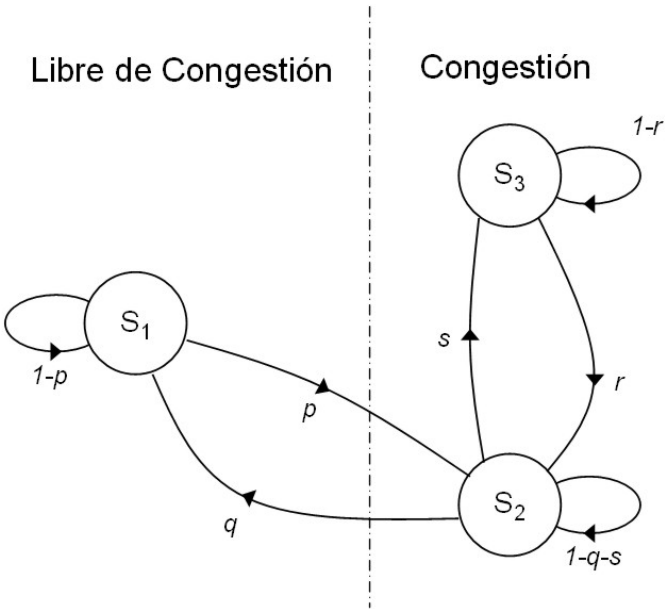


Figura 3.19: Modelo de tres estados para la simulación conjunta de ráfagas pérdidas y recepciones.

cual no sólo se pueda controlar la tasa global de pérdidas y la longitud de las ráfagas, sino también la distancia entre ellas. Para ello, se introduce un nuevo estado S_3 que indica la recepción de un paquete y permite transiciones sólo desde y hacia el estado que indica un paquete perdido (ver figura 3.19). Las probabilidades de estas transiciones vienen dadas por r y s.

La introducción de este estado permite establecer en la red dos condiciones, una condición de congestión y otra libre de ella. Durante la condición libre de congestión se reciben paquetes de forma estable. Esta condición se corresponde con un estado de la red con baja carga. El periodo de esta condición viene dado por la probabilidad de autobucle del estado S_1, $(1-p)$. La condición de congestión se caracteriza por la aparición de pérdidas en ráfagas no excesivamente distanciadas y se corresponde con un estado de saturación en la red. La longitud de las ráfagas viene dada por la probabilidad de autobucle del estado S_2, $(1-q-s)$, mientras que la distancia entre ellas por la probabilidad de autobucle del estado S_3, $(1-r)$. Como puede observarse, este modelo asemeja con bastante realismo la transmisión mediante una red de conmutación de paquetes.

Finalmente, pueden destacarse otras métricas. La distancia entre pérdidas (*ILD - Inter-Loss Distance*) [106] indica la distancia entre paquetes perdidos en términos de números de secuencia. La obtención de ILDs pequeñas en una traza del canal IP implica que las pérdidas de paquetes aparecen a intervalos de tiempo cortos. Esto es indicativo

de la aparición de ráfagas largas así como de la sucesión de ráfagas en periodos cortos de tiempo. Del ILD puede derivarse otra métrica conocida como tasa de pérdidas apreciables (*NLR - Noticeable Loss Rate*) [106]. Para ello, se define un umbral de distancia d_{min}, calculándose el NLR como el número de pérdidas con un ILD inferior a d_{min} dividido por el número total de paquetes. Gracias a esta métrica puede obtenerse una idea acerca de cómo las pérdidas serían apreciables en términos cualitativos.

3.7.2. Impacto de las pérdidas de paquetes sobre el reconocimiento

Como se ha mencionado con anterioridad, las pérdidas de paquetes tienen un efecto degradante sobre la precisión en el reconocimiento, ya que llevan asociadas una pérdida de información. A esto hay que añadirle el hecho de que éstas tienden a producirse en ráfagas, cuyos efectos suelen ser más destructivos que los de las pérdidas aisladas. Es posible identificar entonces dos factores que dominan la influencia de las pérdidas en el reconocimiento:

- El porcentaje de paquetes perdidos, ya que cuantos más paquetes se pierdan menos información estará disponible para el reconocimiento, dificultándolo.
- La longitud media de las ráfagas, puesto que cuanto mayor sea la longitud de las ráfagas, segmentos más largos de voz se habrán perdido, impidiendo un buen desempeño de los algoritmos de mitigación.

Para evaluar la influencia de estos dos factores, se han llevado a cabo una serie de pruebas de reconocimiento con distintos porcentajes de pérdidas y longitudes medias de ráfaga. Para ello, se ha recurrido a un modelo de Gilbert, ya que supone un buen compromiso entre simplicidad y capacidad para simular ráfagas de pérdidas. Gracias a los parámetros p y q es posible establecer de forma independiente un porcentaje global de tramas perdidas, R_{loss}, y una longitud media de ráfaga, L_{loss}. A partir de las expresiones (3.7) y (3.8) se puede deducir que las probabilidades de transición para una determinada R_{loss} y L_{loss} vienen dadas por:

$$\begin{aligned} p &= \frac{R_{loss}}{L_{loss} \cdot (1 - R_{loss})} \\ q &= \frac{1}{L_{loss}} \end{aligned} \tag{3.16}$$

El marco experimental de las pruebas incluye al sistema de referencia propuesto en la sección 2.5. Este sistema se ha organizado de acuerdo con la arquitectura DSR, en donde las tareas para la representación de la señal de voz se realizan en el cliente, antes de la transmisión por la red IP. Para la extracción de características se aplica el estándar de DSR para IP. Una vez obtenidos los vectores características, estos se agrupan en FPs y se empaquetan conforme al estándar. En las pruebas realizadas se considera que cada paquete sólo contiene un FP. Aunque esto supone un desperdicio del ancho de banda, debido a la sobrecarga originada por las cabeceras, está recomendado en la especificación del payload DSR (RFC 3557 [79]), ya que se minimizan los efectos de la pérdida de paquetes. Además, nos permite extrapolar fácilmente los resultados que se obtendrían con transmisiones en las que los paquetes contuvieran más de un FP.

Gracias a la numeración incluida en la cabecera RTP es posible identificar los paquetes perdidos. Los vectores que contuvieran estos paquetes son recuperados mediante el algoritmo de mitigación propuesto por el estándar, el cual consiste en una repetición de los vectores de características más cercanos que hayan sido recibidos.

La figura 3.20 muestra los resultados del reconocimiento obtenidos en este sistema empleando tasas de paquetes perdidos desde el 10 % al 50 % en incrementos del 10 %, así como con longitudes medias de ráfaga de 1, 2, 4, 8, 12, 16 y 20 paquetes. Puede observarse que para una mismo porcentaje de pérdidas, las longitudes medias de ráfaga largas reducen considerablemente más la precisión en el reconocimiento que aquellas más cortas. Resulta relevante el caso en que $L_{loss} = 1$ y $R_{loss} = 0{,}5$. En esta condición se supone que todas las pérdidas son aisladas (rafagas de longitud 1). Al imponer como condición que se pierdan el 50 % de los paquetes, aparece un patron periódico en el que se pierde un paquete cada vez que se recibe uno y viceversa. En esta condición, aunque no es ni mucho menos realista, indica que es posible perder incluso la mitad de la información sin afectar significativamente la precisión en el reconocimiento. Otros autores [108, 109] han obtenido resultados experimentales semejantes, llegando a la conclusión de que el estandar DSR sufre una disminución de la precisión practicamente despreciable incluso a muy altas tasas de pérdidas (de hasta el 80 %) siempre que la longitud de las ráfagas sea muy corta (inferior a cinco vectores).

3.8. Canales con pérdidas. Una visión unificada

Las soluciones examinadas anteriormente, NSR sobre GSM y DSR sobre IP, pueden verse de forma unificada bajo el concepto de *canales con pérdidas*. Estos canales se caracterizan

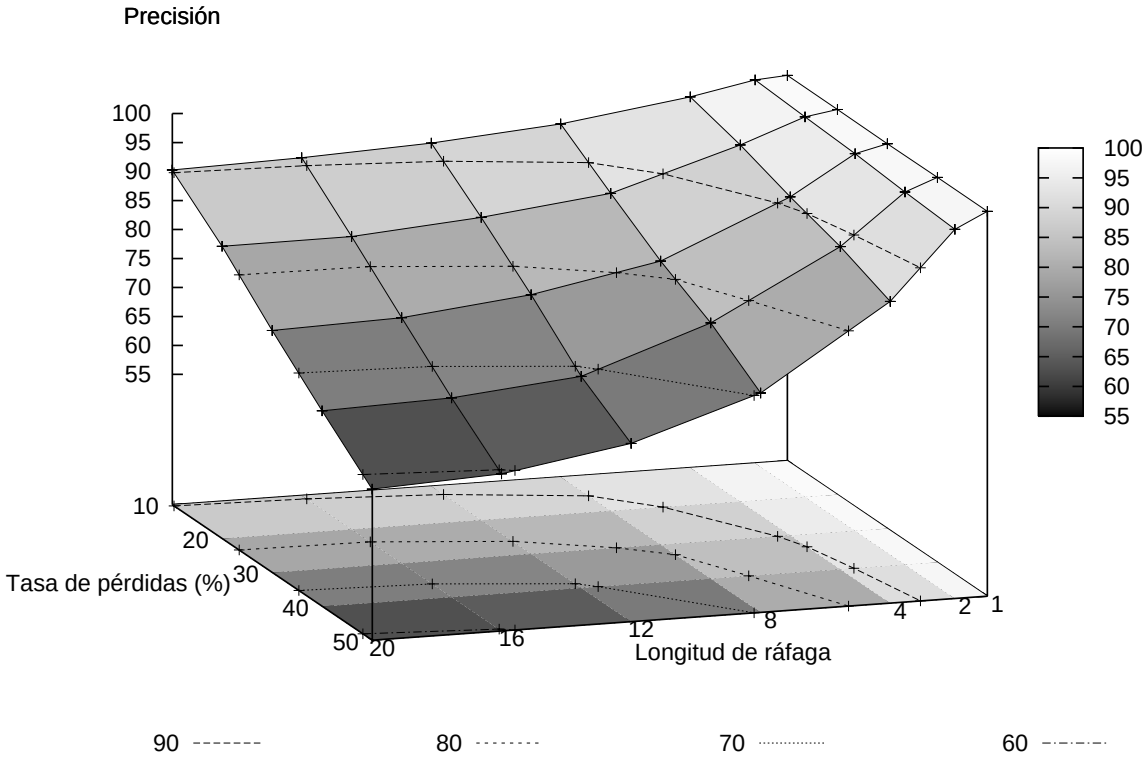


Figura 3.20: Precisión del reconocimiento según el porcentaje de paquetes perdidos y la longitud media de ráfaga, empleando la mitigación del estándar DSR para IP.

por la pérdida de fragmentos completos de información. A diferencia de otros canales, implícitamente se asume que el acceso a información transmitida se realiza en capas superiores, en donde las unidades de transmisión son bloques de datos o paquetes y no bits. Bajo el concepto de canal con pérdidas las aproximaciones anteriores presentan una problemática similar:

- Las redes IP no ofrecen un servicio confiable de entrega de paquetes, debido a su diseño de máxima simplicidad, por lo que, como se argumentó en la sección 3.3.4, los paquetes pueden perderse durante la transmisión. Es responsabilidad de los extremos cooperar para la recuperación de estas pérdidas, usualmente mediante la retransmisión de los paquetes. Sin embargo, en comunicaciones con requerimientos de tiempo real, como el reconocimiento de la voz, la retransmisión provoca más inconvenientes que ventajas (carga aún más la red e incrementa la latencia).

- En el caso de GSM la transmisión se realiza mediante un enlace de radio que está altamente expuesto al ruido. Debido a ello, pueden producirse alteraciones graves en las tramas de voz transmitidas, de tal forma que su síntesis sea especialmente desagradable para el receptor. Estas tramas se descartan, y por tanto la información que contuvieran, siendo sustituidas por una extrapolación y apagado basado en tramas anteriores.

En ambos casos tenemos un canal en donde segmentos completos de información, o bien se pierden, o bien se descartan. Sin embargo, debido a cómo se codifica la información que se transmite, el efecto de una pérdida es diferente. En DSR sobre IP, lo que se transmite son vectores de características. La codificación de estos vectores no es de naturaleza predictiva, no siendo necesario la decodificación del vector anterior para la decodificación del vector actual. Esto es, cuando se pierde un paquete sólo se ven afectados los vectores de características que contuviera ese paquete. El resto permanecen intactos y la degradación queda confinada a la zona donde se ha producido la pérdida.

Esto no ocurre en NSR sobre GSM, la codificación predictiva de la voz obliga a que para la decodificación de una trama, sea necesaria la anterior. La degradación no queda confinada a la zona donde se ha producido la pérdida, sino que ésta se extiende durante un tiempo posterior a ella. Sin embargo, es necesario notar que la degradación posterior a la pérdida es de diferente naturaleza que aquella debida a la pérdida de información. Es decir, la información se ha recibido, y es correcta, sin embargo, no es posible decodificarla sin introducir una distorsión.

Así, mientras que en la solución DSR sobre IP la única degradación se debe a la pérdida de paquetes, en la aproximación NSR sobre GSM podemos identificar principalmente dos distorsiones:

- Ruido de ráfaga, debido al mecanismo de mitigación aplicado sobre aquellas tramas con graves alteraciones. Ya que las tramas son descartadas, se produce una pérdida de información similar a la que se produce en IP cuando se pierden paquetes.

- Ruido de memoria, derivado de las memorias del codec. La degradación provocada por el ruido de ráfaga se extiende más allá de las tramas descartadas durante la síntesis de la voz. Esta degradación provoca una reducción notable de la precisión del reconocimiento del habla.

En este sentido, el ruido de ráfaga en GSM y la pérdida de paquetes en IP pueden tratarse de la misma forma. En ambos casos tenemos una pérdida de información cuyos

efectos tendremos que mitigar mediante alguna técnica de reconstrucción. En cambio, para el ruido de memoria, se deberán aplicar soluciones especificas. En este caso, las tramas son recibidas pero se introduce una distorsión durante su síntesis. Como es de esperar, estas soluciones pasaran por evitar, en la medida de lo posible, sintetizar la voz en el decodificador. A ello se dedica el siguiente capítulo.

Capítulo 4

TRANSPARAMETRIZACIÓN DE VOZ CODIFICADA

4.1. Introducción

El carácter predictivo de la tecnología empleada por los codificadores GSM permite obtener, con una tasa de bits significativamente inferior, una calidad de voz comparable a la de la telefonía fija, habilitando la transmisión de la voz por enlaces de radio, cuyo ancho de banda está limitado. Sin embargo, esta misma tecnología los hace vulnerables frente a la aparición de tramas erróneas. Estas tramas no sólo dan lugar a una pérdida de información, sino que también provocan la aparición de un ruido de memoria posterior. Como se mostró en el capítulo anterior, el ruido de memoria tiene un efecto negativo no despreciable sobre la precisión del reconocimiento.

El efecto del ruido de memoria sobre el reconocimiento puede aliviarse, como se mostrará más adelante en este capítulo, mediante técnicas especializadas aplicadas sobre la señal de voz decodificada. Sin embargo, dado que este ruido se origina durante la síntesis de la señal de voz, la mejor forma de combatirlo consiste en, precisamente, evitar la decodificación de la voz. Una posible solución consiste en realizar el reconocimiento a partir de los parámetros que codifican la voz. Aunque presentados en el contexto del ruido de codificación, los trabajos realizados por Huerta et al. [85, 110] siguen esta línea. Otra solución, en la que se enmarca este capítulo, consiste en la *transparametrización* de la voz. Mediante la transparametrización, los parámetros de voz son convertidos en vectores de características válidos para el reconocimiento. De esta forma, se evita, en la medida de lo posible, la síntesis de la señal y el ruido de memoria asociado a ella.

El concepto de transparametrización no es nuevo, desde que fuera presentado de forma preliminar por Gallardo et al. [111, 112] diferentes variantes han sido propuestas. Así, Kim et al. en [113] describen un transparametrizador para el codec IS-641 robusto en canales con pérdidas y ruido acústico, mientras que Pelaez et al. proponen en [91, 114] diferentes transparametrizadores para los estándares FR, HR y G.723.1. Sin embargo, ningún transparametrizador ha sido propuesto para EFR y AMR, los codecs de segunda generación de telefonía móvil mayoritariamente usados en la actualidad.

A lo largo de este capítulo, nos centraremos en el desarrollo de un transparametrizador para cada uno de estos codecs. Gracias a estos, podremos tratar de forma efectiva el ruido de memoria, consiguiendo, en el caso de EFR, unas prestaciones similares (incluso superiores en algunos casos) a las obtenidas con una arquitectura DSR. En el caso de AMR, podremos comprobar, además, que la transparametrización tiene una influencia positiva frente a la distorsión introducida por el codec. A diferencia de otros transparametrizadores, los aquí desarrollados se caracterizarán por la obtención de vectores de características compatibles con el estándar DSR. Es decir, tendrán como ventaja adicional que no es necesario practicar modificación alguna sobre los back-ends basados en DSR. Por esta razón, en este mismo capítulo se propondrán distintas alternativas para integrar la transparametrización en las redes GSM. Estas alternativas permitirían una transición suave hacia la arquitectura DSR.

4.2. Codificadores de Análisis por Síntesis

Los codecs empleados en GSM (incluyendo los distintos modos de AMR) son *codificadores de análisis por síntesis*. Estos codificadores, también denominados codificadores híbridos, mezclan conceptos propios de los codificadores de forma de onda [115, 116, 117, 118] y de los codificadores paramétricos o vocoders [24, 119, 120, 121, 122, 123], aunando las ventajas de ambos métodos. Por un lado, obtienen una tasa de bits muy baja (de 0.5 a 2 bits/muestra) cercana a los codificadores paramétricos, mientras que por otro lado, el sonido sintetizado alcanza una calidad y naturalidad propia de los codificadores de forma de onda.

Es bien conocido que el sistema auditivo humano realiza una transformación en frecuencia de tiempo corto sobre las señales acústicas anterior a la transducción neuronal y posterior percepción [120]. A diferencia de los codificadores de forma de onda, en los codificadores paramétricos no se intenta reproducir de forma precisa la forma de onda en el dominio del tiempo sino tan sólo una señal auditivamente equivalente a ella. Para

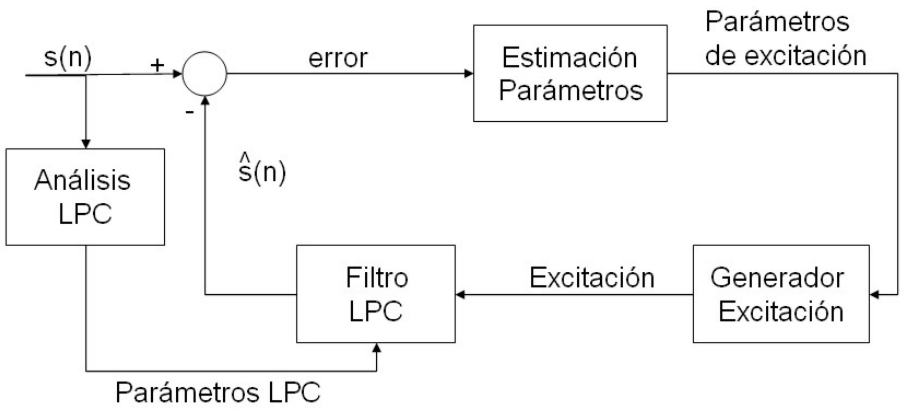


Figura 4.1: Codificador de Análisis por Síntesis generalizado.

ello, se recurre a un modelo matemático de producción de voz que caracteriza la señal de voz, codificándose sólo parámetros de dicho modelo. Este modelo, usualmente denominado modelo de producción de voz, es empleado tanto en el proceso de codificación como en el de decodificación.

En los codificadores de análisis por síntesis se emplea un modelo de producción de señal de voz constituido por un filtro lineal variable en el tiempo. La diferencia con respecto a los codificadores paramétricos radica en cómo se representa la excitación de dicho filtro, determinándose de la misma forma que en los codificadores de forma de onda. Sin embargo, en vez de intentar encontrar una representación que minimice la diferencia con respecto a la señal de excitación original, se busca una representación de la excitación que, para el filtro de síntesis dado, produzca una señal que sea lo más similar posible a la señal original de entrada. Esto se justifica porque cada muestra de la señal de excitación afecta a muchas muestras de la señal de voz reconstruida, debido a la estructura recursiva del filtro de predicción. Por consiguiente, el error ocasionado por la cuantización de la señal de excitación se debe obtener midiendo su efecto sobre la señal de voz reconstruida durante más de una muestra.

En la figura 4.1 se muestra la estructura básica de un codificador de análisis por síntesis. La señal de voz $s(n)$ es comparada con la señal sintetizada $\hat{s}(n)$. A partir del error $e(n)$ entre las dos señales, o señal residual, se realiza una optimización en bucle cerrado de los parámetros que generan la excitación de la señal sintetizada $\hat{s}(n)$. La forma en que se representa la señal excitación es lo que, en última instancia, diferencia a los diferentes codificadores híbridos.

4.2.1. Filtros en el Análisis por Síntesis

En los codificadores híbridos se aplican comúnmente tres filtros diferentes: el filtro de predicción de retardo corto (LPC) que determina la envolvente de la señal, un filtro de retardo largo (LTP) que aproxima la estructura fina de la excitación, y un filtro de peso perceptual que explota la forma en que la señal de voz es percibida por el oído, desplazando el ruido de cuantización a aquellas frecuencias en las que es más difícil percibirlo.

Filtro LPC

Partiendo de que la señal de excitación $u(n)$ nos es totalmente desconocida, el cálculo de los coeficientes LPC del filtro de retardo corto se realiza a partir de un predictor lineal de orden p para la señal $s(n)$. Este predictor calcula una señal predicha $\tilde{s}(n)$ como combinación lineal de las p muestras anteriores,

$$\tilde{s}(n) = -\sum_{k=1}^{p} \alpha_k s(n-k) \tag{4.1}$$

donde α_k, con $k = 1, \ldots, p$ son los coeficientes de predicción lineal, o coeficientes LPC. Estos coeficientes se calculan de forma que la energía del error de predicción o energía residual, E, sea mínima,

$$\frac{\partial E}{\partial \alpha_k} = 0 \quad (k = 1, \ldots, p) \quad \text{con} \quad E = \sum_n e^2(n) \tag{4.2}$$

donde la suma está extendida a la ventana de análisis y $e(n)$ representa el error de predicción o *señal excitación*, definido como,

$$e(n) = s(n) - \tilde{s}(n) = s(n) + \sum\nolimits_{k=1}^{p} \alpha_k s(n-k) \tag{4.3}$$

Los coeficientes LPC del predictor α_k se identifican con los coeficientes del filtro LPC a_k. Esto se justifica por las siguientes razones:

- Si se asume $a_k = \alpha_k$, el error de predicción es $e(n) = Gu(n)$. Para sonidos sonoros, la excitación $u(n)$ es un tren periódico de pulsos, de modo que la excitación de la señal toma valores pequeños la mayor parte del tiempo y la energía residual es pequeña. Ésto es consistente con el hecho de que los coeficientes del predictor son calculados como aquellos para los que se minimiza la energía residual.

- Si la señal es generada por el filtro todo-polos $H(z)$ excitado por un pulso simple o por un ruido blanco estacionario, el modelo autoregresivo garantiza que los coeficientes LPC resultantes (los que minimizan la energía residual) coinciden con los coeficientes del filtro LPC.

La identificación realizada tiene la ventaja de que el método propuesto conduce a un sistema de p ecuaciones lineales, obtenidas de 4.2, que pueden ser resueltas fácilmente. Para ello, existen dos métodos bien estudiados: el de autocorrelación y el de covarianza (descritos en [124]). El primero de ellos tiene la ventaja de garantizar la estabilidad del filtro resultante y de poseer un algoritmo de resolución más rápido que el segundo.

Filtro LTP

Las correlaciones de retardo largo existentes en los segmentos sonoros, debidas a los pulsos glotales casi periódicos (periodo de pitch), pueden ser explotadas para reducir la tasa de bits necesaria para la transmisión de la señal excitación. Por esta razón, los codificadores híbridos suelen incorporar un filtro de largo retardo (*LTP–Long Term Prediction*) denominado tradicionalmente *filtro de pitch*, que tiene la forma general,

$$P(z) = 1 - \sum_{k=-q}^{q} b_k z^{-(M+k)} \tag{4.4}$$

donde M es el retardo en muestras (en el rango de 2 a 20 ms) y b_k son los coeficientes de predicción de retardo largo. Para señales periódicas, el retardo M correspondería a un periodo de pitch, mientras que para señales no periódicas el retardo es aleatorio. Los parámetros del LTP se pueden determinar a partir de la señal excitación, o bien directamente de la señal de voz. El procedimiento para determinar el valor de los coeficientes b_k es similar al del caso de los coeficientes LPC si el valor de M está ya fijado, pudiendo determinarse mediante el método de autocorrelación o covarianza [125]. Sin embargo, la optimización conjunta de los coeficientes b_k junto con M implica una cantidad excesiva de cálculo, ya que habría que determinar los coeficientes b_k para cada valor de M. Así pues, para poder abordar este problema numéricamente, lo que se hace es determinar M como si tuviéramos un filtro con un único coeficiente, calculando el resto de coeficientes una vez que se ha calculado el retardo.

Típicamente se utilizan de uno a tres coeficientes de predicción, y sus valores se adaptan a un ritmo mayor que los coeficientes LPC, de 50 a 200 veces por segundo. La predicción

mejora conforme se aumenta el orden del predictor, pero a cambio se necesitan más bits para codificar los coeficientes adicionales. Por ello, en vez de usar un predictor de orden alto, se puede usar un predictor de menor orden con un retardo no entero [126], el cual permite una cuantización más eficiente de los parámetros.

Filtro de Peso

Por último, para explotar la forma en que la señal de voz es percibida por el oído, se incluye el denominado filtro de peso. Este filtro explota el efecto de enmascaramiento, esto es, la poca capacidad del sistema auditivo para detectar ruido en las bandas de frecuencia en las cuales la señal de voz tiene una alta energía. El filtro de peso se aplica sobre la señal error entre la señal sintetizada y la original, y tiene una forma similar al filtro LPC pero invertida,

$$W(z) = \frac{1 - \sum_{k=1}^{p} \gamma_1^k a_k z^{-k}}{1 - \sum_{k=1}^{p} \gamma_2^k a_k z^{-k}} \tag{4.5}$$

donde los parámetros γ_1, γ_2 se determinan mediante pruebas auditivas, tomando valores entre 0 y 1.

Cuando la señal de error es filtrada por el filtro de peso, las regiones correspondientes con los formantes se suavizan, mientras que se realzan los valles del espectro. De esta forma, al calcular el error cuadrático medio, por el que se guía el proceso de minización, se ponderan más los valles del espectro, donde el oído es capaz de distinguir mejor el ruido, frente a las zonas de los formantes, en donde se produce un mayor efecto de enmascaramiento al contener más energía. Puede observarse que dicho procedimiento de peso no afecta ni a la razón de transmisión ni a la complejidad del procedimiento de síntesis, tan sólo aumenta la complejidad del codificador.

4.2.2. Codificación Excitada por Código

De entre todas las técnicas de análisis por síntesis la más popular es la denominada predicción lineal excitada por código (*CELP–Code Excited Linear Prediction*) [127]. En ella tanto el codificador como el decodificador almacenan un diccionario de secuencias posibles (códigos) para la excitación, de forma que sólo sea necesario transmitir un índice para describir la excitación de cada segmento. De esta forma, los codificadores CELP obtienen las mejores prestaciones con menor tasa de bits.

El método más comúnmente empleado en la codificación excitada por código consiste en la división en dos pasos del bucle cerrado de análisis por síntesis. El primer paso

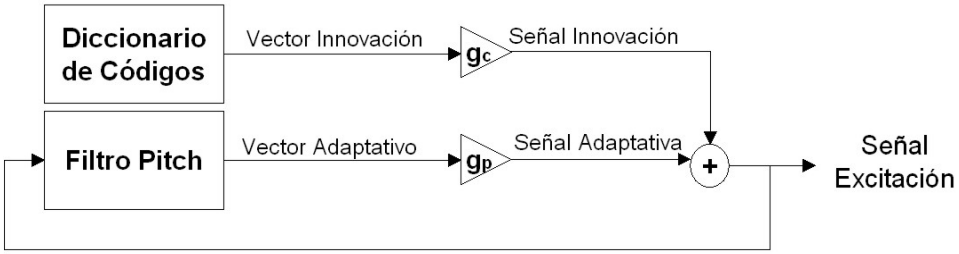


Figura 4.2: Diagrama simplificado de la codificación CELP de la excitación.

consiste en un análisis del pitch con respecto al segmento previo de excitación, que se realiza mediante una búsqueda lineal guiada por el filtro de pitch (LTP). Cuando esta excitación, basada en la excitación previa, se construye, se ajusta mediante un cálculo de la ganancia (*ganancia adaptativa*), de forma que se maximice la correspondencia entre la señal original y la sintetizada. Esta forma de generación de la excitación se conoce como la obtención del diccionario adaptativo o *señal adaptativa*, que no es más que los parámetros que caracterizan al filtro de pitch y la ganancia adaptativa.

El segundo paso del proceso de codificación de la excitación consiste en una búsqueda sobre un diccionario estático de vectores estocásticos orientado al modelado de sonidos sordos. El error residual que se obtiene en la primera parte, también denominado *señal innovación*, se compara con las secuencias posibles del diccionario estático o de códigos. El índice de aquel código que dé lugar al menor error entre la señal sintetizada y la forma de onda original es transmitido al receptor, junto con la correspondiente ganancia (*ganancia estática*). Los diccionarios de códigos varían según la implementación concreta de CELP. En las formulaciones originales de la codificación CELP, el diccionario estaba formado por secuencias aleatorias fijas con una distribución gaussiana de varianza la unidad, puesto que se intentaba generar excitaciones con forma ruidosa para los segmentos sordos.

Mediante esta elegante técnica se evita tomar la decisión de si el sonido es sonoro o sordo. En vez de esto se considera que la excitación es una mezcla de ambos, estableciéndose los pesos de esta mezcla por medio de las ganancias adaptativa y estática (figura 4.2). La principal dificultad de la implementación práctica de un codificador CELP estriba en la enorme cantidad de cálculo que requiere, ya que es necesario que cada una de las secuencias del diccionario sea filtrada tres veces. Por esta razón, han surgido diversas propuestas para mejorar la eficiencia computacional de los codificadores CELP [128].

Una de las más empleadas consiste en utilizar diccionarios dispersos, con un alto porcentaje de ceros (entre un 90-95 %), también denominados *ralos*. Los diccionarios binarios

(-1, -1) y ternarios (-1, 0, 1) ofrecen rendimientos similares a los obtenidos por diccionarios completamente estocásticos, con la diferencia de que su convolución con los filtros CELP implica una menor complejidad, al reducirse las multiplicaciones a adiciones y sustracciones. Estos diccionarios se pueden expresar de forma algebraica dando lugar a lo que se conoce como codificación *ACELP* (*CELP Algebráico*). Los codificadores de mayor calidad en GSM, AMR y EFR, emplean esta técnica de codificación.

Otra opción consiste en el uso de diccionarios estáticos solapados, en los que en vez de almacenarse vectores independientes, se emplee un único vector de gran longitud sobre el que se aplica un desplazamiento fijo. Además de requerir menos bits para su representación, esta técnica permite el uso de algoritmos eficientes que aprovechen los cálculos realizados sobre vectores adyacentes.

4.2.3. Otras técnicas para codificación de la excitación

Existen otras técnicas más sencillas para la codificación de la excitación, aunque éstas obtienen peores resultados que las técnicas CELP. Entre ellas destacan la codificación multipulso y la codificación por pulsos regulares, esta última es la empleada por el estándar FR de GSM.

En los codificadores multipulso (*MPE–MultiPulse Excited*), la excitación se aproxima como una secuencia de pulsos espaciada de forma no uniforme [129]. Debido a la dificultad que supone localizar las posiciones y amplitudes óptimas de los pulsos de forma conjunta, los codificadores MPE siguen un procedimiento subóptimo. Iterativamente se buscan, mediante un procedimiento de análisis por síntesis, la posición y amplitud óptimas pulso a pulso. Algunas implementaciones más precisas vuelven a estimar en cada paso las amplitudes de los pulsos ya establecidos, pero manteniendo sus posiciones. El número de pulsos necesario para la codificación suele variar de 4 a 6 pulsos por segmento de 5 ms. Para cada pulso es necesario codificar tanto las posiciones como las amplitudes, lo cual requiere del orden de 7 a 8 bits por pulso.

Una variante de la codificación multipulso es la codificación con pulsos regulares (*RPE–Regular Pulse Excited*) [126]. Como su propio nombre indica, en esta codificación los pulsos están espaciados uniformemente, utilizándose de 10 a 12 pulsos por segmento de 5ms. De esta forma sólo es necesario codificar la posición y amplitud del primer pulso y las amplitudes del resto, reduciendo la cantidad de información que debe ser transmitida. La posición del primer pulso y las amplitudes de los pulsos pueden determinarse, de forma óptima, mediante un bucle cerrado de análisis por síntesis [125].

4.3. Mejora del reconocimiento sobre voz decodificada

Como punto de partida, en esta sección mostraremos como mejorar el rendimiento de un sistema sólo-servidor o NSR en donde el reconocimiento se realiza sobre voz decodificada. Ésta es la forma más inmediata de reconocer la voz, es decir, obteniendo los correspondientes vectores de características a partir de la voz transmitida y sintetizada. Como se analizó en el capítulo anterior, las degradaciones del canal de comunicaciones dan lugar a varios tipos de distorsiones o ruidos sobre la señal sintetizada. Inicialmente podríamos intentar aliviar estos efectos sobre la propia señal de voz, sin embargo, ya que nuestro objetivo es la mejora de la precisión en el reconocimiento, resulta más eficiente compensar estos ruidos sólo sobre aquellas características de la voz relevantes para el reconocimiento, esto es, sobre los propios vectores de características.

Ya que la naturaleza de cada ruido es distinta y, por tanto, debe tratarse de forma especializada, el primer paso que deberemos afrontar consiste en la clasificación de cada vector de acuerdo con el ruido que le afecta. Sin embargo, a la hora de realizar esta identificación, deben tenerse en cuenta las posibles diferencias de tamaño, así como de desplazamiento, entre las tramas del codec de voz y las ventanas de análisis empleadas durante la extracción de características. Debido a que persiguen objetivos diferentes, transmisión o reconocimiento de voz, éstas suelen diferir, empleándose tramas de mayor tamaño pero solapadas para el reconocimiento del habla, a fin de disponer de una alta resolución espectral.

Así ocurre, de hecho, en nuestro sistema de referencia. Las tramas de los codecs FR, EFR y AMR representan segmentos de voz de 20 ms no solapados, mientras que el extractor de características (al igual que el front-end definido en el estándar DSR) opera en ventanas de análisis de 25 ms obtenidas cada 10 ms, es decir, solapadas en 15 ms. Por sencillez, clasificaremos los vectores como correctos o incorrectos, en estrecha relación con la clasificación de las tramas recibidas que realiza el decoder. Para ello, supondremos que disponemos de la marca de trama incorrecta o BFI de cada segmento, recurriendo a una función de mapeo que marca cada vector. Esta función viene dada por,

$$F_{map}(n) = \begin{cases} 1 & \left(BFI\left(\left\lfloor \frac{n}{2} \right\rfloor\right) = 1\right) \; or \; \left(BFI\left(\left\lfloor \frac{n+1}{2} \right\rfloor\right) = 1\right) \\ 0 & \text{; en otro caso} \end{cases} \tag{4.6}$$

donde n es el índice temporal del vector de características usado en el reconocedor, mientras que $BFI(m)$ es el indicador de trama incorrecta correspondiente la m-ésima trama

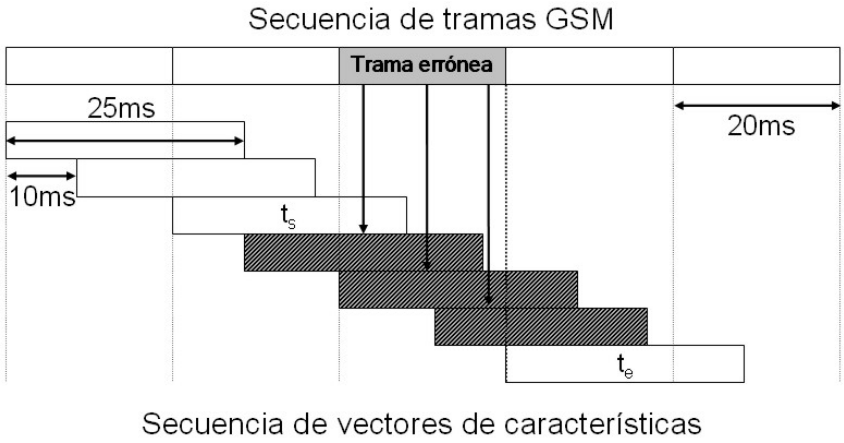


Figura 4.3: Esquema de asociación entre tramas GSM y vectores de características erróneos.

del codec.

Un ejemplo de como actúa este mapeo puede verse en la figura 4.3. En dicha figura puede observarse que la función de mapeo identifica como incorrectos aquellos vectores cuya ventana de análisis contenga muestras de voz correspondientes a tramas GSM incorrectas, con excepción del primer vector anterior a la ráfaga (t_s). Este vector contiene 5ms de señal de voz sintetizada a partir de una trama errónea. Sin embargo, si tenemos en cuenta el algoritmo de sustitución y apagado propuesto por el estándar, en donde la primera trama de la ráfaga se sustituye por la inmediatamente anterior, cabe suponer que las características espectrales en la ventana de análisis se mantienen, pudiéndose considerar este vector como correcto.

4.3.1. Reconstrucción de ráfagas

Como se mencionó en el capítulo anterior, cuando una trama errónea es recibida en el receptor, ésta es descartada, siendo la señal de voz sintetizada el resultado de un algoritmo de mitigación. Debido a ello, los vectores obtenidos a partir de estas tramas, marcados como incorrectos por la función de mapeo, pueden considerarse igualmente perdidos y ser sustituidos por un algoritmo de mitigación orientado al reconocimiento.

La mitigación de errores en GSM está fuertemente limitada por la latencia, consistiendo en un reemplazo parcial con valores extrapolados únicamente a partir de de las tramas correctas anteriores, a la vez que se aplica una reducción de las ganancias. Esto se debe a los estrictos requerimientos temporales de la transmisión de voz. Sin embargo, en el reconocimiento de voz, aunque sigue siendo importante, la latencia no resulta tan crítica. De

hecho, el mismo estándar de DSR no realiza una extrapolación del último vector recibido, sino que espera la recepción del primer vector tras la ráfaga para aplicar una interpolación entre ambos. Esto supone una clara ventaja con respecto a la mitigación empleada para la transmisión de voz, ya que, a cambio de incrementar la latencia, se dispone de más información para realizar la mitigación de los vectores perdidos.

Así pues, de forma análoga al estándar de DSR, los vectores marcados como incorrectos deberían ser sustituidos mediante un algoritmo de mitigación que tenga en cuenta no sólo el último vector recibido antes de la ráfaga, sino también el primero posterior a ella. Aunque en capítulos posteriores se desarrollarán algoritmos de mitigación más complejos para vectores perdidos, inicialmente, propondremos una sencilla interpolación lineal entre el último y el primer vector correcto recibido antes y después de la ráfaga definida como,

$$\hat{\mathbf{x}}(t) = \mathbf{x}(t_s) + \frac{\mathbf{x}(t_e) - \mathbf{x}(t_s)}{t_e - t_s}(t - t_s) \qquad (t_s < t < t_e) \tag{4.7}$$

donde $\hat{\mathbf{x}}(t)$ es el vector de características estimado para el instante de tiempo t, y $\mathbf{x}(t_s)$ y $\mathbf{x}(t_e)$ son, respectivamente, el último y el primer vector recibido antes y después de la ráfaga.

4.3.2. Compensación del ruido de memoria

Incluso si toda la distorsión debida al ruido de sustitución y apagado pudiera eliminarse, aún quedaría una degradación de muy diferente naturaleza debida a la memoria del codec. En este caso ha de recordarse que no estamos tratando una pérdida de información, sino una degradación en la señal que denominamos ruido de memoria. Para tratar esta degradación partiremos de la hipótesis de que este ruido se comporta de forma similar al ruido acústico, es decir, su efecto sobre el dominio del cepstrum y el espectro logarítmico de la señal puede modelarse mediante factores aditivos.

Bajo esta hipótesis, se propone la adaptación de la técnica FCDCN (*Fixed Codeword-Dependent Cepstral Normalization*) diseñada originalmente para la compensación de ruido acústico [32]. La idea básica del FCDCN consiste en la aplicación de un vector de corrección $\mathbf{r}$ sobre el vector ruidoso $\mathbf{y}$ que depende de la SNR instantánea y del propio vector $\mathbf{y}$. La dependencia de $\mathbf{r}$ sobre $\mathbf{y}$ se simplifica mediante una cuantización vectorial de $\mathbf{y}$. Así, la estimación resultante $\hat{\mathbf{x}}$ del vector limpio $\mathbf{x}$ se obtiene como,

$$\hat{\mathbf{x}} = \mathbf{y} + \mathbf{r}(SNR, q) \tag{4.8}$$

donde q es el índice de cuantización correspondiente a $\mathbf{y}$. Para la obtención de los vectores de corrección se recurre a una grabación simultanea de datos de voz limpia y ruidosa, también conocida como grabación de *datos en estéreo*. En el ámbito de la compensación del ruido acústico, esto supone una limitación, ya que la técnica sólo puede aplicarse en aquellas situaciones donde se pueda disponer de datos estéreo. Sin embargo, en nuestro caso, podemos emplear la base de datos de entrenamiento y una simulación del canal para generar cuantos datos estéreo necesitemos.

El principal inconveniente en la aplicación de la técnica FCDCN al ruido de memoria consiste en modelar la SNR instantánea. Como se mostró en la figura 3.14, el nivel de degradación no es el mismo para todos los vectores tras una ráfaga, dependiendo de:

- La longitud de la ráfaga previa, l, ya que conforme ésta aumenta, es de esperar una mayor intensidad del ruido de memoria posterior.
- La distancia al final de la ráfaga, t, puesto que el ruido de memoria disminuye, al ser absorbido por el decoder, conforme avanza el tiempo.

Dado que la SNR instantanea viene dada por los dos factores anteriores, la estimación resultante $\hat{\mathbf{x}}$ del vector limpio $\mathbf{x}$ se puede obtener a través de vectores de corrección dependientes de l, t y q, es decir,

$$\hat{\mathbf{x}} = \mathbf{y} + \mathbf{r}(l, t, q) \tag{4.9}$$

El conjunto de vectores de corrección $\mathbf{r} = \mathbf{r}(l, t, q)$ se obtiene empleando datos en estéreo recreados a partir de la base de datos de entrenamiento. Para cada longitud de ráfaga l se lleva a cabo una simulación sobre la base de datos de entrenamiento de cuantas ráfagas de error sean necesarias para garantizar una buena estimación del vector de corrección. Supuesto que se han obtenido M ráfagas de longitud l, el vector de corrección se obtiene mediante un promediado dado por,

$$\mathbf{r}(l, t, q) = \frac{1}{M} \sum_{m=1}^{M} \left(\mathbf{x}_t^{(m)} - \mathbf{y}_t^{(m)} \right) \qquad q = VQ\left[\mathbf{y}_t^{(m)} \right] \tag{4.10}$$

donde $\mathbf{x}_t^{(m)}$ y $\mathbf{y}_t^{(m)}$ corresponden, respectivamente, al vector de características limpio y ruidoso (con ruido de memoria) en el instante de tiempo t después del final de la ráfaga m-ésima. En adelante, nos referiremos a esta adaptación de la técnica FCDCN para la compensación de ruido de memoria como *A-FCDCN* o FCDCN adaptado.

Como puede observarse, el cálculo de los vectores de corrección requiere una importante cantidad de cómputo, ya que es necesario simular M ráfagas, siendo M un número suficientemente alto para obtener una estadística significativa, con l longitudes distintas. Pese al bajo coste computacional de la ecuación (4.10), ésta debe aplicarse $l \cdot t$ veces, implicando $m \cdot l \cdot t$ cuantizaciones. Sin embargo, el cálculo de los vectores de corrección sólo debe realizarse una vez. Una vez conocidos y almacenados, tan sólo es necesaria una cuantización del vector ruidoso para aplicarlos. De esta forma, una de las ventajas de la técnica A-FCDCN es su bajo coste computacional en el receptor.

La compensación del ruido de memoria por la técnica A-FCDCN debe usarse en conjunción con la técnica de reconstrucción de ráfagas, en este caso, la interpolación lineal. Esta última debe aplicarse tras la corrección realizada por el algoritmo A-FCDCN, de forma que se asegure el uso de una versión corregida del primer vector de características después de la ráfaga.

4.3.3. Compensación del ruido del codec

De forma accesoria, la técnica A-FCDCN puede emplearse para la compensación parcial del ruido introducido por la codificación. Anteriormente, los vectores de corrección eran estimados comparando siempre voz sintetizada, transmitida en condiciones de canal limpio y ruidoso (con ráfagas de tramas erróneas). Si en vez de comparar con voz sintetizada, comparásemos con voz que no ha sufrido un proceso de codificación y decodificación, podríamos aliviar, además del ruido de memoria, la distorsión introducida por el codec.

La compensación del ruido del codec puede extenderse a aquellas tramas no afectadas por el ruido de memoria (el principio de la frase hasta la primera ráfaga y tramas posteriores a las del ruido de memoria) mediante la aplicación de un nuevo conjunto de correcciones $\mathbf{r}(q)$. Este conjunto se obtiene comparando voz codificada y sin codificar, no dependiendo de las ráfagas (parámetros l y t) como el anterior, tan sólo del índice de cuantización del vector. Los vectores afectados por el ruido del codec pero no por el ruido de memoria (y que, por lo tanto, aún no han sido compensados) son modificados de la forma,

$$\hat{\mathbf{x}} = \mathbf{y} + \mathbf{r}(q) \tag{4.11}$$

Denominaremos *A-FCDCN extendido* a esta extensión del algoritmo A-FCDCN para el tratamiento adicional del ruido de codificación. Al igual que antes, la interpolación se realiza tras la aplicación de esta versión extendida del A-FCDCN.

4.3.4. Resultados experimentales sobre EFR

Las técnicas descritas anteriormente han sido aplicadas a una arquitectura NSR sobre una red GSM. El marco experimental de las pruebas coincide con el descrito en el capítulo anterior (sección 3.6.2). Esto es, la señal de voz se adquiere en el emisor y se transmite, empleando el codec EFR y su canal asociado (TCH/EFS), al receptor, donde se realiza el reconocimiento sobre voz decodificada. Para la simulación del canal se emplean los patrones de error EP1, EP2 y EP3, junto con la condición de canal limpio. Aunque en estas pruebas se emplea el codec EFR, los resultados pueden extrapolarse con facilidad a otros codecs CELP.

Previamente a la aplicación de la técnica A-FCDCN, es necesario decidir la cuantización vectorial a usar, así como los limites máximos para l y t. En nuestra implementación cada vector de características es fragmentado en siete subvectores, cada uno con dos características (MFCCs 1 y 2, 3 y 4,..., 11 y 12, y el MFCC0 y el logaritmo de la energía), coincidiendo con la codificación aplicada por el front-end estándar de DSR. De hecho, se han empleado los cuantizadores SVQ suministrados por el estándar de ETSI para obtener cada índice de cuantización q. A fin de limitar los requerimientos de memoria, la longitud máxima de las ráfagas se ha fijado en cinco tramas, reutilizándose los vectores de corrección para longitudes $l = 5$ en caso de que aparezca una ráfaga con $l > 5$. Esto se debe a que, como se mostró en la figura 3.13, el efecto sobre el reconocimiento de estas ráfagas (superiores a 5 tramas) es despreciable. Adicionalmente, se asumirá que el ruido de memoria debido a las ráfagas no sobrepasa 20 vectores de características ($t = 20$), ya que para valores mayores de t los factores de corrección obtenidos son prácticamente nulos.

La tabla 4.1 muestra la precisión obtenida durante el reconocimiento con DSR, EFR, EFR con interpolación, EFR con interpolación y A-FCDCN, y EFR con interpolación y A-FCDCN extendido, bajo las condiciones de canal limpio, EP1, EP2 y EP3. Puede observarse que, a pesar de la simplicidad de la interpolación propuesta, aplicando únicamente esta técnica (4ª columna) puede mejorase significativamente la precisión del reconocimiento con EFR. Sin embargo, los resultados siguen siendo inferiores a los obtenidos por DSR. Esto se debe a que aún está presente, en la señal sintetizada, el ruido de memoria. Si se mitigan los efectos tanto del ruido de ráfaga como del ruido de memoria, a través de la aplicación conjunta de A-FCDCN e interpolación (5ª columna), EFR puede aproximar el rendimiento de DSR para las condiciones EP1 y EP2, e incluso mejorarlo para la condición EP3. Adicionalmente, si se emplea interpolación y A-FCDCN extendido (6º columna), es

Canal	DSR	EFR	EFR Interp.	EFR Interpolación y A-FCDCN	EFR Interpolación y A-FCDCN extendido
Limpio	99.04	98.70	98.70	98.70	98.81
EP1	99.04	98.44	98.43	98.46	98.64
EP2	98.95	96.91	97.55	97.82	98.19
EP3	93.41	84.48	90.76	94.04	94.04

Tabla 4.1: Precisión del reconocimiento obtenida con DSR, EFR, EFR con interpolación, EFR con interpolación y A-FCDCN, y EFR interpolación y A-FCDCN extendido, bajo distintas condiciones de canal GSM.

posible obtener una ligera mejora de aquellas condiciones menos ruidosas (limpio, EP1 y EP2) al mitigar en parte la distorsión por codificación.

4.4. Transparametrización del estándar Enhanced Full Rate

Las técnicas propuestas en la sección anterior requieren el conocimiento de, al menos, la marca de trama incorrecta o BFI junto con la señal decodificada. Esta información podría obtenerse supuesta una red cooperativa en donde las muestras PCM obtenidas a partir de una trama incorrecta fueran marcadas de alguna forma, por ejemplo, empleando algún tipo de patrón sobre los bits menos significativos. Otra opción consistiría en el acceso directo a la secuencia de bits transmitidos (las posibles alternativas para realizar este acceso se describirán más adelante, en la sección 4.6). Si se supone este acceso, no sólo se puede decodificar la marca BFI, sino igualmente, el resto de parámetros que representan la voz codificada.

En este caso, en el que los parámetros del codec están disponibles, se puede aplicar una transparametrización de los parámetros de voz EFR. Mediante esta transparametrización, los parámetros del codec son convertidos directamente en vectores de características orientados al reconocimiento. El objetivo no es otro que evitar la síntesis de la señal que, como se mostró en el capítulo anterior, conduce a una propagación de los errores, dando lugar al ruido de memoria.

A continuación, tras examinar brevemente como se codifica la voz mediante el estándar EFR, a fin de conocer cuales son y cómo se codifican los parámetros que describen la voz, proponemos una técnica de transparametrización desde EFR a DSR. Esta técnica se caracteriza por la obtención de vectores de características completamente compatibles

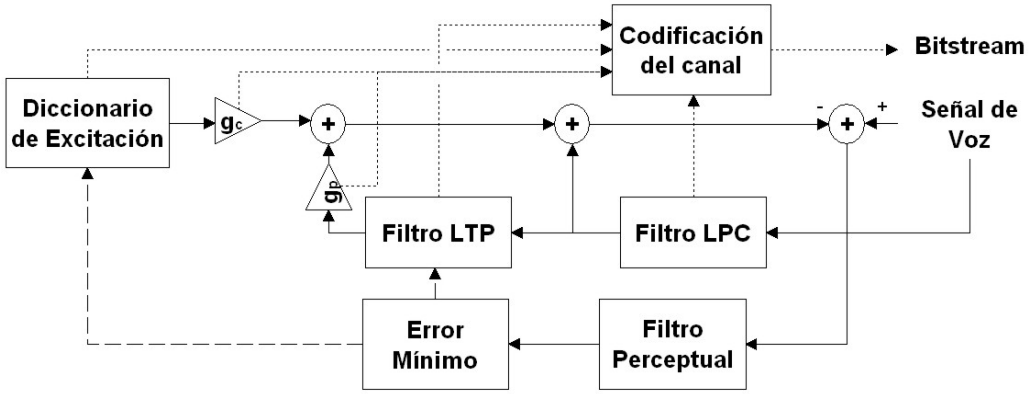


Figura 4.4: Diagrama simplificado del codificador EFR.

con el estándar DSR, esto es, no es necesaria aplicar modificación alguna (redefinir los vectores de características, reentrenamiento, ...) sobre los back-ends basados en DSR.

4.4.1. Codificación de la voz con GSM Enhanced Full Rate

Como mencionamos con anterioridad, el codificador de voz EFR [130] es un codificador híbrido de tipo ACELP. La figura 4.4 muestra un diagrama simplificado del proceso de codificación [131]. El codificador de voz trabaja sobre tramas o segmentos de señal de 20 ms, muestreada a 8 kHz. Cada trama de 160 muestras, a su vez, se divide en 4 subtramas, cada una de 5 ms (40 muestras), dando un procesamiento ligeramente diferente a las subtramas impares (primera y tercera) frente a las pares (segunda y cuarta).

De forma resumida, EFR emplea un modelo LPC de producción de voz que separa el tracto vocal de la excitación, codificando ambos elementos de forma independiente. La información sobre el tracto vocal se transmite mediante líneas de pares espectrales (*LSP–Line Spectral Pairs*). Se aplica dos veces un análisis LPC de orden 10, usando dos ventanas asimétricas distintas de 30 ms, sobre cada trama. Este par de ventanas utiliza muestras tanto de la trama actual como de la anterior (los últimos 10 ms), pero nunca de la trama siguiente. Las expresiones de estas ventanas vienen dadas por las siguientes

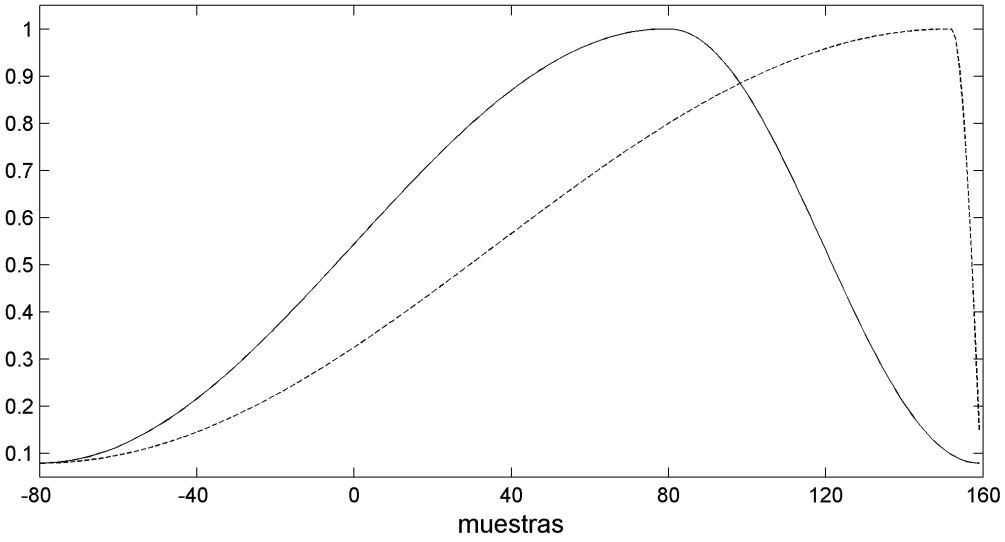


Figura 4.5: Ventanas empleadas para el cálculo de los coeficientes de autocorrelación en EFR.

ecuaciones:

$$w_I(n) = \begin{cases} 0{,}54 - 0{,}46\cos\left(\frac{\pi n}{L_1^{(I)}-1}\right), & n = 0, ..., L_1^{(I)} - 1 \\ 0{,}54 - 0{,}46\cos\left(\frac{\pi(n-L_1^{(I)})}{L_2^{(I)}-1}\right), & n = L_1^{(I)}, ..., L_1^{(I)} + L_2^{(I)} - 1 \end{cases}$$
$$w_{II}(n) = \begin{cases} 0{,}54 - 0{,}26\cos\left(\frac{2\pi n}{2L_1^{(II)}-1}\right), & n = 0, ..., L_1^{(II)} - 1 \\ \cos\left(\frac{2\pi(n-L_1^{(II)})}{4L_2^{(I)}-1}\right), & n = L_1^{(II)}, ..., L_1^{(II)} + L_2^{(II)} - 1 \end{cases} \tag{4.12}$$

La primera ventana consiste en dos medias ventanas de Hamming con tamaños diferentes. Se emplean los valores $L_1^{(I)} = 160$ y $L_2^{(I)} = 80$, para concentrar su peso en la segunda subtrama. La primera parte de la segunda ventana consiste en media ventana de Hamming, mientras que la segunda parte es un cuarto de ciclo de la función seno, concentrando su peso en la cuarta subtrama ($L_1^{(II)} = 232$, $L_2^{(II)} = 8$). En la figura 4.5 se muestran las dos ventanas sobre una trama de 20 ms (160 muestras).

Los coeficientes LPC se obtienen por el método de autocorrelación, mediante el algoritmo de Levinson-Durbin. Los coeficientes LPC obtenidos usando la ventana de análisis $w_1(n)$ son asociados a la segunda subtrama, mientras que los obtenidos usando la segunda ventana $w_2(n)$ se asocian a la cuarta subtrama. Estos dos conjuntos de parámetros LPC son convertidos en LSP. Las subtramas impares (primera y tercera) no llevan asociados

parámetros LPC, en el decodificador se emplea una interpolación de los parámetros LSP de las subtramas adyacentes,

$$\mathbf{LSP}_1^{(n)} = 0{,}5 \cdot \mathbf{LSP}_4^{(n-1)} + 0{,}5 \cdot \mathbf{LSP}_2^{(n)}, \quad \mathbf{LSP}_3^{(n)} = 0{,}5 \cdot \mathbf{LSP}_2^{(n)} + 0{,}5 \cdot \mathbf{LSP}_4^{(n)} \quad (4.13)$$

donde $\mathbf{LSP}_i^{(n)}$ es el vector LSP correspondiente a la subtrama i-ésima de la trama n. Una predicción de *media móvil* de primer orden se aplica sobre los dos conjuntos LSP obteniendo dos vectores residuales, $\mathbf{r}_1(n)$ y $\mathbf{r}_2(n)$, de la forma,

$$\mathbf{r}_1(n) = (\mathbf{LSP}_2(n) - \mathbf{LSP}_m) - 0{,}65 \cdot \hat{\mathbf{r}}_1(n-1), \quad (4.14)$$

$$\mathbf{r}_2(n) = (\mathbf{LSP}_4(n) - \mathbf{LSP}_m) - 0{,}65 \cdot \hat{\mathbf{r}}_2(n-1), \quad (4.15)$$

donde $\mathbf{LSP}_m$ es el vector LSP medio, y $\hat{\mathbf{r}}_1(n-1)$ y $\hat{\mathbf{r}}_2(n-1)$ los vectores residuales cuantizados de la trama anterior. Los vectores residuales de la trama actual son cuantizados conjuntamente empleando SMQ (*Split Matriz Quantization*) con 5 submatrices de dimensión 2×2 (dos elementos para cada conjunto).

Un filtro LTP (o de pitch) con un retardo fraccional se aplica sobre la señal de excitación, modelando los pulsos glotales y separando la señal adaptativa y la señal innovación. La búsqueda de los parámetros de la señal adaptativa se realiza con una combinación de bucle en lazo abierto y lazo cerrado, como la descrita en [131]. En las tramas impares se considera un filtro de pitch que varía dentro del rango $\left[17^{3/6}, 94^{3/6}\right]$ en incrementos de $1/6$ y de forma entera dentro del rango $[94, 143]$. Para las segunda y cuarta subtrama se emplea siempre una resolución de $1/6$ en el rango $\left[T - 5^{3/6}, T + 4^{3/6}\right]$, donde T es el entero más cercano al retardo fraccional de la subtrama anterior. El retardo del pitch se codifica por medio del entero más próximo (T en las subtramas impares), junto con un valor fraccional k.

La señal innovación se codifica mediante un índice que representa una secuencia de un diccionario de excitaciones. En EFR se emplea un diccionario algebraico basado en un diseño de permutaciones entrelazadas de pulsos simples (*ISPP–Interleaved Single-Pulse Permutation*). Éste consta de vectores de innovación bipolares, con amplitudes de $+1$ o -1. Sólo 10 pulsos en 40 posibles posiciones son distintos de cero. Las 40 posiciones en cada subtrama se organizan en 5 pistas, donde cada pista contiene dos pulsos. Para cada pista se definen 8 posibles posiciones entrelazadas en las que se puede colocar el pulso. La posición del pulso se indica con un índice sobre la lista de posiciones de la pista. La tabla 4.2 muestra la distribución de pistas, posiciones y pulsos para EFR.

Pista	*Pulso*	*Posiciones (índices)*
1	i_0, i_5	0, 5, 10, 15, 20, 25, 30, 35
2	i_1, i_6	1, 6, 11, 16, 21, 26, 31, 36
3	i_2, i_7	2, 7, 12, 17, 22, 27, 32, 37
4	i_3, i_8	3, 8, 13, 18, 23, 28, 33, 38
5	i_4, i_9	4, 9, 14, 19, 24, 29, 34, 39

Tabla 4.2: Distribución de pulsos por pista en el diccionario algebraico

Mediante un proceso de análisis por síntesis, en el que la señal de error obtenida entre la señal original y sintetizada es modificada por un filtro de peso, se busca en el diccionario el índice que mejor representa la excitación. El filtro de peso empleado para tal objetivo tiene la misma forma que el descrito en la sección 4.2.1:

$$W(z) = \frac{H(z/\gamma_1)}{H(z/\gamma_2)} \tag{4.16}$$

donde $H(z)$ es el filtro LPC, y $\gamma_1 = 0{,}9$ y $\gamma_2 = 0{,}6$ son pesos perceptuales.

Los pulsos obtenidos se codifican indicando sus posiciones mediante codificación Gray, a fin de mejorar la robustez frente a los errores del canal. Tan sólo se almacena el signo del primer pulso, ya que el signo del segundo pulso depende de su posición relativa respecto al primero. Si la posición del segundo pulso es menor a la del primero entonces tiene signo opuesto a este, y el mismo en caso contrario.

Finalmente, la ganancia de la señal adaptativa (ganancia adaptativa) se cuantiza escalarmente en el rango [0, 1.2], mientras que la ganancia de la señal de innovación (ganancia algebraica) se representa por el residuo de una predicción de media móvil de cuarto orden con coeficientes fijos.

4.4.2. Transparametrización de los coeficientes LPC

Como se ha visto, el codificador EFR opera sobre tramas de voz de 160 muestras divididas en 4 subtramas de 40 muestras. Cada trama EFR contiene dos conjuntos de LSP obtenidos por medio de dos ventanas asimétricas. Estas ventanas están centradas en la segunda y cuarta subtrama respectivamente. Por otro lado, el algoritmo de extracción de características de DSR se aplica sobre 200 muestras cada 80 muestras o, en terminos de subtramas, sobre 5 subtramas cada 2 subtramas. Dado que los conjuntos LSP y las tramas DSR se producen a la misma tasa (una cada 80 muestras, es decir, 2 subtramas), una asignación sencilla podría consistir en asignar un conjunto LSP para cada trama DSR.

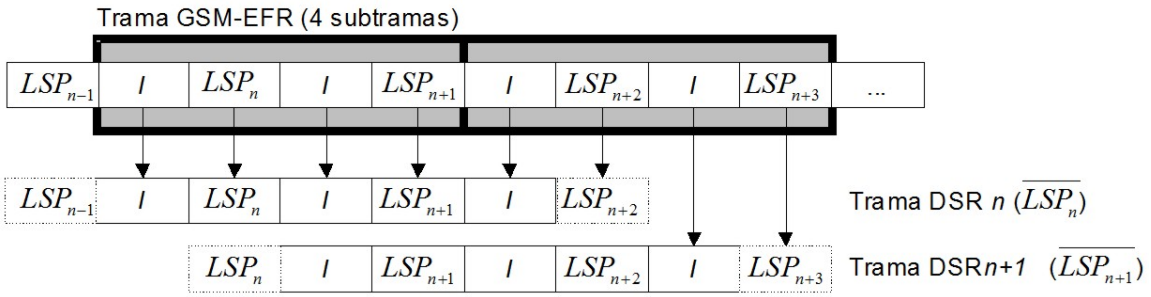


Figura 4.6: Asignación de los conjuntos LSP de EFR a cada ventana de análisis DSR.

Sin embargo, esta asignación sólo describiría de forma precisa la segunda subtrama de la ventana de análisis de DSR. Mediante un promediado de los conjuntos LSP involucrados en la síntesis de cada subtrama, puede darse una representación más precisa para la ventana de análisis,

$$\overline{\mathbf{LSP}_n} = \frac{\mathbf{LSP}_{n-1} + 4\mathbf{LSP}_n + 4\mathbf{LSP}_{n+1} + \mathbf{LSP}_{n+2}}{10} \tag{4.17}$$

donde $\overline{\mathbf{LSP}_n}$ es el conjunto LSP asignado a la n-ésima trama DSR y $\mathbf{LSP}_n$ es el n-ésimo conjunto LSP recibido. La figura 4.6 muestra como quedan asignados los LSP a cada subtrama. Las subtramas marcadas con *'I'* indican que el conjunto de LSP correspondiente se obtiene por interpolación de los adyacentes. A cada ventana de análisis DSR le corresponde el conjunto LSP promedio entre todos los conjuntos LSP de sus correspondientes subtramas.

Los coeficientes MFCC(k) (k=0,...,12) pueden obtenerse usando el procedimiento descrito en el estándar de DSR, pero sustituyendo el espectro FFT por el espectro LPC,

$$|H(\omega_i)| = \sigma |H'(\omega_i)| \qquad (\omega_i = 2\pi i/N;\ i = 0, \ldots, 255) \tag{4.18}$$

donde σ es la ganancia LPC y $|H'(\omega_i)|$ es el espectro LPC normalizado en ganancia,

$$|H'(\omega_i)| = \frac{1}{\left|1 + \sum_{l=1}^{10} a_l e^{-j\omega_i l}\right|} \qquad (\omega_i = 2\pi i/N;\ i = 0, \ldots, 255) \tag{4.19}$$

donde a_l son los coeficientes LPC.

Si se aplica un banco de filtros triangular con $M = 23$ filtros a $|H(\omega_i)|$, junto con una

transformación DCT a las salidas logarítmicas del banco de filtros, se deriva que,

$$MFCC(k) = \begin{cases} M \log \sigma + MFCC'(k) & k = 0 \\ MFCC'(k) & k = 1, \ldots, 12 \end{cases} \tag{4.20}$$

donde $MFCC'(k)$ representa el coeficiente cepstral correspondiente al espectro normalizado LPC, $|H'(\omega_i)|$.

4.4.3. Transparametrización de la energía

Para obtener el $MFCC(0)$, así como la energía logarítmica, es necesario calcular la energía media de la señal excitación del filtro LPC. Como en cualquier otro codificador de tipo CELP (ver figura 4.2), la señal excitación para cada subtrama EFR se obtiene como,

$$u(n) = g_p v(n) + g_c c(n) \tag{4.21}$$

donde $v(n)$ y g_p son, respectivamente, la señal de excitación del diccionario adaptativo (o vector adaptativo) y la ganancia adaptativa, mientras que $c(n)$ es la señal del diccionario de innovación (o vector innovación) y g_c la ganancia algebraica. Así pues, la energía de excitación de cada subtrama m, $\sigma^2(m)$, puede obtenerse como,

$$\sigma^2(m) = g_p^2 E_d(m) + g_c^2 E_c(m) \tag{4.22}$$

donde $E_d(m)$ es la energía del vector adaptativo y $E_c(m)$ la del vector innovación.

Las valores de las ganancias g_p y g_c pueden decodificarse de forma directa a partir de la secuencia de bits (*bitstream*), mientras que $E_c(m)$ puede obtenerse del vector innovación $c(n)$ $(n = 0, \ldots, 39)$ como,

$$E_c(m) = \frac{1}{N_{sf}} \sum_{n=0}^{N_{sf}-1} c^2(n) \tag{4.23}$$

donde $N_{sf} = 40$ es el tamaño de la subtrama. Sin embargo, de acuerdo con la descripción del codec, el vector innovación $c(n)$ se representa mediante un código algebraico de 10 pulsos, distribuidos entre las 40 posiciones. Puesto que las amplitudes de estos pulsos son +1 o -1, y la distribución de estos pulsos no afecta a la energía de la señal innovación, ésta puede conocerse a priori, siendo 1/4.

Para calcular $E_d(m)$ es necesario conocer el vector adaptativo $v(n)$. Sin embargo, éste no puede decodificarse directamente del bitstream, ya que se obtiene mediante un filtrado LTP con pitch fraccional aplicado sobre la señal excitación previa $u(n)$. Para simplificar

los cálculos, consideraremos sólo el entero más próximo al retardo de pitch, T, el cual puede obtenerse directamente del bitstream. De esta forma el filtrado LTP se convierte en un simple desplazamiento de la señal.

Puesto que la señal no llega a sintetizarse en ningún momento, la excitación previa $u(n)$ no está disponible. Supongamos entonces que la energías $E(m)$ de las subtramas previas sí están disponibles. Bajo esta suposición, la energía de $v(n)$ correspondiente a la subtrama m, $E_d(m)$, puede aproximarse a través de una media pesada de las energías correspondientes a las subtramas previas, de donde el vector adaptativo sería extraído, dada por,

$$E_d(m) = \frac{E\left(m - \lfloor \frac{T}{N_{sf}-1} \rfloor\right) N_{prev} + E\left(m - \lfloor \frac{T}{N_{sf}} \rfloor\right) (N_{sf} - N_{prev})}{N_{sf}} \tag{4.24}$$

donde $N_{prev} = T \bmod N_{sf}$. Aplicando entonces las expresiones (4.22), (4.23) y (4.24) podemos conocer la energía $E(m)$ de la subtrama. La suposición inicial se hace entonces innecesaria, ya que, operando de modo iterativo, podemos disponer de las energías $E(m)$ de las subtramas previas.

Como puede observarse, la expresión (4.24) sólo es válida para retardos de pitch superiores o iguales a 40 muestras. Cuando $T < 40$ el decoder EFR usa muestras ya calculadas de la señal excitación de la subtrama actual para obtener el vector adaptativo de la propia subtrama. Obviamente, esto no puede hacerse trabajando en términos de subtramas, por tanto, en este caso particular, la energía del vector adaptativo de la subtrama m sólo puede ser aproximada mediante la energía de la excitación correspondiente a la subtrama previa $m - 1$. Esto revela dos problemas en nuestra aproximación:

- El retardo mínimo de pitch considerado en EFR es de $T = 17$ muestras. En este caso, la energía de la señal adaptativa sería toscamente aproximada por una subtrama de 40 muestras cuando realmente sólo intervienen sus 17 últimas muestras durante el cálculo del vector adaptativo.

- Igualmente, la energía de excitación dentro de una subtrama podría estar balanceada al principio o al final de ésta. Debido al promediado implícito, podría obtenerse una mala estimación de $E_d(m)$.

Para aliviar estos problemas proponemos un incremento de la resolución temporal de las ecuaciones (4.23) y (4.24), de forma que se consideren medias subtramas ($N_{sf} = 20$) en vez de subtramas completas. En este caso, la energía de la señal innovación de la media

subtrama no puede conocerse a priori (depende de la distribución de pulsos). Sin embargo, su cálculo es muy eficiente, gracias a la forma en que se codifican los pulsos del código algebraico (indicando la posición).

Empleando medias subtramas, la ecuación (4.24) puede emplearse hasta que $T < 20$, por tanto, en el peor caso, estaremos aproximando la energía media de 17 muestras por aquella obtenida sobre 20 muestras. Esta estimación es mucho mejor en comparación con la primera propuesta.

Finalmente, la energía media de la excitación σ^2 de cada trama DSR se obtiene como la suma de las energías medias de excitación de las 5 subtramas que la componen (esto es, de las 10 medias subtramas). Conocido σ^2, el coeficiente $MFCC(0)$ se calcula por medio de la ecuación (4.20), y el logaritmo de la energía como,

$$logE = \log \frac{1}{2\pi} \int_{-\pi}^{\pi} \left| \frac{\sigma}{1 + \sum_{l=1}^{10} a_l e^{-j\omega l}} \right|^2 d\omega \tag{4.25}$$

4.4.4. Tratamiento de los errores residuales

Como se comentó en la sección 4.4.1, una predicción de media móvil de primer orden se aplica durante la cuantización de los coeficientes LSP. Esto implica que los coeficientes LSP de la primera trama EFR recibida tras una ráfaga no puedan decodificarse correctamente, ya que se hacen necesarios los residuos de la predicción en la trama inmediatamente anterior (que se ha perdido). Debido a esto, los coeficientes cepstrales desde el orden 1 al 12 ($MFCC(1-2)$) de las dos primeras tramas DSR calculadas después de la ráfaga estarán dañados, como se deduce de la ecuación (4.20). Algo similar sucede con los parámetros $MFCC(0)$ y $\log E$. En este caso, aunque también están presentes ciertos artificios de codificación que introducen errores (análogos a los descritos en la decodificación de los LSP), la principal causa de degradación se debe al cómputo recursivo de la señal adaptativa (que emplea las T muestras anteriores). Este cálculo recursivo está presente también en la ecuación (4.24), provocando que la señal de excitación al completo y su energía, σ^2, contenga errores durante un periodo de tiempo mayor. Durante este periodo, los parámetros $MFCC(0)$ y $\log E$ siguen inevitablemente afectados por el ruido de memoria. En la figura 4.7 se resume el efecto de una ráfaga sobre los vectores de características obtenidos por transparametrización.

A pesar de las degradaciones descritas, los coeficientes $MFCC(1-12)$ de la tercera trama DSR después de la ráfaga y siguientes son correctos. Como puede observarse, la transparametrización permite la eliminación parcial del ruido de memoria ($MFCC(1-$

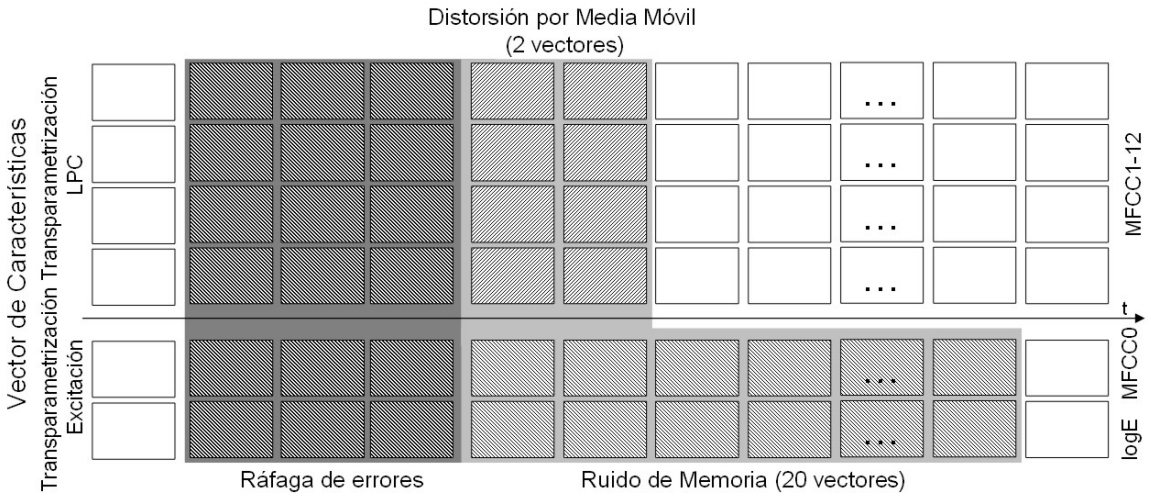


Figura 4.7: Transparametrización EFR/DSR y efecto de los errores de canal en los vectores de características.

12)). Esto constituye la principal ventaja de la aproximación por transparametrización frente a la extracción de características sobre voz decodificada.

El ruido de memoria restante (sobre $MFCC(0)$ y $\log E$), así como la degradación presente en los coeficientes cepstrales $MFCC(1-12)$ en las dos primeras tramas DSR tras la ráfaga, puede compensarse por medio de la técnica A-FCDCN, ya descrita en la sección 4.3.2. Como ya hiciéramos en la sección 4.3.3, los vectores de corrección se deben obtener comparando con voz limpia sin codificar. De esta forma también compensaremos parcialmente la distorsión introducida por el codec. En concreto introduciremos un conjunto de vectores de corrección A-FCDCN para cada uno de los siguientes elementos:

1. Los coeficientes cepstrales $MFCC(1-12)$ pertenecientes a las dos primeras tramas DSR tras una ráfaga de errores.

2. Los parámetros $MFCC(0)$ y $logE$ de las 20 primeras tramas DSR tras una ráfaga.

3. Una compensación A-FCDCN para la compensación del ruido del codec aplicada al resto de vectores de características DSR de la frase.

La primera compensación está dedicada a la reducción de la degradación debida a la codificación mediante predicción MA sobre los conjuntos LSP. El segundo conjunto de correcciones está destinado a la compensación del ruido de memoria que afecta a los parámetros $MFCC(0)$ y $\log E$. Por último, mediante el tercer conjunto de correcciones, se aplica una compensación parcial de la distorsión del codec sobre el resto de parámetros.

Canal	DSR	EFR	EFR mejorado	T-EFR	T-EFR mejorado
Limpio	99.04	98.70	98.81	98.73	98.88
EP1	99.04	98.44	98.64	98.68	98.82
EP2	98.95	96.91	98.19	98.26	98.57
EP3	93.41	84.48	94.04	94.08	95.57

Tabla 4.3: Precisión de reconocimiento obtenida con DSR, EFR, EFR mejorado, EFR transparametrizado (T-EFR) y transparametrización mejorada (T-EFR mejorado).

4.4.5. Resultados experimentales

El transparametrizador básico desarrollado para EFR, al que nos referiremos como *T-EFR*, así como el transparametrizador con todas las mejoras introducidas en la sección 4.4.4, al que nos referiremos como *T-EFR mejorado*, han sido evaluados en un marco experimental similar al descrito en la sección 3.6.2. Éste incluye la adquisición, y codificación mediante EFR, de la voz en el emisor y la transmisión por el canal asociado (TCH/EFS), sobre el que se simulan errores (patrones de error EP1, EP2 y EP3, y canal limpio). En el receptor, el conjunto formado por el decodificador EFR y el extractor de parámetros se sustituye por el transparametrizador (mejorado o no, según corresponda). Finalmente, los vectores de características correspondientes a las ráfagas (considerados erróneos) se mitigan aplicando un interpolación lineal entre el último y primer vector, previo y posterior respectivamente, a la ráfaga.

Los resultados sobre la precisión de reconocimiento obtenidos mediante el transparametrizador básico (experimento T-EFR), bajo las diferentes condiciones de canal, se muestran en la tabla 4.3, junto con los obtenidos con las aproximaciones DSR, EFR, y EFR mejorado, esto es, con todas las mejoras introducidas en la sección 4.3 (tabla 4.1, experimento *EFR Interpolación y A-FCDCN extendido*). Puede observarse que la aproximación por transparametrización es ligeramente superior al EFR mejorado (excepto en condiciones de canal limpio).

Los resultados obtenidos pueden mejorarse substancialmente si, previamente a la interpolación, se realiza una compensación de los ruidos residuales empleando el transparametrizador mejorado. Los resultados obtenidos mediante esta aproximación pueden encontrarse también en la tabla 4.3 (experimento T-EFR mejorado). Como puede observarse, el rendimiento de esta solución aproxima al obtenido por DSR bajo la condición limpia, EP1 y EP2. Bajo la condición EP3, esta técnica de transparametrización con compensado de ruido residual supera en más de un 2 % a la solución DSR.

4.5. Transparametrización del estándar Adaptive MultiRate

El codec AMR es el más avanzado de la familia GSM. Este codec está basado en el concepto de tasa de bits variable adaptativa. Al contrario que otros codificadores menos avanzados, AMR realiza una asignación variable de los bits dedicados a la codificación de la voz y a la codificación del canal. Conforme la calidad del enlace de radio se degrada, se reduce la tasa de bits del codificador de voz a favor del codificador de canal que, al disponer de más ancho de banda, puede llevar a cabo una protección más efectiva de los parámetros de voz. Frente a la recepción de parámetros de voz erróneos, que deben ser descartados y sustituidos por un algoritmo de mitigación fuertemente limitado por la latencia, es preferible disponer de esos mismos parámetros codificados con menor tasa de bits, aunque esto conduzca a la síntesis de una señal de voz degradada. Este hecho asegura el buen comportamiento de la estrategia de adaptación dinámica de la tasa de bits. Como consecuencia, AMR ofrece una mejor calidad de voz en condiciones de canal ruidoso.

El codificador de voz AMR es un codificador de tipo ACELP que, para proporcionar una tasa de bits variable, integra 8 modos de operación a 4.75, 5.15, 5.9, 6.7, 7.95, 10.2 y 12.2 kbps. El codificador es capaz de cambiar dinámicamente entre estos modos cada 20 ms, esto es, cada vez que codifica una trama, a petición de la estrategia de control. Esta estrategia de control monitoriza el estado del enlace de radio y decide la tasa de bits más apropiada de acuerdo con la condición del canal. La información sobre el modo empleado para la codificación de la voz se transmite en banda, es decir, utilizando el mismo canal por el que se transmiten los datos de voz. Los distintos modos de operación del codec AMR se basan en los mismos principios de codificación que EFR. De hecho, el modo de mayor calidad (12.2 kbps) es casi idéntico a EFR. El resto de modos recurren a los mismos parámetros para representar la voz (LSP, retardo del pitch, ganancias adaptativa y de innovación y código algebraico) pero reducen el número de bits dedicados a la codificación de cada uno de ellos. Esta reducción da lugar un calidad de voz inferior, pero preferible a la obtenida mediante un algoritmo de mitigación para tramas descartadas.

La disponibilidad de un mayor ancho de banda permite una codificación de canal más robusta para de los parámetros de voz, pero no asegura la correcta recepción de estos bajo cualquier condición. Además, aunque la estrategia de control estableciera el modo de operación óptimo para un determinado estado de canal, la decisión debe tomarse antes

de que la voz sea codificada y transmitida, basándose por tanto en una predicción. Esto implica que AMR no está exento de la recepción de tramas incorrectas, aunque, como es de esperar, el número de éstas será significativamente inferior. Por ello, también en AMR se establece un mecanismo de mitigación de tramas incorrectas que consiste, como para el resto de codecs, en una sustitución de los parámetros junto con un apagado progresivo de la señal (estándar GSM 6.91 [62]).

Dado que ambos codecs se basan en los mismos principios de codificación (ACELP), la transparametrización descrita para EFR puede extenderse al codec AMR sin implicar cambios mayores. Por esta misma razón, la transparametrización de AMR hacia el estándar DSR incide positivamente sobre el reconocimiento, ya que, al igual que ocurría con EFR, el efecto del ruido de memoria ocasionado por las tramas erróneas se ve reducido. Sin embargo, resulta también relevante la degradación inducida por la distorsión del codec ya que, en canales ruidosos, se trabaja con una tasa de bits menor. Frente a esta degradación, la transparametrización puede también proporcionar incrementos en la precisión, como veremos más adelante.

4.5.1. Extensión de la transparametrización a AMR

Al igual que EFR, AMR es un codificador de voz tipo ACELP, pero a diferencia de aquel, integra 8 modos de operación que proporcionan una tasa de bits variable. Es por ello que, aunque los mismos conceptos empleados para la transparametrización de EFR pueden ser aplicados, es necesario una adaptación especifica a cada uno de los modos. Mediante el indicador de modo de operación, que es transmitido en banda junto con los demás parámetros, el transparametrizador puede adaptarse dinámicamente a la codificación de cada trama, de la misma forma que lo haría el decodificador AMR.

A continuación, examinaremos las diferencias de cada unos de los modos de operación con respecto a EFR, describiendo los cambios necesarios en el transparametrizador. Siguiendo la notación empleada por el estándar identificaremos cada modo por la tasa de bits que le caracteriza.

Coeficientes LPC

La obtención de los coeficientes LPC es una de las diferencias más importantes entre el modo 12.2 (el de mayor calidad) y el resto de modos (10.2 - 4.75). En el modo de alta calidad se extraen dos conjuntos de coeficientes LPC por cada trama, empleándose las ventanas de 30 ms descritas en la ecuación 4.12. Estas ventanas concentran su peso en

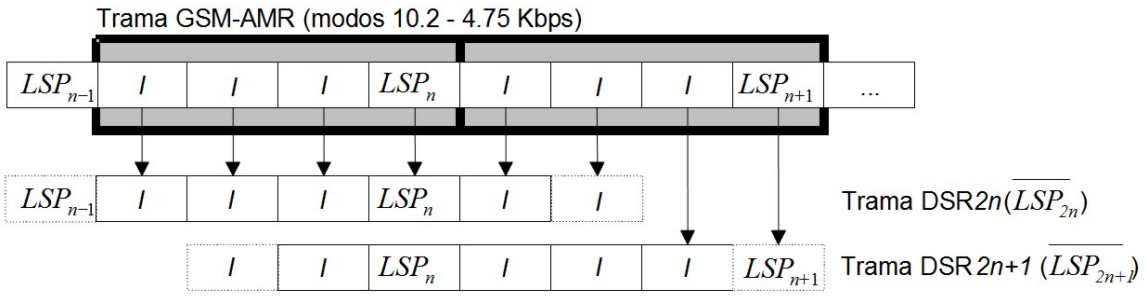


Figura 4.8: Asignación de los conjuntos LSP de los modos AMR 10.2 - 4.75 kbps a cada ventana de análisis DSR.

la segunda y cuarta subtrama, como lo hiciera el codec EFR. Para el resto de modos, en cambio, se obtiene un único conjunto de coeficientes LPC. Este conjunto se obtiene empleando una ventana asimétrica que concentra su peso en la cuarta subtrama y, al contrario que EFR y el modo 12.2, incluye 40 muestras de la trama siguiente (en lo que se denomina *lookahead*). Al igual que en EFR, en todos los modos los coeficientes LPC se transforman en LSP que finalmente son codificados mediante una predicción MA de primer orden.

Esto afecta al promediado de conjuntos LSP que realizábamos para describir el espectro LPC de la trama DSR. Para el modo 12.2 es posible reutilizar el promediado descrito por la ecuación 4.17. Sin embargo, para el resto de modos, es necesario definir uno nuevo ,acorde con la nueva distribución de conjuntos LSP. Distinguiremos entonces entre tramas DSR pares e impares, aplicándose el siguiente par de ecuaciones:

$$\overline{\mathbf{LSP}_{n*2}} = \frac{6\mathbf{LSP}_{n-1} + 13\mathbf{LSP}_n + \mathbf{LSP}_{n+1}}{20} \tag{4.26}$$

$$\overline{\mathbf{LSP}_{n*2+1}} = \frac{\mathbf{LSP}_{n-1} + 13\mathbf{LSP}_n + 6\mathbf{LSP}_{n+1}}{20} \tag{4.27}$$

donde $\overline{\mathbf{LSP}_n}$ es el conjunto LSP asignado a la n-ésima trama DSR y $\mathbf{LSP}_n$ es el n-ésimo conjunto LSP recibido. Nótese que, para estos modos de operación, los conjuntos LSP se generan a la mitad de la tasa (uno cada 160 muestras) con la que se producen tramas DSR (una cada 80 muestras). La figura 4.8 muestra como quedan asignados los LSP a cada subtrama.

Por último, otra diferencia con respecto a EFR radica en que todos los modos incorporan un detector de resonancia en el espectro LPC. El objetivo de este detector es la

identificación de posibles problemas que deriven en la aparición de filtros inestables en segmentos de señal altamente correlados. Este monitor no modifica los coeficientes, tan sólo informa del resultado de la detección, por lo que, por el momento, carece de interés en el transparametrizador.

Vector Adaptativo

El vector adaptativo se obtiene mediante la aplicación de un filtro LTP fraccional sobre la excitación previa. En el modo 12.2, este filtro fraccional tiene el mismo rango y resolución que en EFR. Igualmente se distingue entre subtramas impares, en las que se codifica el retardo de forma absoluta, y subtramas pares, en las que se codifica de forma relativa a la subtrama previa.

Para el resto de modos se modifica la resolución y rangos del retardo de pitch. En todos estos modos, el retardo de pitch se codifica de forma absoluta para la primera y tercera subtramas, y de forma relativa en la segunda y cuarta subtramas, con excepción de los modos 5.15 y 4.75, en donde sólo se codifica de forma absoluta en la primera subtrama (en las restantes subtramas se codifica de forma relativa). Para las subtramas en que el pitch se codifica de forma absoluta, se emplea una resolución de 1/3 en el rango $\left[19^{1/3}, 84^{2/3}\right]$ y valores enteros dentro del rango $[85, 143]$. En las tramas con codificación relativa los rangos considerados dependen del modo:

- Modos 10.2 y 7.40. Para la segunda y cuarta se emplea siempre una resolución de 1/3 en el rango $[T - 52/3, T + 42/3]$, donde T es el entero más cercano al retardo fraccional de la subtrama anterior.
- Modo 7.95. Como en los modos anteriores pero un rango $\left[T - 10^{2/3}, T + 9^{2/3}\right]$.
- Modo 6.7 y 5.9. Para la segunda y cuarta se emplea una resolución entera en el rango $[T - 5, T + 4]$, con una resolución de 1/3 en el rango $[T - 12/3, T + 2/3]$.
- Modo 5.15 y 4.75. Se aplican los mismos rangos anteriores (modos 6.7 y 5.9), pero para la segunda, tercera y cuarta subtrama.

La diferencia de resolución empleada para el retardo fraccional no supone un problema para la trasparametrización, ya que la energía del vector adaptativo se aproxima empleando el entero más próximo al retardo del pitch. Igualmente, el rango mínimo permitido por todos los modos es superior a 17 muestras, por lo que no es necesario incrementar

la resolución de las ecuaciones (4.23) y (4.24) para considerar unidades inferiores a medias subtramas. La única modificación necesaria consiste en la correcta decodificación del retardo de pitch (T) en cada modo.

Vector Innovación

Al igual que en EFR, la estructura del diccionario algebraico esta basada en un diseño ISSP, con pulsos de amplitud ± 1, para todos los modos. La única diferencia radica en el número de pistas en que es divida la subtrama, el número de posiciones y el número de pulsos que contiene cada una de ellas:

- Modo 12.2. Idéntico a EFR, divide la subtrama en 5 pistas de 8 posiciones con dos pulsos por pista. Cada pista distribuye las posiciones de forma creciente y uniforme (la posición del pulso se codifica con un índice). Por ello, el cálculo de la ecuación (4.23) puede optimizarse teniendo en cuenta si cada pulso ésta antes (primera media subtrama) o después (segunda media subtrama) del cuarto índice de cada pista (ver tabla 4.2).

- Modo 10.2. Las 40 posiciones de la subtrama se dividen en 4 pistas con 10 posiciones, cada una con dos pulsos. La ecuación (4.23) se optimiza como antes pero considerando el quinto índice de cada pista.

- Modos 7.95 y 7.40. Se emplean 4 pistas, las 3 primeras con 8 posiciones y la última con 16, cada una con un pulso. En este caso se considera el cuarto índice en las tres primeras pistas, mientras que la posición del último pulso debe ser completamente decodificada.

- Modo 6.7. Se establecen 3 pistas con un único pulso, como en los modos anteriores, no se distribuyen uniformemente las posiciones, la primera pista cuenta con 8 posiciones, mientras que las dos últimas con 16. Hay un único pulso por pista. Sólo puede aplicarse la optimización en la primera pista, considerando el cuarto índice.

- Modo 5.9. Las 40 posiciones se dividen en una pista de 16 posiciones y otra con las 24 restantes, cada una con un pulso. En este modo es necesario decodificar completamente la posición de los pulsos.

- Modos 5.15 y 4.75. Se emplean 5 pistas, pero se distinguen dos subconjuntos de dos pistas por cada subtrama con un pulso por cada pista. Mediante un bit se identifica

el subconjunto empleado. Al igual que antes, es necesario decodificar completamente la posición de los pulsos.

Ganancia Adaptativa y Algebraica

El cálculo de la ganancia adaptativa se realiza en todos los modos de operación de igual forma que en EFR, a la vez que se obtiene el retardo de pitch. Sin embargo, la forma en que esta se codifica difiere según el modo empleado. Para los modos 12.2 y 7.9 la ganancia adaptativa se codifica de forma escalar, mientras que para el resto de modos, ésta se codifica vectorialmente junto con el error residual de predicción de la ganancia algebraica (como en EFR, se aplica a un predictor MA de cuarto orden con coeficientes fijos).

Una función adicional, no incluida en EFR, es el control de las ganancias adaptativa y algebraica en todos los modos. Si el detector de resonancia en el espectro está activo, se aplica una limitación de la ganancia adaptativa y algebraica. Esta limitación se aplica en el codificador, por lo que no es posible evitarla mediante la transparametrización.

Como ocurría con el vector adaptativo, la transparametrización no tiene que modificarse, salvo para la correcta decodificación de los parámetros.

4.5.2. Transparametrización y distorsión del codec AMR

El codec AMR basa la protección frente al ruido del canal en una codificación de la voz con una tasa de bits variable. Gracias a ésta, se incrementa el ancho de banda disponible para la protección de los parámetros conforme el canal se degrada. Esto implica una cuantización más ruda de los parámetros de voz que da lugar a una mayor degradación de la voz o, lo que es lo mismo, a una mayor distorsión debida al codec. Así, a diferencia del resto de los codificadores GSM, la degradación debida al canal se traduce en AMR en una distorsión de la voz debida al codec. Esta distorsión, como es de esperar, afecta negativamente al reconocimiento.

La transparametrización puede también ser útil frente a este tipo de degradación. Esto se justifica en cómo se reduce el número de bits dedicados a la codificación de cada parámetro de voz y al uso que hace la transparametrización de cada uno de ellos. Durante la transparametrización se distinguen dos procesamientos bien diferenciados, el procesado de los coeficientes LPC, del que se obtienen las características con información del tracto vocal ($MFCC1-12$), y el procesado del resto de parámetros (retardo de pitch, código algebraico y ganancias), que da lugar a las características con información energética ($MFCC0$ y $LogE$). La tabla 4.4 muestra el número de bits dedicados a cada parámetro

Modo	Envolvente Espectral	Energía				Tasa
	LSP	Pitch	Código C. Algebraico	Ganancias (adap.+alg.)	Total	
12.2	38	30	140	16+20	206	244
10.2	26	26	124	28	178	204
7.95	27	28	68	16+20	132	159
7.40	26	26	68	28	122	148
6.70	26	24	56	28	108	134
5.90	26	24	44	24	92	118
5.15	23	20	36	24	80	103
4.75	23	20	36	16	72	95

Tabla 4.4: Asignación de bits por parámetros para cada trama AMR.

según el modo de operación. Como puede observarse, la reducción del número de bits es más agresiva sobre aquellos parámetros que representan la excitación (debe notarse que los modos 10.2-4.75 sólo tienen que codificar un conjunto LSP). Si se procede a la síntesis y análisis posterior de la voz, los efectos de la fuerte cuantización de estos parámetros pueden extenderse a todo el vector de características. Sin embargo, aplicando la transparametrización tan sólo es de esperar una reducción en la calidad sobre los coeficientes energéticos.

Para comprobar esta hipótesis, se han llevado a cabo una serie de pruebas de reconocimiento. El entorno experimental de estas pruebas está basado en aquel descrito en la sección 2.5 convenientemente modificado para incluir la transmisión de voz con AMR. En el emisor, el codificador AMR emplea uno de los 8 modos de operación disponibles para codificar la voz, mientras que en el receptor, o bien los parámetros son directamente transparametrizados a DSR, o bien se extraen los vectores DSR de la señal voz sintetizada a partir de ellos. Los resultados obtenidos empleando voz sintentizada se muestran en la figura 4.9. En ella puede observarse un comportamiento poco predecible del codec AMR, como ya describiera Hirsch [96], en donde modos con menor tasa de bits ofrecen mejor precisión en el reconocimiento (modos 4.75 y 7.40 frente a los modos 5.15 y 7.95, respectivamente). En cualquier caso, puede apreciarse una tendencia de mejora en la precisión de reconocimiento conforme se incrementa la tasa de bits.

Esta misma figura muestra los resultados obtenidos aplicando el transparametrizador descrito para los distintos modos. Resulta relevante el efecto positivo que tiene la transparametrización sobre el reconocimiento frente a la degradación debida a la codificación.

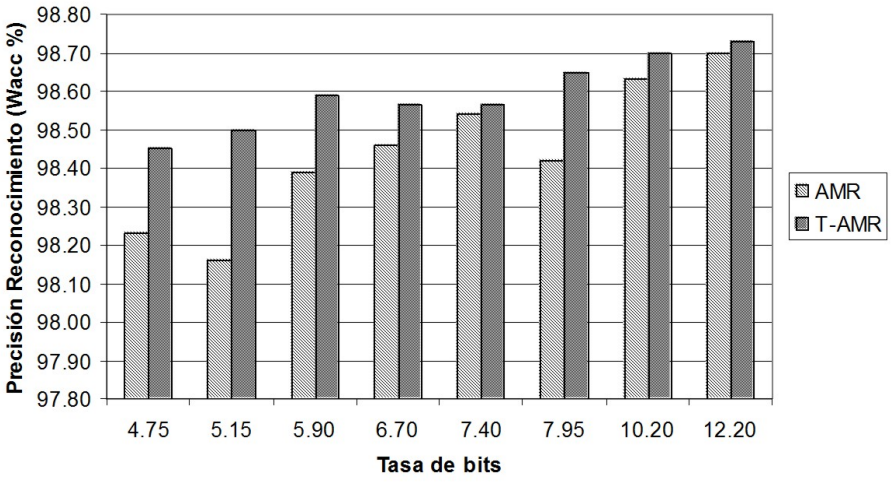


Figura 4.9: Precisión del reconocimiento para cada modo de operación de AMR empleando voz decodificada (AMR) o transparametrización (T-AMR).

4.6. Aplicación de Trasparametrización a redes GSM

En general, el acceso desde el servidor a los parámetros que codifican la voz, así como la marca BFI, no es posible. Antes de ser transportada por el enlace fijo, la voz es sintetizada y adaptada al estándar de 64 kbps utilizado por la red digital de telefonía (recomendación G.711 de la ITU-T con ley-A o ley-μ [132]). Esta conversión, denominada transcodificación, se realiza en la TRAU (Transcoder Rate Adaptor Unit), una entidad funcional que físicamente suele estar localizada en el controlador de estación base o BSC (ver sección 3.2.2). Así, en una configuración normal de llamada MS-MS (de móvil a móvil), la señal de voz es codificada en el MS emisor, transmitida por el enlace de radio y convertida a G.711 por la TRAU local. La voz codificada a 64 kbps se trasmite por la red fija, siendo codificada de nuevo por la TRAU remota y enviada por el enlace de radio al MS receptor, tal y como se muestra en la figura 4.10a.

Esto supone un problema a la hora de aplicar la transparametrización, puesto que en el servidor no se dispondría del bitstream. Lo mismo ocurre con las técnicas propuestas para la mejora del reconocimiento sobre voz codificada, en donde se requiere la marca BFI. Por esta razón, proponemos las siguientes alternativas:

- Una modificación menor en la TRAU permitiría que los segmentos de voz sintetizados a partir de tramas incorrectas fueran marcados de alguna forma. Esto permitiría el uso de las técnicas propuestas para la mejora del reconocimiento sobre voz codificada (figura 4.11a).

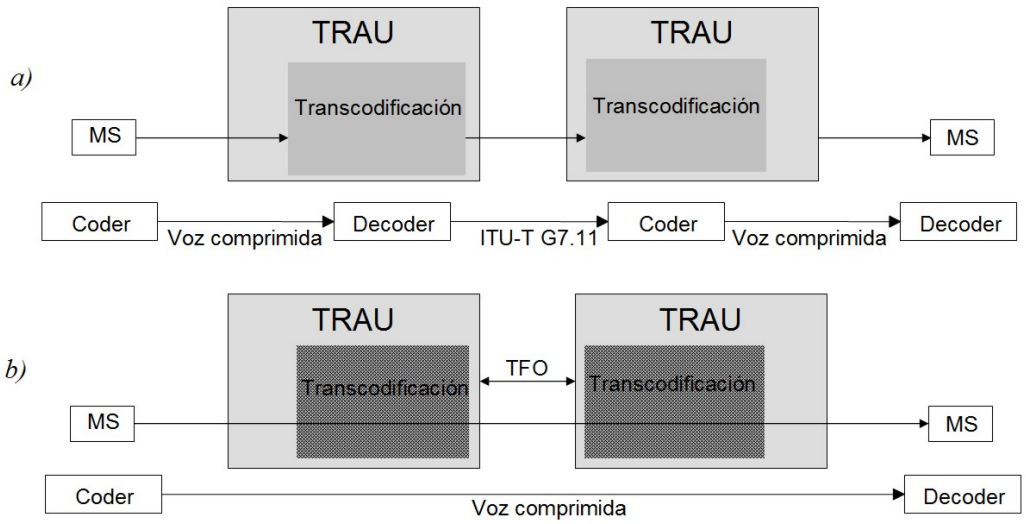


Figura 4.10: Configuración de llamada móvil a móvil: a) Operación típica con codecs en tandem, b) Operación libre de tandem (TFO).

- Mediante una modificación mayor, la transparametrización podría constituir una parte integral de la TRAU. Es decir, al igual que ésta incluye transcodificadores para los distintos codecs hacia PCM (para transmisión de voz), pueden añadirse transparametrizadores hacia el estándar DSR (para reconocimiento del habla). Sin provocar cambios en el hardware de usuario, la voz codificada podría ser directamente convertida en características de voz compatibles con DSR, siendo transmitidas por medio de los mismos mecanismos que los que se proponen para la arquitectura DSR (figura 4.11b).

Como es evidente, en estos casos sería necesario realizar cambios hardware en la red, concretamente en la TRAU. Sin embargo, esta aproximación sólo implicaría cambios en hardware centralizado. En las redes más antiguas la TRAU se encuentra localizada en la BTS, mientras que en las redes más modernas la TRAU suele localizarse en la BTC o, incluso, en el MSC. Esto se debe a las ventajas que proporciona retrasar la transcodificación de la voz en la red. Al requerir menos ancho de banda que la voz a 64 kbps, se prefiere transmitir los parámetros del codec por los enlaces que interconectan BTS, BTC y MSC (*A* y *A bis*), optimizando su uso.

Sin embargo, todos estos cambios pueden evitarse si la red GSM subyacente da soporte al protocolo *TFO* (*Tandem Free Operation*). En la configuración de llamada de móvil a móvil descrita anteriormente, los dos codecs de voz están en lo que se conoce como *"Tandem Operation"*. El principal inconveniente de la configuración en tandem es

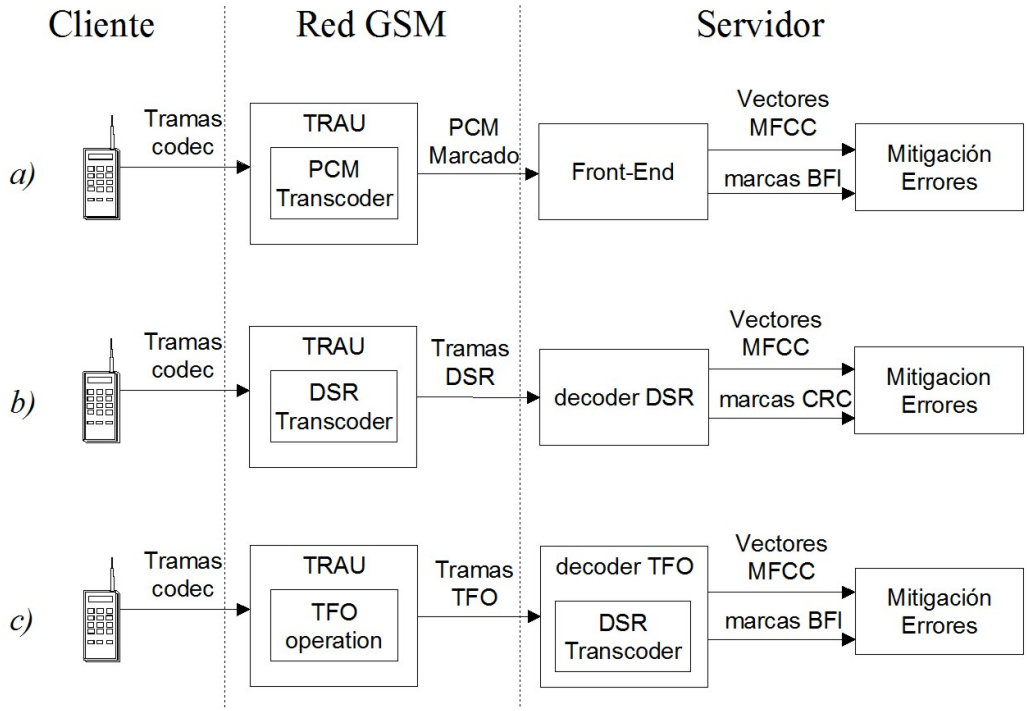


Figura 4.11: Soluciones alternativas a DSR: a) marcado PCM, b) protocolo TFO, c) trasparametrización DSR en la TRAU. En cada una de ellas se evita la modificación del dispositivo móvil.

la degradación de la calidad que sufre la voz debido al doble proceso de transcodificación. Esta degradación es más notable conforme los codecs de voz operan a menor tasa de bits, como es el caso de AMR. Para evitar esta degradación, ETSI propone el uso del estándar ETSI TS 128 062 [133] o protocolo TFO. Mediante este protocolo, las unidades de transcodificación intercambian tramas TFO a lo largo de la red fija (figura 4.10b) mediante el robo del bit menos significativo de cada muestra de voz. Las tramas TFO contienen la voz comprimida (bitstream del codificador) además de señalización en banda (marcas BFI entre otras).

Los procedimientos empleados para el establecimiento de TFO son considerados independientes del sistema y pueden ser extendidos a otras configuraciones de llamada que implique, como indica el propio estándar, otros sistemas como teléfonos RDSI, servidores de voz, multimedia sobre IP u otros sistemas inalámbricos. Por tanto, una tercera solución consistiría en conectar a la red un dispositivo compatible con TFO que no sólo accediera a la marca BFI, sino que también decodificara los parámetros del codec. Este dispositivo se encargaría de la transparametrización propuesta, obteniendo vectores de características compatibles con el servidor de reconocimiento (figura 4.11c).

4.7. Resumen de resultados y conclusiones

Este capítulo examina el concepto de transparametrización de voz codificada y su utilidad de cara a la supresión parcial del ruido de memoria. El análisis de la voz codificada con EFR, descrito en el capítulo previo, revelaba dos fuentes relevantes de degradación cuando se emplea un canal de voz GSM propenso a errores: el ruido de sustitución y apagado, y el ruido de memoria. El ruido de memoria es característico en los codificadores CELP, que emplean un filtro de largo retardo (LTP) al representar la señal de excitación, y consiste en una propagación hacia delante del efecto de los errores de trama. Tanto el ruido de sustitución y apagado como el ruido de memoria pueden ser tratados directamente sobre la señal decodificada, como se muestra en la sección 4.3. En este sentido, la técnica FCDCN, originalmente propuesta en un contexto de ruido acústico, ha resultado, con los cambios apropiados, muy efectiva para la reducción del ruido de memoria, a la vez que fácilmente extensible a otros codificadores CELP. Sin embargo, la transparametrización de voz codificada permite obtener mejores resultados.

En este capítulo se propone y evalúa una transparametrización EFR/DSR. Gracias a esta transparametrización, el ruido de memoria queda confinado en las características energéticas ($MFCC0$ y $logE$), permitiendo un incremento significativo en la precisión del reconocedor. Este ruido de memoria residual puede tratarse mediante la técnica A-FCDCN como se hiciera sobre la voz decodificada. Como resultado, el transparametrizador de EFR a DSR, junto con las técnicas de mitigación descritas, proporciona un rendimiento en el reconocimiento similar o incluso superior a aquel obtenido por el estándar de ETSI para DSR. Adicionalmente, la transparametrización descrita es compatible con los reconocedores basados en este estándar, no siendo necesario el re-entrenamiento de los mismos.

Una extensión natural de la transparametrización EFR/DSR consiste en su aplicación al codec AMR. Este codec está compuesto de varios modos de operación con diferentes tasas de bits, de forma que conforme se degrada el canal, se elige un modo de operación con menor tasa, permitiendo una codificación de canal más robusta. Cada uno de estos modos de operación se basa en los mismos conceptos que EFR, siendo este último, de hecho, el modo de mayor calidad. Aunque AMR reduce la presencia de tramas incorrectas no es inmune a ellas, estando afectado, aunque en menor medida, por los ruidos descritos para EFR. Dado que el uso de la transparametrización resultaría beneficioso para combatir estos problemas, en la sección 4.5.1 se describe una extensión de ésta para englobar al codec AMR. Por otro lado, debe considerarse la degradación introducida por el proceso

de codificación de AMR sobre el reconocimiento. Mientras que en EFR esta degradación puede ser despreciable, no es así en los modos de menor tasa de bits de AMR. Es por ello que en este capítulo también se evalúa el efecto de la transparametrización sobre este tipo de ruido, concluyéndose que la transparametrización puede resultar también positiva frente a este tipo de ruido.

Finalmente, en este capítulo también se proponen distintas alternativas para lograr el acceso directo al bitstream, necesario para aplicar los métodos propuestos. En el caso de que una red con soporte TFO esté disponible, no es necesario hacer ninguna modificación en el dispositivo del usuario ni en la red, dado que el bitstream puede ser obtenido mediante un dispositivo compatible con el protocolo TFO. Si la red no ofrece este soporte, su inclusión podría resultar muy interesante para el operador móvil ya que no sólo se podría ofrecer un reconocimiento de calidad comparable a DSR (pero sin efectuar modificaciones en el terminal de usuario), sino también una mejor calidad de voz en llamadas entre móviles. Si esto no fuera posible, el transcoder podría implementarse en la TRAU como otra función más de transcodificación. Ambas soluciones muestran dos ventajas importantes: modificaciones centralizadas y escalabilidad. Mientras que el terminal de usuario permanece sin modificaciones, los cambios son realizados de forma localizada en la red, esto es, no es necesario una actualización global, sino tan sólo de aquellas partes terminales donde, debido a su localización física (edificios públicos, aeropuertos, etc.) el reconocimiento de voz fuese un servicio a ofrecer.

Las soluciones presentadas en este capítulo no compiten con el uso de la arquitectura DSR, puesto que pueden coexistir con esta solución. Justamente al contrario, facilitan la transición hacia esta arquitectura, ya que presentan una alternativa robusta en el paradigma del reconocimiento remoto de la voz, permitiendo un mayor número de opciones entre las que elegir de cara a cada situación.

Capítulo 5

MITIGACIÓN DE PÉRDIDAS EN EL RECEPTOR

5.1. Introducción

Como indicamos en el capítulo 3, el concepto de canal con pérdidas permite una visión unificada de las arquitecturas NSR y DSR, en donde los errores del canal derivan en la pérdida completa de tramas con información de voz. La única diferencia entre ambas arquitecturas radica en la forma en que la voz es codificada. Si la voz se representaba de forma predictiva, como ocurre con los codificadores CELP (empleados extensivamente en GSM), además de la consecuente pérdida, se originaba un ruido de memoria sobre las subsiguientes tramas recibidas. Para combatir dicho ruido, en el capítulo anterior proponíamos técnicas específicamente adaptadas, así como la transparametrización de los parámetros de voz. Resta entonces compensar los efectos originados exclusivamente por la pérdida de información propiamente dicha.

La pérdida de tramas trae consigo la pérdida de los vectores de características correspondientes a dichas tramas. Esto resulta evidente en el caso en que se use un extractor de características y se transmitan tramas con parámetros de reconocimiento (arquitectura DSR). Sin embargo, también es extensible al caso en el que transmitimos tramas con parámetros de un codec. Cuando se usa un codificador de voz, el descarte o pérdida de una trama da lugar a una señal artificial fruto de un algoritmo generalmente limitado por la latencia. Esta señal carece de información relevante, por lo que los vectores de características correspondientes pueden ser igualmente considerados perdidos, como ya hiciéramos en la sección 4.3.1.

Como se mostró en las secciones 3.6 y 3.7, la pérdida de vectores de características (sea cual fuere su origen) afecta negativamente al reconocimiento. Por tanto, se hace necesario el uso de técnicas que prevengan, corrijan y compensen sus efectos. En general, nos referiremos a ellas como técnicas de recuperación. Siguiendo una clasificación similar a la propuesta por Perkins et al. [134] en el contexto de la transmisión de voz, estas técnicas pueden agruparse en :

- Técnicas basadas en el emisor. Estas técnicas se caracterizan porque requieren la participación del emisor, pudiendo ser clasificadas en activas o pasivas. Las primeras se refieren a los esquemas de retransmisión, mientras que las segundas incluyen los códigos de corrección hacia delante y el entrelazado. Estas técnicas tratan de evitar las pérdidas, o al menos, que el impacto de éstas sobre el rendimiento del sistema sea menos degradante.

- Técnicas de mitigación. Se aplican en el receptor y no requieren de la participación del emisor. A diferencia de las anteriores, que pueden considerarse como preventivas, el objetivo de estas técnicas consiste en mitigar las pérdidas una vez se han producido. Por ello, resultan especialmente útiles cuando las técnicas basadas en el emisor resultan insuficientes o cuando el emisor no puede participar en la recuperación.

- Tratamiento de pérdidas en el reconocedor. El potente modelo estadístico presente en el reconocedor no sólo puede aprovecharse para el reconocimiento sino, además, para tratar las pérdidas. Estas técnicas consisten en modificaciones practicadas sobre el reconocedor para que este pueda trabajar con datos perdidos o poco fiables. Evidentemente, son especificas para el reconocimiento de voz.

En este capítulo nos centraremos sobre el segundo grupo de técnicas, las técnicas de mitigación, explorando las restantes en el capítulo siguiente. Estas técnicas resultan muy atractivas tanto para la arquitectura NSR como DSR. En la primera porque, al no requerirse su colaboración, no es necesario hacer modificaciones sobre el emisor, conservándose la principal ventaja de esta arquitectura. En la segunda, en cambio, su utilidad radica en la idea subyacente de disponer de clientes de baja capacidad junto a potentes servidores que puedan realizar tareas de reconocimiento complejas. De esta forma, resulta lógico beneficiarse de la alta capacidad de cómputo de dicho servidores para introducir técnicas de mitigación cuyo coste sea razonablemente inferior al requerido por el propio reconocedor.

5.2. Aproximaciones a la mitigación de pérdidas

Existen diferentes aproximaciones o tipos de técnicas de mitigación aplicables a la pérdida de información. Exceptuando las técnicas más simples, como el ensamblado de tramas, las técnicas de mitigación aprovechan la alta redundancia de la voz para mitigar los efectos de las pérdidas. Generalmente, estas redundancias se presentan en forma de correlaciones entre las distintas características de los vectores de reconocimiento, siendo con frecuencia la correlación temporal la dominante.

En esta sección se hace una breve clasificación de estas técnicas en base a su complejidad. En este sentido, veremos como las técnicas de mitigación más complejas, aquellas basadas en un modelo de voz, terminan por agrupar a todas las anteriores.

5.2.1. Ensamblado de tramas

El ensamblado o unión de tramas es una de las técnicas más sencillas para tratar las pérdidas. Consiste en unir los fragmentos recibidos antes y después de una ráfaga, de forma que no haya ningún hueco. Sin embargo, este esquema presenta un importante problema, ya que la temporización del flujo se perturba. Por esta razón, suele ser una técnica rechazada tanto en transmisión como en reconocimiento de la voz. En el primer caso, además de obtener unos resultados pobres [135], introduce una alteración en la memoria intermedia previa a la reproducción que, dependiendo del codificador de voz usado, puede degradar las prestaciones de todo el sistema. En el caso del reconocimiento del habla, el ensamblado de tramas da lugar a una pérdida de información temporal. Al no respetarse la temporización, se perturba la secuencia natural de transiciones entre estados del HMM, de forma que las características recibidas terminan siendo analizadas empleando un estado inapropiado del modelo. Este problema se agudiza cuando se producen pérdidas en ráfaga, ya que se impide la evolución natural entre estados del modelo correspondiente a un largo periodo de tiempo, forzando un salto brusco entre el estado anterior y posterior a la ráfaga que afecta seriamente la precisión del reconocimiento [136].

5.2.2. Técnicas de Inserción

Las técnicas basadas en inserción reconstruyen las tramas perdidas insertando un relleno sencillo en su lugar. De esta forma, a diferencia de las anteriores, no se perturba la información temporal. Se pueden distinguir tres tipos según el relleno que se utilice: silencio, ruido o repetición de las tramas adyacentes.

En la sustitución por silencio se rellena el hueco que se ha perdido con tramas de silencio, manteniendo así la relación temporal entre todos los paquetes. Sin embargo es sólo efectivo para segmentos cortos y bajas tasas de pérdidas. A pesar de esto se utiliza extensamente en transmisión de voz debido a su facilidad de implementación [137].

La sustitución por ruido rellena el hueco perdido con tramas de ruido. Esta sustitución funciona mejor que la anterior de cara a la transmisión de voz [138]. Diversos estudios, como por ejemplo el de Warren [139], muestran que el cerebro humano subconscientemente recupera mejor segmentos con ruido que con silencio. De cara al reconocimiento, es preferible realizar las inserciones con el vector promedio. De esta forma se evitan los posibles errores de inserción que podrían provocar los silencios al situarse en medio de una palabra.

Por último, la sustitución por repetición reemplaza las tramas perdidas con copias de las inmediatamente recibidas antes o después de la ráfaga. En la repetición hacia adelante se emplea como sustituto la última trama recibida antes de la ráfaga, mientras que en la repetición hacia atrás se emplea la primera trama tras la ráfaga. Debido a que no introduce ningún retardo, se prefiere la primera frente a esta última. En reconocimiento de voz suele aplicarse una mezcla de las dos anteriores conocida como repetición de la trama más próxima, ya que la latencia no es tan crítica en reconocimiento como en transmisión de voz. Ésta consiste en la repetición de la última trama recibida antes de la ráfaga durante la primera mitad del segmento perdido, mientras que la segunda mitad se recupera mediante una copia de la primera trama recibida después de la ráfaga. Esta técnica es la usada por el estándar de la ETSI para DSR [74] en redes GSM. Posteriormente, Quercia et al. [140] evaluaron la precisión de DSR sobre redes IP empleando la misma base de datos y front-end del grupo Aurora, obteniendo buenos resultados al utilizar esta técnica para la mitigación de paquetes perdidos. Finalmente, el uso de esta técnica se ha extendido al estándar de definición de carga para IP [79].

5.2.3. Técnicas de Interpolación

Las técnicas de interpolación constituyen una aproximación más avanzada, aunque sencilla, para el tratamiento de la pérdida de información. Estas técnicas obtienen los reemplazos necesarios a partir de características recibidas previa y posteriormente, y una función de interpolación.

De forma general, dada una ráfaga de vectores perdidos de longitud T, un vector interpolado para el instante temporal t se obtiene como,

$$\hat{\mathbf{x}}_t = F\left(t; \mathbf{x}_{-\mathcal{M}+1}, \mathbf{x}_{-\mathcal{M}+2}, \ldots, \mathbf{x}_0, \mathbf{x}_{T+1}, \mathbf{x}_{T+2}, \mathbf{x}_{T+\mathcal{N}}\right) \quad (1 \leq t \leq T) \tag{5.1}$$

donde F es la función de interpolación, y $\mathcal{M}$ y $\mathcal{N}$ son, respectivamente, el número de vectores correctamente recibidos antes y después de la ráfaga, en base a los cuales se realizará la interpolación. Debe notarse que $\mathcal{M}$ y $\mathcal{N}$ representan valores máximos ya que, debido a la existencia de ráfagas previas y posteriores, cabe la posibilidad de que dichos vectores no estén disponibles.

Las técnicas de repetición pueden considerarse un caso específico de las técnicas de interpolación. La repetición hacia delante se obtiene considerando $\mathcal{M} = 1$ y $\mathcal{N} = 0$, tal que $\hat{\mathbf{x}}_t = \mathbf{x}_0$, y la estimación hacia atrás con $\mathcal{M} = 0$, $\mathcal{N} = 1$, $\hat{\mathbf{x}}_t = \mathbf{x}_{T+1}$. De forma análoga, la repetición del vector más cercano vendría dada por,

$$\hat{\mathbf{x}}_t = \begin{cases} \hat{\mathbf{x}}_0 & \text{si } t \leq \lceil T/2 \rceil \\ \hat{\mathbf{x}}_{T+1} & \text{si } t > \lceil T/2 \rceil \end{cases} \tag{5.2}$$

en donde $\mathcal{M} = \mathcal{N} = 1$.

Comúnmente, se escoge un función polinomial como F, siendo $\mathcal{M} + \mathcal{N} - 1$ el grado mínimo requerido para el polinomio. Considerando únicamente el vector previo y posterior a la ráfaga se obtiene una interpolación lineal, dada por,

$$\hat{\mathbf{x}}_t = \mathbf{x}_0 + \frac{t}{T+1}(\mathbf{x}_{T+1} - \mathbf{x}_0) \tag{5.3}$$

La interpolación lineal [141] ha sido usada previamente en reconocimiento remoto bajo diferentes contextos. Sin embargo, diferentes autores [142, 143] han verificado que, en comparación con la mitigación por repetición, la interpolación lineal obtiene peor rendimiento. Este comportamiento ocurre pese a que la interpolación lineal ofrece una transición más suave desde $\mathbf{x}_0$ a $\mathbf{x}_{T+1}$ y que, como puede probarse, proporciona un menor error cuadrático medio que la repetición del más próximo. Este problema puede entenderse mejor en términos del alineamiento llevado a cabo por el algoritmo de Viterbi empleado en el reconocimiento. Tan et al. [143] probaron que la interpolación lineal causa frecuentes transiciones fuera del estado actual, dando lugar a una duración media por estado menor a la obtenida con las características originales. Por el contrario, la reconstrucción por repetición del vector más cercano aproxima mejor estas duraciones.

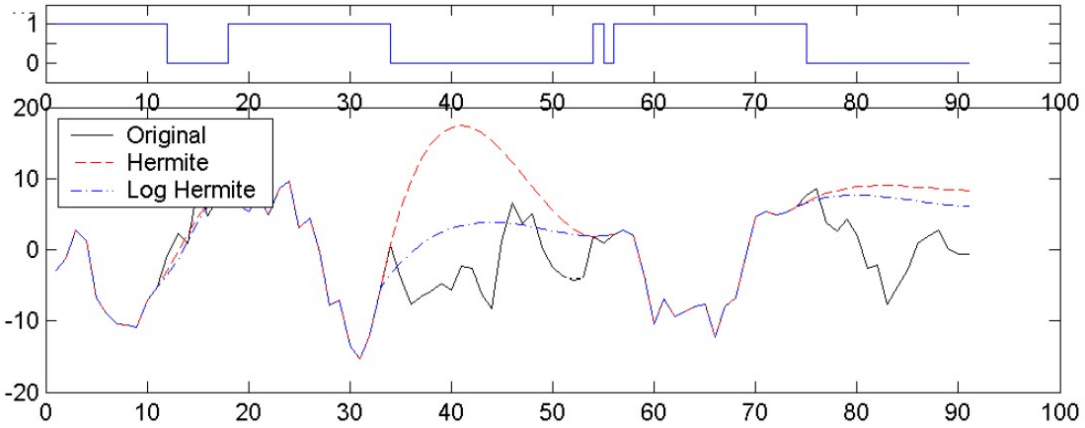


Figura 5.1: Ejemplo de reconstrucción comparando la interpolación cúbica de Hermite con y sin dominio logarítmico.

James et al. proponen en [109] una interpolación basada en polinomios de orden superior en donde se consideren las dos últimas y primeras tramas recibidas previa y posteriormente a la ráfaga, respectivamente ($\mathcal{M} = \mathcal{N} = 2$). Para ello, se emplean polinomios cúbicos de Hermite ($\mathcal{M}+\mathcal{N}-1 = 3$), de forma que la interpolación quede expresada como,

$$\hat{\mathbf{x}}_t = \mathbf{a}_0 + \mathbf{a}_1\bar{t} + \mathbf{a}_2\bar{t}^2 + \mathbf{a}_3\bar{t}^3 \tag{5.4}$$

donde $\bar{t} = t/(T+1)$ y los coeficientes $\mathbf{a}_i$ $(i = 0, \ldots, 3)$ son los coeficientes multivariados del polinomio. Esta interpolación asegura una trayectoria suave al forzar no sólo la continuidad de los valores interpolados sino, además, su primera derivada al principio y al final de la ráfaga. Despues de obtener los coeficientes, la ecuación (5.4) puede expresarse como,

$$\begin{aligned}\hat{\mathbf{x}}_t = \quad & \mathbf{x}_0(\bar{t} - 3\bar{t}^2 + 2\bar{t}^3) + \mathbf{x}_{T+1}(3\bar{t}^2 + 2\bar{t}^3) + \\ & \mathbf{x}'_0(\bar{t} - 2\bar{t}^2 + \bar{t}^3) + \mathbf{x}'_{T+1}(\bar{t}^2 + \bar{t}^3)\end{aligned} \tag{5.5}$$

en donde las derivadas pueden ser aproximadas por $\mathbf{x}'_0 = T(\mathbf{x}_0 - \mathbf{x}_{-1})$ y $\mathbf{x}'_{T+1} = T(\mathbf{x}_{T+2} - \mathbf{x}_{T+1})$. Adicionalmente, James propone que la interpolación se realice en el dominio logarítmico. Esto se debe a que en la práctica, bajo fluctuaciones rápidas de las características, la interpolación puede dar reemplazos incorrectos al sobrestimar las velocidades (figura 5.1).

Finalmente, la extrapolación puede considerarse un caso particular de la interpolación en donde $\mathcal{N} = 0$. Aunque la extrapolación ofrece aproximaciones de los vectores perdidos

más pobres, puede resultar útil en sistemas fuertemente limitados por el retardo.

5.2.4. Técnicas de Estimación

Al igual que las técnicas de interpolación, el objetivo de las técnicas de estimación consiste en ofrecer una sustitución para aquellos parámetros que no están disponibles debido a errores del canal. La diferencia con las técnicas anteriores radica en que las técnicas de estimación hacen uso, de forma explícita, de un modelo estadístico de la voz. De forma implícita, las técnicas de interpolación también hacen uso de un modelo, al aplicarse una función paramétrica para obtener la sustitución, pero a diferencia de la estimación, este modelo no ha sido entrenado con voz, sino que su selección está basada en consideraciones como la forma de la trayectoria de las características de voz.

El entorno probabilístico provisto por los modelos de voz permite la obtención de estimaciones a partir de los datos disponibles. De forma amplia, se pueden considerar como datos disponibles todos los datos recibidos. Sin embargo, por cuestiones de latencia, los datos considerados en la estimación suelen limitarse a unos pocos datos recibidos previa y posteriormente a la ráfaga de error, y a los datos recibidos durante la propia ráfaga. Estos últimos resultan útiles en aquellos canales que pueden ofrecer datos parcialmente correctos durante una ráfaga, como por ejemplo, los inalámbricos. En el contexto de canales con pérdidas, que se caracterizan precisamente por la pérdida de la información durante las ráfagas, generalmente no se consideran estos datos durante la estimación. Esto conduce a ciertas simplificaciones y optimizaciones. Sin embargo, conviene tener presente esta posibilidad ya que, como veremos en el siguiente capítulo, mediante técnicas aplicadas en el emisor es posible recibir cierta información durante la propia ráfaga.

En lo que resta de este capítulo presentaremos varios tipos de técnicas de mitigación basadas en estimación de vectores perdidos. Éstas se derivan de dos métodos de estimación bien conocidos: la estimación basada en el mínimo error cuadrático medio (*MMSE–Minimum Mean Square Error*) y la estimación basada en máximo a posteriori (*MAP–Maximum a Posteriori*).

5.3. Estimación basada en Mínimo Error Cuadrático Medio

5.3.1. Fundamentos de la estimación MMSE

La estimación MMSE se basa en el cálculo del valor esperado del vector correspondiente al instante de tiempo t, dado por,

$$\hat{\mathbf{x}}_t = E\left[\mathbf{x}_t|\text{datos disponibles, modelo}\right] \tag{5.6}$$

Considerando una cuantización previa de los vectores de características, tal que $\mathbf{x}_t$ pertenezca a un conjunto finito de símbolos $\left\{\mathbf{x}^{(i)}; i = 0, \ldots, N-1\right\}$, la estimación MMSE del vector en el instante de tiempo t puede expresarse como,

$$\hat{\mathbf{x}}_t = \sum_{i=0}^{N-1} \mathbf{x}^{(i)} P\left(\mathbf{x}_t^{(i)}|\Lambda\right) \tag{5.7}$$

en donde, por simplicidad, se emplea la notación $\mathbf{x}_t^{(i)}$ para expresar $\mathbf{x}_t = \mathbf{x}^{(i)}$. Las probabilidades $P(\mathbf{x}_t^{(i)}|\Lambda)$ son proporcionadas por el modelo, siendo Λ el conjunto de datos disponibles. Supuesta una ráfaga de vectores erróneos desde $t = 1$ hasta $t = T$, Λ puede expresarse como (X^-, Y_1^T, X^+), donde $X^- = (\mathbf{x}_{-\mathcal{M}+1}, \ldots, \mathbf{x}_0)$ representa los $\mathcal{M}$ vectores recibidos antes de la ráfaga, $X^+ = (\mathbf{x}_{T+1}, \ldots, \mathbf{x}_{T+\mathcal{N}})$ los $\mathcal{N}$ vectores recibidos tras la ráfaga, y $Y_1^T = (\mathbf{y}_1, \ldots, \mathbf{y}_T)$, los T vectores recibidos durante la ráfaga (supuestos erróneos). Generalmente, en los canales con pérdidas no se recibe ningún vector durante la ráfaga por lo que Λ puede reducirse a (X^-, X^+). Sin embargo, por el momento mantendremos la formulación más general, simplificándola más adelante. Esto se debe a que en el capítulo siguiente se propondrán ciertas técnicas capaces de introducir vectores durante una ráfaga.

La estimación MMSE permite combinar de forma eficiente la información a priori disponible acerca de la evolución de la fuente y la información recibida del canal durante una ráfaga. La información a priori de la fuente se modela por medio de un proceso discreto de Markov, al que denominaremos *modelo de voz*, gracias al cual podemos conocer la probabilidad de que un determinado símbolo $x^{(i)}$ sea transmitido. Por su parte, la información del canal puede modelarse mediante un *modelo de canal*, gracias al cual será posible conocer la probabilidad de que un símbolo transmitido $x^{(i)}$ sea recibido como $\mathbf{y}_t$, esto es, $P(\mathbf{y}_t|\mathbf{x}^{(i)})$.

El proceso discreto de Markov que modela la fuente puede establecerse de distintos ordenes, siendo el más sencillo de todos el de orden 0. En este caso, se considera $\Lambda = \mathbf{y}_t$ ($\mathcal{M} = \mathcal{N} = 0$), de forma que la expresión (5.7) queda reducida a,

$$\hat{\mathbf{x}}_t = \sum_{i=0}^{N-1} \mathbf{x}^{(i)} P\left(\mathbf{x}^{(i)}|\mathbf{y}_t\right) \tag{5.8}$$

$$P\left(\mathbf{x}^{(i)}|\mathbf{y}_t\right) = \frac{P\left(\mathbf{y}_t|\mathbf{x}^{(i)}\right) P_i}{P(\mathbf{y}_t)} \tag{5.9}$$

en donde el modelo de voz queda descrito completamente por las probabilidades a priori de los símbolos P_i ($P_i = P(\mathbf{x}^{(i)})$). Sin embargo, esta aproximación carece de interés para aquellos canales en los que no se recibe nada durante las ráfagas, ya que conduce a la sustitución por la media global de los vectores. Si no se recibe ningún vector erróneo $\mathbf{y}_t$, todas las probabilidades $P(\mathbf{y}_t|\mathbf{x}^{(i)})$ pueden considerarse idénticas y la ecuación (5.9) queda regida sólo por las probabilidades a priori, reduciéndose al cálculo de la media.

Para mejorar la estimación puede emplearse un modelo de voz de primer orden en donde no sólo se tengan en cuenta las probabilidades a priori, sino también las probabilidades de transición entre símbolos. Conceptualmente, esto es equivalente a emplear un modelo HMM para la estimación MMSE. En este caso, cada símbolo se asocia a un estado, de forma que las probabilidades de transición entre estados vienen dadas por las probabilidades transición entre símbolos $a_{ij} = P(\mathbf{x}_t^{(j)}|\mathbf{x}_{t-1}^{(i)})$, y las probabilidades de observación por $b_i(\mathbf{y}_t) = P(\mathbf{y}_t|\mathbf{x}^{(i)})$.

Considerando que $(\mathbf{x}_0, \mathbf{y}_1, \ldots, \mathbf{y}_T, \mathbf{x}_{T+1})$ es la secuencia de vectores recibida, en donde $\mathbf{x}_0$ es el último vector correcto antes de una ráfaga de error de longitud T, y $\mathbf{x}_{T+1}$ el primero posterior a ésta ($\mathcal{M} = \mathcal{N} = 1$), la estimación MMSE para el vector en el instante de tiempo t viene dada por,

$$\hat{\mathbf{x}}_t = E[\mathbf{x}_t|\mathbf{x}_0, \mathbf{y}_1, \ldots, \mathbf{y}_T, \mathbf{x}_{T+1}] = \sum_{i=0}^{N-1} \mathbf{x}^{(i)} \gamma_t(i) \quad (1 \leq t \leq T) \tag{5.10}$$

donde la probabilidad condicional $\gamma_t(i)$ se define como,

$$\gamma_t(i) = P(\mathbf{x}_t^{(i)}|\mathbf{x}_0, \mathbf{y}_1, \ldots, \mathbf{y}_T, \mathbf{x}_{T+1}) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=0}^{N-1} \alpha_t(j)\beta_t(j)} \tag{5.11}$$

donde $\alpha_t(i)$ y $\beta_t(i)$ son, respectivamente, la probabilidad condicional hacia delante y la

probabilidad condicional hacia atrás, definidas como,

$$\alpha_t(i) = P(\mathbf{x}_t^{(i)}|\mathbf{x}_0, \mathbf{y}_1, \ldots, \mathbf{y}_t) \tag{5.12}$$

$$\beta_t(i) = P(\mathbf{y}_{t+1}, \ldots, \mathbf{y}_T, \mathbf{x}_{T+1}|\mathbf{x}_t^{(i)}) \tag{5.13}$$

Estas probabilidades condicionales pueden calcularse a partir de las probabilidades de transición a_{ij} mediante un procedimiento recursivo conocido como *algoritmo adelante-atrás* [124]. Para obtener $\alpha_t(i)$, la recursión se inicializa en $t = 0$ como,

$$\alpha_0(i) = P_i b_i(\mathbf{x}_0)/K_0 \qquad (i = 0, 1, \ldots, N-1) \tag{5.14}$$

$$b_j(\mathbf{x}_0) = \begin{cases} 0 & \mathbf{x}^{(i)} \neq \mathbf{x}_0 \\ 1 & \mathbf{x}^{(i)} = \mathbf{x}_0 \end{cases} \tag{5.15}$$

obteniéndose para el resto de instantes de tiempo ($t > 0$) por,

$$\alpha_t(i) = \left[\sum_{j=0}^{N-1} \alpha_{t-1}(j) a_{ji}\right] b_i(\mathbf{y}_t)/K_t \tag{5.16}$$

donde K_t ($t = 0, \ldots, T$) es el factor de normalización en el instante t.

Para calcular $\beta_t(i)$, la recursión se inicializa en $t = T + 1$ como,

$$\beta_{T+1}(i) = 1 \qquad (i = 0, 1, \ldots, N-1) \tag{5.17}$$

$$b_j(\mathbf{y}_{T+1}) = b_j(\mathbf{x}_{T+1}) = \begin{cases} 0 & \mathbf{x}^{(i)} \neq \mathbf{x}_{T+1} \\ 1 & \mathbf{x}^{(i)} = \mathbf{x}_{T+1} \end{cases} \tag{5.18}$$

obteniéndose para el resto de instantes de tiempo ($t \leq T$) por,

$$\beta_t(i) = \sum_{j=0}^{N-1} a_{ij} b_j(\mathbf{y}_{t+1}) \beta_{t+1}(j) \tag{5.19}$$

Las probabilidades a priori y de transición pueden obtenerse a partir de la base de datos de entrenamiento. Las probabilidades de observación, en cambio, deberán obtenerse teniendo en cuenta el esquema de transmisión empleado y las propiedades del canal considerado. La técnica resultante, denominada forward-backward MMSE (*FB–MMSE*) [142, 144, 145], obtiene estimaciones precisas de los vectores erróneos incluso en condiciones de canal muy adversas, como se demuestra en el trabajo de Peinado et al. [142], en

donde la técnica FB–MMSE se aplica al reconocimiento de voz mediante una arquitectura DSR operando sobre GSM.

5.3.2. Reconstrucción basada en modelo de primer orden

Por la propia naturaleza de los canales con pérdidas, es de esperar que durante la aparición de una ráfaga, no se disponga de ningún vector de características, ni siquiera erróneo. Al no disponer de ningún vector observado, la aproximación más coherente consiste en asumir que la probabilidad de observación $b_i(\mathbf{y}_t)$ es uniforme y equiprobable para todos los símbolos. De esta forma, las ecuaciones (5.16) y (5.19) pueden formularse como,

$$\alpha_t(i) = \left[\sum_{j=0}^{N-1} \alpha_{t-1}(j) a_{ji}\right] \tag{5.20}$$

$$\beta_t(i) = \sum_{j=0}^{2^M-1} a_{ij} \beta_{t+1}(j) \tag{5.21}$$

en donde se ha considerado $b_i(\mathbf{y}_t) = 1$ ya que, gracias a la normalización aplicada, el valor en concreto resulta irrelevante. Las probabilidades condicionales hacia delante y hacia atrás quedan así reducidas a,

$$\alpha_t(i) = P(\mathbf{x}_t^{(i)} | \mathbf{x}_0) \tag{5.22}$$

$$\beta_t(i) = P(\mathbf{x}_{T+1} | \mathbf{x}_t^{(i)}) \tag{5.23}$$

las cuales podrían obtenerse directamente a partir de la base de datos de entrenamiento, como ya se hiciera con las probabilidades a_{ij}. Como puede observarse, en este caso la técnica FB–MMSE queda únicamente guiada por la información del modelo de voz (probabilidades a priori y de transición). Conceptualmente, esto es equivalente a no considerar de forma explicita un modelo de canal.

Sin embargo, la aplicación de la técnica FB–MMSE a canales con pérdidas no mejora los resultados obtenidos mediante la sencilla repetición del vector más próximo (véase sección 5.3.6). Esto se debe a la degradación que sufren las probabilidades en los extremos. Conforme se incrementa t la probabilidad hacia adelante tiende a hacerse equiprobable para todos los símbolos. Lo mismo ocurre con la probabilidad hacia atrás conforme t decrece. Suponiendo una ráfaga suficientemente larga, la información que puede ofrecer en

un extremo el vector recibido en el extremo contrario es prácticamente nula. Sin embargo, ambas probabilidades se ponderan igual en la ecuación (5.11), degradándose mutuamente.

A fin de evitar esta mutua degradación, puede recurrirse a una estrategia F+B–MMSE. En ella la ráfaga se divide en dos partes de igual tamaño. Sobre la primera parte se aplica una estimación hacia adelante basada en el último vector recibido antes de la ráfaga, mientras que en la segunda se aplica una estimación hacia atrás basada en el primer vector recibido tras ella. Supuesta una ráfaga de longitud $2B$, la estimación F+B–MMSE para un canal con pérdidas vendría dada por,

$$\hat{\mathbf{x}}_t = \sum_{i=0}^{N-1} \mathbf{x}^{(i)} P(\mathbf{x}_t^{(i)}|\mathbf{x}_0) \qquad (1 \leq t < B) \tag{5.24}$$

$$\hat{\mathbf{x}}_t = \sum_{i=0}^{N-1} \mathbf{x}^{(i)} P(\mathbf{x}_t^{(i)}|\mathbf{x}_{T+1}) \qquad (B \leq t < 2B) \tag{5.25}$$

Como puede observarse, estas estimaciones no emplean información instantánea, dependiendo únicamente del símbolo recibido antes y después de la ráfaga. Supuesto que el símbolo recibido fuera j en $t = 0$ y $t = T$, entonces,

$$\hat{\mathbf{x}}_t(j) = \sum_{i=0}^{N-1} \mathbf{x}^{(i)} P(\mathbf{x}_t^{(i)}|\mathbf{x}_0^{(j)}) \qquad (1 \leq t < B) \tag{5.26}$$

$$\hat{\mathbf{x}}_t(j) = \sum_{i=0}^{N-1} \mathbf{x}^{(i)} P(\mathbf{x}_t^{(i)}|\mathbf{x}_{T+1}^{(j)}) \qquad (B \leq t < 2B) \tag{5.27}$$

En general, para cada símbolo j recibido en un cierto instante t (por simplicidad asumiremos $t = 0$), podemos calcular una estimación, $\hat{\mathbf{x}}_l(j)$, del vector correspondiente a l instantes de tiempo (posteriores o anteriores), como,

$$\hat{\mathbf{x}}_l(j) = \sum_{i=0}^{N-1} \mathbf{x}^{(i)} P(\mathbf{x}_l^{(i)}|\mathbf{x}_0^{(j)}) \quad (-L \leq l < 0), (0 < l \leq L) \tag{5.28}$$

donde L es la distancia máxima considerada. Entonces, es posible precalcular una *secuencia de estimaciones hacia delante*, $E_F(j)$, y *secuencia de estimaciones hacia atrás*, $E_B(j)$, dado un símbolo j, definidas como,

$$E_F(j) = (\hat{\mathbf{x}}_1(j), \hat{\mathbf{x}}_2(j), ..., \hat{\mathbf{x}}_L(j)) \tag{5.29}$$

$$E_B(j) = (\hat{\mathbf{x}}_{-L}(j), \hat{\mathbf{x}}_{-L+1}(j), ..., \hat{\mathbf{x}}_{-1}(j)) \tag{5.30}$$

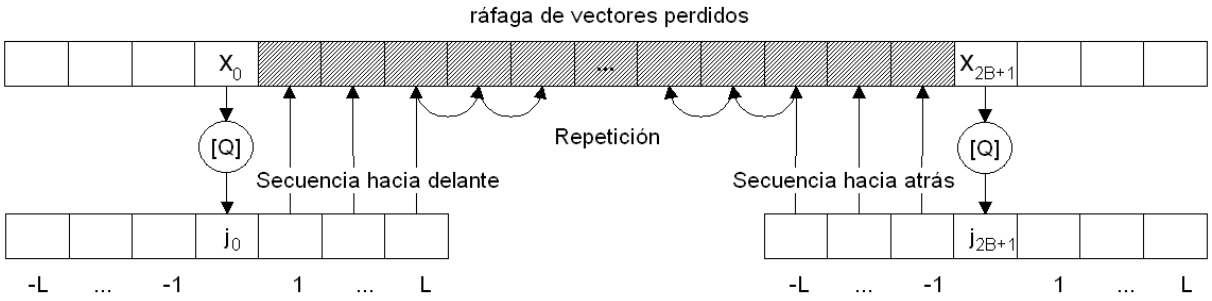


Figura 5.2: Ejemplo de reconstrucción basada en modelo de primer orden. La repetición se emplea para ráfagas superiores a la longitud de las secuencias estimadas.

Puesto que el conjunto de símbolos es finito, es posible precalcular y almacenar las secuencias hacia adelante y hacia atrás para cada uno de los símbolos. Este cálculo se puede realizar de forma muy eficiente sobre la base de datos de entrenamiento. Así, para calcular las secuencias de estimaciones correspondiente a un símbolo j, sólo han de aplicarse los siguientes pasos:

1. Se identifican, en la base de datos de entrenamiento, todas las apariciones del símbolo j (obtenido mediante una cuantización vectorial). Posterior y anteriormente a cada aparición del símbolo buscado, se halla, respectivamente, una secuencia de vectores hacia adelante y otra hacia atrás para dicho símbolo.

2. La secuencia de estimaciones hacia delante se puede estimar mediante el promediado de todas las secuencias de vectores posteriores al símbolo, mientras que la estimación hacia atrás se estima por el promedio de todas las secuencias de vectores anteriores al símbolo.

La cantidad de memoria requerida para almacenar estos datos es relativamente pequeña. Tan sólo es necesario almacenar una tabla de vectores estimados de tamaño $N \cdot 2L$, siendo N el número de símbolos y L la longitud de la secuencia (hay dos secuencias por símbolo).

Considerando una ráfaga que comienza en el instante $t = 1$ y termina en el instante $t = 2B$, los primeros B vectores se reconstruirán mediante la estimación hacia delante $E_F(j_0)$, donde j_0 es el último símbolo recibido antes de la ráfaga; y los últimos B pares de características se reconstruyen con la estimación hacia atrás $E_B(j_{2B+1})$, donde (j_{2B+1}) es el primer símbolo recibido después de la ráfaga. Puesto que las secuencias de estimaciones tienen una longitud finita L, puede ocurrir que la longitud de la ráfaga sea superior al número de estimaciones disponibles ($B > L$), en dicho caso recurriremos a la repetición de

los vectores estimados $\hat{\mathbf{x}}_L(j)$ y $\hat{\mathbf{x}}_{-L}(j)$ hacia la mitad de la ráfaga. La figura 5.2 muestra un ejemplo de reconstrucción mediante esta técnica.

5.3.3. Reconstrucción basada en modelo de segundo orden

El uso de un modelo de fuente de orden superior nos puede suministrar mejores estimaciones. Esto es especialmente útil en los canales con pérdidas, donde el modelo de voz, junto con los vectores recibidos, constituyen la única fuente de información disponible. Gracias a las simplificaciones efectuadas, la estrategia descrita en el subapartado anterior puede extenderse de forma que las estimaciones estén basadas en dos símbolos en vez de en uno solo. Para ello, la ecuación (5.28) se modifica de la forma:

$$\hat{\mathbf{x}}_l(j,k) = \sum_{i=0}^{N-1} \mathbf{x}^{(i)} P(\mathbf{x}_l^{(i)} | \mathbf{x}_0^{(j)}, \mathbf{x}_1^{(k)}) \quad (-L \leq l < 0), (1 < l \leq L+1) \tag{5.31}$$

tal que la secuencia hacia delante y hacia atrás venga dada por un par de símbolos,

$$E_F(j,k) = (\hat{\mathbf{x}}_2(j,k), \hat{\mathbf{x}}_3(j,k), ..., \hat{\mathbf{x}}_{L+1}(j,k)) \tag{5.32}$$

$$E_B(j,k) = (\hat{\mathbf{x}}_{-L}(j,k), \hat{\mathbf{x}}_{-L-1}(j,k), ..., \hat{\mathbf{x}}_{-1}(j,k)) \tag{5.33}$$

Es decir, en este caso deben ser calculadas una estimación hacia delante y hacia atrás por cada posible combinación de dos índices. La estimación de estas secuencias se realiza de forma análoga a la descrita en el apartado anterior, siendo igualmente eficiente, pero en este caso, teniendo en cuenta cada posible combinación de dos símbolos. Sin embargo, la cantidad de memoria requerida es significativamente mayor, necesitándose almacenar una tabla de vectores estimados de tamaño $N^2 \cdot 2L$.

Durante la reconstrucción de una ráfaga, los B primeros vectores son reconstruidos a través de la estimación hacia delante correspondiente a los dos últimos símbolos recibidos antes de la ráfaga $E_F(j_{-1}, j_0)$, y los últimos B vectores se reconstruyen a través de la estimación hacia atrás de los dos primeros símbolos recibidos tras la ráfaga $E_F(j_{2B+1}, j_{2B+2})$.

5.3.4. Reconstrucción basada en modelo de orden M reducido

Aunque teóricamente es posible extender la formulación anterior hacia un modelo de voz de orden M, esto resulta completamente inviable en la práctica. No sólo porque la memoria requerida para almacenar los vectores estimados crece de forma exponencial ($N^M \cdot 2L$),

sino también porque se requeriría una base de datos de entrenamiento casi ilimitada para poder entrenar la ingente cantidad de combinaciones posibles.

Precisamente de este último hecho surgen las bases de la reconstrucción que proponemos. Si analizamos las combinaciones de símbolos que aparecen en la base de datos de entrenamiento es fácil apreciar que no todas las combinaciones aparecen con la misma frecuencia, es más, algunas de ellas ni siquiera aparecen. Si una combinación tiene una baja frecuencia de aparición en la base de datos, presumiblemente las secuencias de estimaciones derivadas de ellas serán pobres, ya que se dispone de pocas muestras sobre las que calcularlas.

La reconstrucción basada en modelo de orden M reducido trabajará con lo que denominaremos *"registros"*. Un registro no es más que una secuencia de símbolos, de una determinada longitud M, que ha superado cierto proceso de criba. Inicialmente existen tantos registros como secuencias o combinaciones posibles de símbolos de longitud M (es decir, N^M combinaciones). La clave del modelo reducido consiste en considerar sólo aquellas combinaciones tales que su frecuencia de aparición en la base de datos es superior a un determinado umbral η. Estos registros se pueden obtener de forma eficiente mediante un algoritmo de comparación de todos con todos, en donde cada combinación de símbolos de la base de datos es comparada con el resto de combinaciones presentes en ella. De esta forma, puede extraerse un subconjunto de combinaciones cuya frecuencia de aparición es superior a η y que llamaremos registros.

Para estos registros se repite el mismo procedimiento de estimación que se realizaba en los modelos anteriores, pero teniendo en cuenta que ahora la estimación depende de los M símbolos que conforman un registro, $[j_1, j_2, \ldots, j_M]$. Formalmente,

$$\begin{aligned}\hat{\mathbf{x}}_l\left([j_1,\ldots,j_M]\right) &= \sum_{i=0}^{N-1}\mathbf{x}^{(i)}P\left(\mathbf{x}_l^{(i)}|\mathbf{x}_0^{(j_1)},\ldots,\mathbf{x}_{M-1}^{(j_M)}\right) \\ &\quad (-L\leq l<0),(M\leq l<M+L)\end{aligned} \tag{5.34}$$

tal que la secuencia hacia delante y hacia atrás viene dada por,

$$E_F\left([j_1,\ldots,j_M]\right) = (\hat{\mathbf{x}}_M([j_1,\ldots,j_M]),\ldots,\hat{\mathbf{x}}_{M+L-1}([j_1,\ldots,j_M])) \tag{5.35}$$

$$E_B\left([j_1,\ldots,j_M]\right) = (\hat{\mathbf{x}}_{-L}([j_1,\ldots,j_M]),\ldots,\hat{\mathbf{x}}_{-1}([j_1,\ldots,j_M])) \tag{5.36}$$

El procedimiento para obtener las secuencias de estimaciones hacia delante y hacia atrás es análogo al descrito en las secciones anteriores. Mediante una búsqueda se iden-

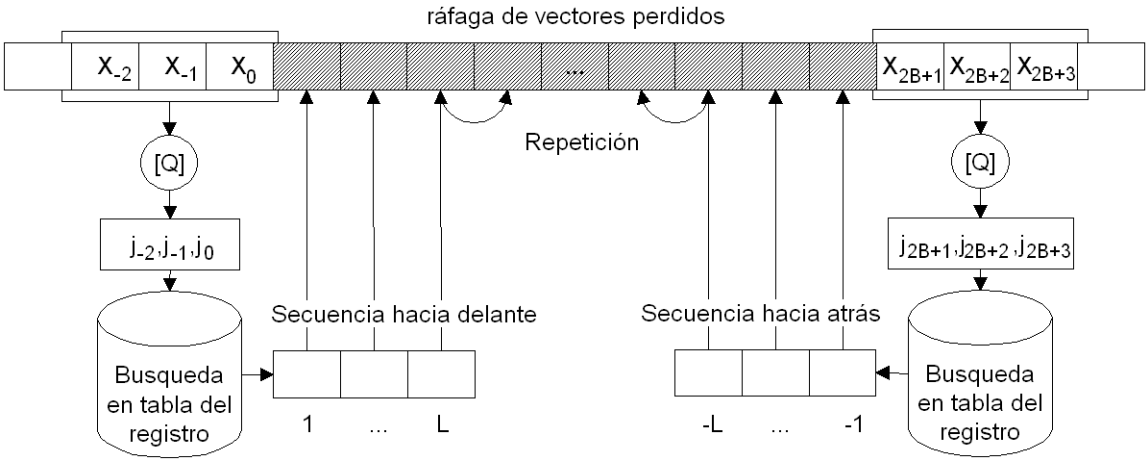


Figura 5.3: Ejemplo de reconstrucción basada en modelo de orden 3 reducido. La repetición se emplea para ráfagas superiores a la longitud de las secuencias estimadas.

tifican, en la base de datos de entrenamiento, todas las apariciones del registro al que se le pretenden calcular sus secuencias de estimaciones. Un promediado entre las secuencias posteriores y anteriores a cada aparición del registro permite obtener una estimación de la secuencia hacia delante y hacia atrás, respectivamente, para cada registro. Estas estimaciones, junto con el registro al que corresponden, quedan almacenadas en una tabla.

La reconstrucción de una ráfaga difiere ligeramente de la propuesta en las secciones anteriores. Para reconstruir el primer segmento de la ráfaga se construye un *"registro de referencia previo"* de longitud M. Si el primer vector perdido se produce en el instante $t = 1$, entonces se tomarán los símbolos anteriores $(j_{-(M-1)}, j_{-(M-2)}, \ldots, j_0)$. Cuando el registro de referencia previo se ha construido, se aplica una búsqueda en la tabla de registros. Una vez identificado, se extrae la correspondiente secuencia de estimaciones hacia delante, reemplazando los B primeros vectores de características perdidos. Al igual que antes, si B es mayor que L entonces las estimaciones más cercanas disponibles se repiten hacia la mitad de la ráfaga. La figura 5.3 muestra un ejemplo de reconstrucción mediante esta técnica.

Debido a la presencia de una ráfaga anterior a la que estamos tratando o al comienzo de una frase, es posible que el registro de referencia no pueda ser completamente construido. Es decir, puede ocurrir que no existan vectores recibidos en los instantes $t = (M - 1), \ldots, -r$. Se proponen entonces dos alternativas:

- Aproximación estática, en donde se aplica una mitigación sencilla sobre el propio registro de referencia consistente en la repetición de símbolos. Así, se supone que los

símbolos desconocidos, $(j_{-(M-1)}, \ldots, j_{-r})$, son idénticos al primer símbolo conocido j_{-r+1}. En este caso, la reconstrucción es idéntica a la descrita anteriormente.

- Aproximación dinámica, que consiste en reducir la longitud del registro de referencia desde M a r. Esta opción requiere calcular dinámicamente la secuencia de estimaciones hacia delante correspondiente a un registro $[j_1, \ldots, j_r]$ recibido. Las estimaciones de esta secuencia pueden obtenerse conocidas las correspondientes a todas las combinaciones de M símbolos como,

$$\hat{\mathbf{x}}_l\left([j_1, \ldots, j_r]\right) = \sum_{[i_1,\ldots,i_M]} \left[\hat{\mathbf{x}}_l\left([i_1, \ldots, i_M]\right) P\left(\mathbf{x}_0^{(i_1)}, \ldots, \mathbf{x}_{M-1}^{(i_M)} | \mathbf{x}_{M-1-r}^{(j_1)}, \ldots, \mathbf{x}_{M-1}^{(j_r)}\right)\right] \tag{5.37}$$

  Es decir, la secuencia de estimaciones hacia adelante correspondiente al registro de longitud r puede obtenerse como un promedio pesado (por la frecuencia de aparición) de las secuencias hacia delante de todas aquellas combinaciones de longitud M $([i_1, \ldots, i_M])$ que contienen la secuencia de longitud r al final $([i_1, \ldots, i_{M-r}, j_1, \ldots, j_r])$. Sin embargo, no disponemos de todas las combinaciones posibles, sino de un subconjunto de ellas (los registros). Por ello, tan sólo podremos aproximar con cierto sesgo estas estimaciones. Adicionalmente, se hace necesario almacenar la frecuencia de aparición de cada registro.

Para realizar la reconstrucción hacia atrás, de forma análoga a la anterior, se construye un *"registro de referencia posterior"*. Si el último vector perdido se produjo en el instante $t = 2B$, se toman los símbolos posteriores $(j_{2B+1}, j_{2B+2}, \ldots, j_{2B+M})$ para construir el registro de referencia. Tras ello, se realiza una búsqueda en la tabla y se reemplazan los últimos B pares de características con la correspondiente estimación hacia atrás. Al igual que antes, pueden aparecer problemas si existe una ráfaga después de la actual o se llega al final de la frase, de tal forma que no existan vectores en los instantes $t = 2B + r, \ldots, 2B + M$. Igualmente, se puede optar por dos opciones:

- Reemplazar los símbolos desconocidos, $(j_{2B+r}, \ldots, j_{2B+M})$, con copias del último símbolo conocido, j_{2B+r-1} (aproximación estática).

- Reducir dinámicamente la longitud de los registros (aproximación dinámica) y re-

calcular las estimaciones hacia atrás del registro recibido $[j_1, \ldots, j_r]$ como,

$$\hat{\mathbf{x}}_l\left([j_1, \ldots, j_r]\right) = \sum_{[i_1,\ldots,i_M]} \left[\hat{\mathbf{x}}_l\left([i_1, \ldots, i_M]\right) P\left(\mathbf{x}_0^{(i_1)}, \ldots, \mathbf{x}_{M-1}^{(i_M)} | \mathbf{x}_0^{(j_1)}, \ldots, \mathbf{x}_{r-1}^{(j_r)}\right)\right] \tag{5.38}$$

Esto es, mediante un promedio pesado, por la frecuencia de aparición, entre las estimaciones hacia atrás de todas las combinaciones de longitud M ($[i_1, \ldots, i_M]$) que contienen el registro de referencia de longitud r al principio ($[j_1, \ldots, j_r, i_{r+1}, \ldots, i_M]$).

La reducción de los requerimientos de memoria mediante la preselección de un conjunto de registros tiene como desventaja que, durante la mitigación de ráfagas, aparezcan casos en los que el registro de referencia no figure en la tabla precalculada. Es decir, no se disponga de una secuencia hacia delante o hacia atrás para una cierta combinación de símbolos. Por ello, esta técnica requiere siempre de una mitigación secundaria. Inicialmente proponemos como técnica secundaria la repetición del vector más cercano. Esto nos asegura que, en el peor caso, los resultados obtenidos sean iguales a los alcanzados por dicha técnica. Sin embargo, no es de esperar que estos casos se produzcan a menudo, ya que las combinaciones de símbolos eliminadas se caracterizan precisamente por una baja aparición en la base de datos de entrenamiento.

5.3.5. Complejidad computacional y requerimientos de memoria

Resulta evidente que la mayor carga computacional de estas técnicas reside en el cálculo de las secuencias de estimaciones. Sin embargo, al no depender de datos instantáneos, éstas pueden ser precalculadas, liberando al proceso de reconstrucción de esta tarea. Esta ventaja lleva asociado un inconveniente, los datos precalculados deben almacenarse, incrementando sustancialmente los requerimientos de memoria.

A diferencia de las técnicas basadas en modelos de primer y segundo orden, donde el orden del modelo fijaba los requerimientos de memoria, en el modelo de orden M reducido, el umbral η permite controlar estos requerimientos puesto que de él depende la cantidad de registros considerados. El uso de este umbral, en cambio, lleva asociado un pequeño incremento del coste computacional. En las técnicas anteriores, al mantenerse tablas exhaustivas, se puede llevar a cabo un acceso inmediato a la estimación, siendo tan sólo necesario la cuantización de los vectores recibidos antes y después de la ráfaga para obtener los símbolos que conforman el registro. En el modelo reducido no ocurre así,

siendo necesaria, además de las correspondientes cuantizaciones, la búsqueda del registro en la correspondiente tabla.

En principio puede emplearse un algoritmo de búsqueda lineal para encontrar el registro en la tabla. La complejidad asociada a una búsqueda lineal es de orden $O(n)$, donde n es el número de registros en la tabla. Un esquema más eficiente consiste en la ordenación de los registros de la tabla, de forma que sea posible aplicar un algoritmo de búsqueda por bipartición [146]. En este caso, la complejidad asociada se reduce a $O(\log_2 n)$. Igualmente, otros algoritmos más avanzados [146, 147], por ejemplo una búsqueda *hash* cuya complejidad viene dada por $O(1)$, podrían emplearse. Sin embargo, debe tenerse en cuenta que estos algoritmos de búsqueda sólo pueden aplicarse cuando se emplea la aproximación estática. En la aproximación dinámica no se busca un único registro, sino todos aquellos que contengan un determinado fragmento de longitud r.

Finalmente, si se dispone de un cuantizador preciso, es posible almacenar las secuencias de estimaciones hacia delante y hacia atrás de forma cuantizada. Esto permite una importante reducción en los requerimientos de memoria ya que, en vez de almacenar vectores completos, se almacenan únicamente sus índices de cuantización.

5.3.6. Resultados experimentales

Las técnicas propuestas anteriormente han sido evaluadas bajo un marco experimental basado en el sistema de referencia descrito en la sección 2.5. Dado que nuestro interés se centra ahora en la pérdida de información y no en otros ruidos derivados, este marco experimental emplea una arquitectura DSR para el reconocimiento remoto de la voz, suponiéndose un canal con pérdidas subyacente. En esta arquitectura, los vectores de características pueden decodificarse sin depender de la recepción del vector anterior, por lo que los efectos de una pérdida quedan limitados únicamente a los vectores perdidos (no aparece un ruido de memoria posterior). De esta forma, los resultados obtenidos a partir de las pruebas pueden interpretarse de una forma clara, ya que es no necesario considerar más efecto que el producido por la no disponibilidad los vectores durante una ráfaga.

De acuerdo con esto, la voz es analizada en el emisor, extrayéndose los parámetros de voz relevantes para el reconocimiento (vectores de características). Para esta tarea se emplea el front-end estandarizado por ETSI [74]. Se establecerá como canal con pérdidas la transmisión de los vectores mediante una red IP, propensa a la pérdida de paquetes. Así, los vectores de características serán empaquetados conforme al RFC 3557 [79] en donde se describe el formato de carga útil para DSR sobre IP. Como se comentó con anterioridad,

dicho estándar no especifica un número concreto de parejas de vectores (FP) que deben enviarse, sólo indica que debería minimizarse. Por ello, en los experimentos realizados se considera la transmisión de un único FP por paquete, esto es, cuando se produce una pérdida de un paquete son dos vectores de características los que se pierden.

Puesto que se pretenden evaluar únicamente técnicas de mitigación, la simulación de las pérdidas paquetes se realiza a través de un modelo de Gilbert, descrito con anterioridad en la sección 3.7.1. Este modelo permite evaluar las técnicas propuestas con diferentes tasas de pérdidas y con distintas longitudes de ráfaga. Se establecen un total de 25 condiciones de canal a fin de obtener un muestreo amplio del rendimiento de los algoritmos bajo diferentes condiciones. Estas condiciones implican un porcentaje global de pérdidas (R_{loss}) desde el 10 % al 50 %, en incrementos del 10 %, así como longitudes de ráfaga (L_{loss}) de 1, 2, 4, 8 y 12 paquetes. Como se mostró en la sección 3.7.2, las ráfagas largas (incluso con un porcentaje global de pérdidas bajo) son la principal causa de degradación. Para probar la bondad de las técnicas propuestas frente a condiciones adversas con ráfagas largas, algunas condiciones podrían mostrar porcentajes pocos realistas de pérdidas. Sin embargo, el propósito de esto no es más que el de suministrar un número significativo de ráfagas. Esto se debe a que, conforme se incrementa la longitud de las ráfagas (manteniendo el porcentaje global de pérdidas) el número de éstas es cada vez menor, ya que ráfagas más largas implican un mayor número de pérdidas.

La tabla 5.1 muestra los resultados obtenidos bajo estas condiciones aplicando como técnica de mitigación el algoritmo propuesto por el estándar, esto es, la repetición del vector recibido más próximo. Estos resultados constituyen la referencia con la que compararemos nuestras técnicas.

Durante la aplicación de las técnicas propuestas se reutilizará la codificación SVQ empleada por el estándar. De esta forma evitaremos un doble proceso de cuantización. Esto no supone una diferencia sustancial con respecto a la formulación original aplicada durante el desarrollo de las técnicas. Ahora, un símbolo es un índice de cuantización SVQ que representa un par de características (en vez de un vector completo). La estimación entonces debe realizarse para cada uno de los 7 pares de características (14 características en total) de forma independiente. En tal caso, el único aspecto a destacar es el diferente tamaño de los diccionarios empleados para cada par (6 diccionarios de 64 centros o símbolos para los $MFCC1-12$ y uno de 256 centros para el $MFFC0$ y $logE$).

Los resultados obtenidos empleando como técnica de mitigación la técnica FB–MMSE se muestran en la tabla 5.2. Como ya adelantáramos en la sección 5.3.2, esta técnica sólo supera ligeramente a la mitigación estándar para ráfagas muy cortas (longitud media de

Tasa de	*Long. media ráfaga*				
pérdidas	*1*	*2*	*4*	*8*	*12*
10 %	98.98	98.61	96.50	93.43	91.69
20 %	98.96	98.08	94.03	87.28	84.74
30 %	98.88	97.56	90.92	81.43	77.25
40 %	98.92	96.81	88.18	76.61	69.00
50 %	98.90	96.21	84.57	70.36	63.27

Tabla 5.1: Resultados de reconocimiento con repetición del vector más cercano.

Tasa de	*Long. media ráfaga*				
pérdidas	*1*	*2*	*4*	*8*	*12*
10 %	99.01	98.60	96.43	93.03	91.51
20 %	98.99	98.01	93.82	86.91	84.41
30 %	98.95	97.57	90.43	80.57	76.28
40 %	98.97	96.92	87.46	75.38	67.61
50 %	98.90	96.09	82.74	68.33	61.35

Tabla 5.2: Precisión del reconocimiento (Wacc) aplicando la técnica FB–MMSE.

Tasa de	*Long. media ráfaga*				
pérdidas	*1*	*2*	*4*	*8*	*12*
10 %	99.00	98.64	96.82	93.64	91.93
20 %	98.97	98.17	94.76	88.21	85.15
30 %	98.92	97.72	91.91	82.77	77.76
40 %	98.91	97.15	89.60	77.89	69.81
50 %	98.92	96.62	86.22	72.25	64.23

Tabla 5.3: Precisión del reconocimiento (Wacc) aplicando la técnica F+B–MMSE o reconstrucción basada en modelo de primer orden.

Tasa de	*Long. media ráfaga*				
pérdidas	*1*	*2*	*4*	*8*	*12*
10 %	99.01	98.67	96.97	94.27	92.68
20 %	98.97	98.23	95.08	89.16	86.71
30 %	98.92	97.74	92.49	84.06	79.46
40 %	98.92	97.26	90.08	79.76	72.36
50 %	98.94	96.57	86.89	74.17	67.38

Tabla 5.4: Precisión del reconocimiento (Wacc) aplicando la reconstrucción basada en modelo de segundo orden.

Tasa de pérdidas	*Long. media ráfaga*				
	1	*2*	*4*	*8*	*12*
10 %	99.03	98.74	97.33	94.82	93.22
20 %	98.98	98.25	95.53	90.16	87.65
30 %	98.94	97.95	93.42	85.68	81.21
40 %	98.94	97.49	91.22	81.58	74.54
50 %	98.98	96.88	88.43	76.51	69.75

Tabla 5.5: Precisión del reconocimiento (Wacc) aplicando la reconstrucción basada en modelo reducido de orden $N = 3$ estático.

Tasa de pérdidas	*Long. media ráfaga*				
	1	*2*	*4*	*8*	*12*
10 %	99.01	98.67	97.17	94.47	92.75
20 %	98.97	98.15	95.14	89.44	86.79
30 %	98.94	97.76	92.77	84.65	80.01
40 %	98.92	97.32	90.40	80.07	72.62
50 %	98.93	96.74	87.05	74.80	67.67

Tabla 5.6: Precisión del reconocimiento (Wacc) aplicando la reconstrucción basada en modelo reducido de orden $N = 3$ dinámico.

un paquete). Para el resto de condiciones, la degradación mutua que sufren las probabilidades da lugar a una reducción del rendimiento. Como comentábamos entonces, es posible evitar esta degradación mutua por medio de una estrategia F+B–MMSE. La tabla 5.3 muestra los resultados obtenidos mediante esta técnica. La longitud de las secuencias de estimaciones se ha establecido en 20 vectores ($L = 20$), es decir, sin recurrir a la repetición, el algoritmo puede recuperar hasta 20 paquetes perdidos, ofreciendo un margen seguro para las condiciones con longitudes medias de ráfaga más largas. Como puede observarse, ésta supera el rendimiento de la técnica FB–MMSE, obteniendo una ligera mejora global sobre la reconstrucción estándar.

Estos resultados pueden mejorarse si, en vez de considerar un modelo de primer orden, consideramos uno de orden superior. La tabla 5.4 muestra los resultados obtenidos empleando la reconstrucción MMSE basada en modelo de segundo orden, descrita en la sección 5.3.3. Al igual que antes, la longitud de las secuencias de estimaciones se ha establecido en 20 vectores ($L = 20$). Como puede observarse, esta técnica ofrece una mejora significativa del rendimiento del reconocedor para todas las condiciones propuestas en comparación con las técnicas anteriores.

Las tablas 5.5 y 5.6 muestran los resultados obtenidos empleando la reconstrucción ba-

Tasa de pérdidas	*Long. media ráfaga*				
	1	*2*	*4*	*8*	*12*
10 %	98.97	98.65	96.99	94.29	92.85
20 %	98.94	98.21	95.12	89.20	86.76
30 %	98.89	97.74	92.85	84.46	80.12
40 %	98.92	97.25	90.50	80.16	73.49
50 %	98.94	96.76	87.42	75.00	68.07

Tabla 5.7: Precisión del reconocimiento (Wacc) aplicando la reconstrucción basada en modelo reducido de orden $N = 4$.

sada en un modelo de orden $M = 3$ reducido con un umbral para la obtención de registros de $\eta = 10$ apariciones empleando la aproximación estática y dinámica, respectivamente. Mediante tablas exhaustivas, la extensión hacia un modelo de tercer orden requeriría almacenar las secuencias de estimaciones de más de 18 millones de combinaciones posibles ($6 \cdot 64^3 + 256^3$). Sin embargo, limitando el almacenamiento a sólo aquellos registros que aparecen más de 10 veces, el número se reduce a 131.462 registros, es decir, los requerimientos de memoria se reducen en dos órdenes de magnitud. Los resultados de estas tablas se corresponden con las dos posibles soluciones presentadas para la construcción de registros cuando no están los M índices disponibles. En la primera se usan copias del vector recibido más cercano (apróximación estática), mientras que en la segunda se reduce dinámicamente la longitud de los registros, recalculándose las estimaciones (aproximación dinámica). En ambos casos se ha establecido una longitud de secuencia de 20 vectores. Dado que la cuantización SVQ realizada por el front-end ha demostrado no afectar al reconocimiento [51], las estimaciones precalculadas han sido almacenadas empleando esta codificación, lo cual implica una nueva reducción de los requerimientos de memoria (aproximadamente a una octava parte).

En comparación, la aproximación estática resulta claramente superior, conduciendo a una mejora sustancial de los resultados (la aproximación dinámica sólo produce una mejora marginal con respecto a un modelo de segundo orden). A esto debe añadirse la menor complejidad computacional de la aproximación estática. Aplicando un algoritmo de bipartición, la búsqueda sobre los 131.462 registros requiere menos de 18 pasos. Dado que ésta es la operación de mayor complejidad del algoritmo (las estimaciones están precalculadas), el coste computacional puede compararse al de la mitigación estándar.

Finalmente, la tabla 5.7 muestra los resultados considerando un modelo reducido de cuarto orden, con $\eta = 10$ y una aproximación estática. Puede observarse que, aunque se obtiene una importante mejora con respecto a la mitigación estándar, los resultados son

inferiores al modelo reducido de orden 3 estático. La reducción del rendimiento se justifica en base a un problema de insuficiencia de datos. Dado que existe un mayor número de combinaciones posibles en un modelo de cuarto orden, es de esperar un incremento del numero de registros. Sin embargo, al incrementar la longitud de los registros, su frecuencia de aparición en la base de datos de entrenamiento es cada vez menor. Así, durante el entrenamiento de este modelo se observó que el número de combinaciones aceptadas ($\approx$ 95000) resultaba inferior a un modelo de orden 3 con los mismos parámetros. Este hecho revela que se ha elegido para la técnica un modelo de orden excesivo, ya que la base de datos resulta insuficiente para entrenarlo. En este caso es preferible usar la reconstrucción basada en un modelo de orden inmediatamente inferior.

5.4. Estimación basada en Máximo a Posteriori

5.4.1. Fundamentos de la estimación MAP

La estimación MAP se basa en el cálculo del valor del vector correspondiente al instante de tiempo t que maximice la probabilidad condicionada a los valores de los vectores observados, es decir,

$$\hat{\mathbf{x}}_t = arg\, max_{\mathbf{x}_t} \left\{ P\left(\mathbf{x}_t | \Lambda\right) \right\} \tag{5.39}$$

donde la probabilidad condicionada vendrá dada por el modelo de voz considerado, siendo Λ el conjunto de vectores de voz observados.

En el contexto de canales con pérdidas, puede considerarse un conjunto completo de vectores $\mathbf{X}$ compuesto por un subconjunto de vectores perdidos, $\mathbf{X}_m$, y un subconjunto de vectores observados o recibidos, $\mathbf{X}_o$. La estimación MAP consiste en encontrar aquella secuencia de vectores perdidos tal que, de acuerdo con el modelo considerado, maximice su probabilidad condicionada a los vectores recibidos, esto es,

$$\hat{\mathbf{X}}_m = arg\, max_{\mathbf{X}_m} \left\{ P\left(\mathbf{X}_m | \mathbf{X}_o\right) \right\} \tag{5.40}$$

Desde un punto de vista computacional, la estimación MAP puede resultar atractiva a la hora de obtener una estimación de la secuencia de vectores perdidos óptima. Por simplicidad, puede asumirse que los conjuntos de vectores recibidos y perdidos están organizados en un único vector, tal que, el conjunto completo de vectores viene dado por,

$$\mathbf{X} = [\mathbf{X}_o, \mathbf{X}_m] \tag{5.41}$$

Esto no supone ninguna pérdida de generalidad ya que cualquier conjunto de M elementos puede representarse como un vector en un espacio M-dimensional, en donde el orden de los componentes simplemente denota cómo se organizan las distintas dimensiones.

Supuesto que la distribución de $\mathbf{X}$ es una gaussiana multivariada de vector medio $\boldsymbol{\mu}$ y matriz de covarianza Σ, $P(\mathbf{X};\boldsymbol{\mu},\Sigma)$, se demuestra que las distribuciones de $\mathbf{X}_o$ y $\mathbf{X}_m$, $P(\mathbf{X}_o;\boldsymbol{\mu},\Sigma)$ y $P(\mathbf{X}_m;\boldsymbol{\mu},\Sigma)$ respectivamente, son igualmente gaussianas multivariadas [148]. Si $\boldsymbol{\mu}_o$ y $\boldsymbol{\mu}_m$ son, respectivamente, los vectores medios de $\mathbf{X}_o$ y $\mathbf{X}_m$, así como Σ_{oo} y Σ_{mm} sus matrices de covarianza, entonces,

$$\boldsymbol{\mu} = [\boldsymbol{\mu}_o, \boldsymbol{\mu}_m] \tag{5.42}$$

y

$$\Sigma = \begin{bmatrix} \Sigma_{oo} & \Sigma_{om} \\ \Sigma_{mo} & \Sigma_{mm} \end{bmatrix} \tag{5.43}$$

donde Σ_{om} es la covarianza cruzada entre $\mathbf{X}_o$ y $\mathbf{X}_m$, y $\Sigma_{mo} = \Sigma_{om}^T$. Puede mostrarse entonces que,

$$\begin{aligned} P\left(\mathbf{X}_m|\mathbf{X}_o;\boldsymbol{\mu},\Sigma\right) = C \cdot \exp[-\frac{1}{2}\cdot & \left(\mathbf{X}_m - \boldsymbol{\mu}_m - \Sigma_{mo}\Sigma_{oo}^{-1}(\mathbf{X}_o - \boldsymbol{\mu}_o)\right)^T \cdot \\ & (\Sigma_{mm} - \Sigma_{mo}\Sigma_{oo}^{-1}\Sigma_{om})^{-1} \cdot \\ & (\mathbf{X}_m - \boldsymbol{\mu}_m - \Sigma_{mo}\Sigma_{oo}^{-1}(\mathbf{X}_o - \boldsymbol{\mu}_o))] \end{aligned} \tag{5.44}$$

donde C es una constante de normalización. Se deriva entonces de las expresiones (5.40) y (5.44) [149],

$$\hat{\mathbf{X}}_m = arg\,max_{\mathbf{X}_m}\left\{P\left(\mathbf{X}_m|\mathbf{X}_o;\boldsymbol{\mu},\Sigma\right)\right\} = \boldsymbol{\mu}_m + \Sigma_{mo}\Sigma_{oo}^{-1}(\mathbf{X}_o - \boldsymbol{\mu}_o) \tag{5.45}$$

Esto es, bajo las suposiciones anteriores, la estimación MAP queda simplificada a una regresión lineal.

Una hipótesis sencilla a la hora de establecer el modelo estadístico de la voz consiste en considerar la secuencia de vectores de características como la salida de un proceso aleatorio estacionario en sentido amplio con una distribución gaussiana. De esta forma, todos los posibles vectores se asumen como observaciones individuales de un único proceso. Durante la reconstrucción MAP, los parámetros estadísticos que definen este proceso son empleados para la obtención de las estimaciones de los vectores perdidos.

Sea $X(t_1, k_1)$ el k_1-ésimo componente del vector de características en el instante de tiempo t_1, y $X(t_2, k_2)$ el k_2-ésimo componente del vector de características en el instante

t_2. El valor medio de dichas componentes, así como la covarianza cruzada entre ellas se define como,

$$\begin{aligned} &\mu(t_1, k_1) = E[X(t_1, k_1)] \\ &\mu(t_2, k_2) = E[X(t_2, k_2)] \\ &c\,(t_1, t_2, k_1, k_2) = E\left[\left(X(t_1, k_1) - \mu(t_1, k_1)\right)\left(X(t_2, k_2) - \mu(t_2, k_2)\right)\right] \end{aligned} \tag{5.46}$$

donde $E[\cdot]$ representa el valor esperado. La suposición de estacioneidad en sentido amplio implica que las medias de los vectores y las covarianzas entre los componentes de los vectores son independientes del tiempo, derivándose las siguientes expresiones,

$$\begin{aligned} &\mu(t_1, k) = \mu(t, k) = \mu(k) \\ &c\,(t_1, t_1 + \tau, k_1, k_2) = c\,(t, t + \tau, k_1, k_2) = c\,(\tau, k_1, k_2) \end{aligned} \tag{5.47}$$

Es decir, el valor esperado $\mu(k)$ de la k-esima componente no depende del instante temporal en el que ocurra en la secuencia de vectores. Igualmente, la covarianza entre dos componentes sólo depende de la distancia temporal entre ellas y no de del instante en que se producen en la secuencia. La media de los componentes de los vectores de características, así como las distintas covarianzas $c(\tau, k_1, k_2)$ pueden ahora obtenerse de la base de datos de entrenamiento, como,

$$\begin{aligned} &\mu(k) = E[X(t, k)] \\ &c\,(\tau, k_1, k_2) = E\left[\left(X(t, k_1) - \mu(k_1)\right)\left(X(t + \tau, k_2) - \mu(k_2)\right)\right] \end{aligned} \tag{5.48}$$

La suposición de un proceso aleatorio con distribución gaussiana implica que la distribución conjunta de cualquier subconjunto de componentes es también gaussiano. Por tanto, estas medias y covarianzas serán los únicos parámetros necesarios para estimar los vectores perdidos, ya que describen completamente el proceso aleatorio.

A la hora de reconstruir una secuencia de vectores incompleta $\mathbf{X}$, los componentes de los vectores recibidos son organizados en un vector $\mathbf{X}_o$, mientras que los componentes de los vectores perdidos se organizan en un vector $\mathbf{X}_m$. Dado que conocemos los valores medios de todos los componentes, así como las covarianzas entre cualesquiera de ellos, pueden construirse tanto los vectores medios $\boldsymbol{\mu}_o$ y $\boldsymbol{\mu}_m$, como la matriz de autocovarianza Σ_{oo} y la matriz de covarianza cruzada Σ_{mo} entre $\mathbf{X}_m$ y $\mathbf{X}_o$. Estos parámetros permiten obtener una estimación MAP de las componentes de los vectores perdidos mediante la ecuación

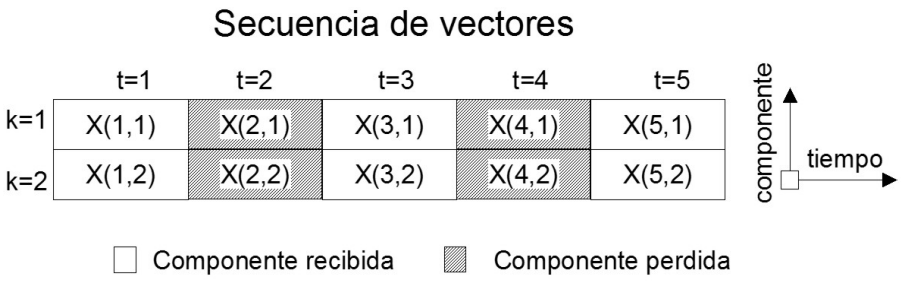


Figura 5.4: Secuencia de ejemplo sencilla en la que se aplicará la reconstrucción MAP.

(5.45). La técnica resultante se denomina *reconstrucción basada en covarianzas*, por el uso intensivo que realiza de estos estimadores. Ésta fue propuesta por Ramakrishnan [149] y, aunque originalmente fue ideada para la reconstrucción de espectrogramas incompletos como mecanismo para el robustecimiento frente a ruido acústico, puede extenderse con facilidad a la reconstrucción de vectores perdidos.

Sirva un ejemplo para mostrar la aplicación de esta técnica en el contexto de canales con pérdidas. En la figura 5.4 se muestra una secuencia muy sencilla de cinco vectores, en la que el primer, tercer y quinto vector se han recibido y el segundo y cuarto se han perdido. Por simplicidad, se supone que los vectores constan sólo de dos componentes. En primer lugar deben construirse los vectores $\mathbf{X}_o$ y $\mathbf{X}_m$, estos son respectivamente,

$$\mathbf{X}_o = [X(1,1), X(1,2), X(3,1), X(3,2), X(5,1), X(5,2)]^T$$

$$\mathbf{X}_m = [X(2,1), X(2,2), X(4,1), X(4,2)]^T$$

Los valores de este último son desconocidos. Sin embargo, al suponerse un proceso estacionario en sentido amplio, el valor esperado para cada componente es el mismo independientemente del instante de tiempo en que ha sido recibida. Por tanto, los vectores medios de $\boldsymbol{\mu}_o$ y $\boldsymbol{\mu}_m$ quedan construidos como,

$$\boldsymbol{\mu}_o = [\mu(1), \mu(2), \mu(1), \mu(2), \mu(1), \mu(2)]$$

$$\boldsymbol{\mu}_m = [\mu(1), \mu(2), \mu(1), \mu(2)]$$

La matriz de autocovarianza de $\mathbf{X}_o$ es una matriz 6x6 dada por,

$$\Sigma_{oo} = \begin{pmatrix} c(0,1,1) & c(0,1,2) & c(2,1,1) & \dots & c(4,1,2) \\ c(0,2,1) & c(0,2,2) & c(2,1,2) & \dots & c(4,2,2) \\ c(-2,1,1) & c(-2,1,2) & c(0,1,1) & \dots & c(2,1,2) \\ \dots & \dots & \dots & \dots & \dots \\ c(-4,2,1) & c(-4,2,2) & c(-2,2,1) & \dots & c(0,2,2) \end{pmatrix}$$

en donde cada elemento se obtiene de la siguiente forma,

$$\Sigma_{oo}(n,m) = c\left(\Delta[\mathbf{X}_o(n), \mathbf{X}_o(m)], K[\mathbf{X}_o(n)], K[\mathbf{X}_o(m)]\right) \tag{5.49}$$

donde $\Delta[\cdot]$ y $K[\cdot]$ son dos operadores definidos como,

$$\Delta\left[X(t_1,k_1), X(t_2,k_2)\right] = t_1 - t_2 \tag{5.50}$$

$$K\left[X(t,k)\right] = k \tag{5.51}$$

La matriz de covarianza cruzada entre $\mathbf{X}_m$ y $\mathbf{X}_o$ es una matriz de 4x6 dada por,

$$\Sigma_{mo} = \begin{pmatrix} c(-1,1,1) & c(-1,1,2) & c(1,1,1) & \dots & c(3,1,2) \\ c(-1,2,1) & c(-1,2,2) & c(1,1,2) & \cdots & c(3,2,2) \\ c(-3,1,1) & c(-3,1,2) & c(-1,1,1) & \dots & c(1,1,2) \\ c(-3,2,1) & c(-3,2,2) & c(-1,2,1) & \dots & c(1,2,2) \end{pmatrix}$$

o, de forma algebraica, como,

$$\Sigma_{mo}(n,m) = c\left(\Delta[\mathbf{X}_m(n), \mathbf{X}_o(m)], K[\mathbf{X}_m(n)], K[\mathbf{X}_o(m)]\right) \tag{5.52}$$

Construidos todos estos parámetros, la estimación $\hat{\mathbf{X}}_m$ se calcula aplicando la expresión (5.45).

5.4.2. Reconstrucción MAP progresiva

El mayor inconveniente de la reconstrucción MAP, empleando un esquema como el anterior, es que ésta debe realizarse de forma global, es decir, en un único paso se reconstruyen todos los vectores perdidos en base a todos los vectores recibidos. Si se supone una frase corta representada por 200 vectores (2 segundos), con 14 características por vector, en donde el 50 % de los vectores se han perdido, las matrices Σ_{oo} y Σ_{mo} llegan a contar con 1400×1400 elementos. La estimación requeriría entonces de la inversión de una ma-

triz de 1400×1400, además de la multiplicación de dos matrices de 1400×1400. Para frases más largas el cómputo requerido es mayor. Incluso suponiendo frases cortas, si el porcentaje de pérdidas es pequeño, la complejidad computacional se incrementa significativamente, ya que la dimensionalidad de la matriz de autocovarianzas (observaciones), que debe invertirse, es mayor. Claramente esta aproximación resulta impracticable, no solo por la prohibitiva cantidad de cálculo, sino también por la elevada latencia que implicaría así como el número de covarianzas $c(\tau, k_1, k_2)$ que deberían ser estimadas.

Obviamente, la solución consiste en aplicar la reconstrucción no de forma global, sino localmente en cada ráfaga, empleando como observaciones los vectores recibidos antes y después de ésta (figura 5.5a). Sin embargo, esta aproximación sigue presentando algunos inconvenientes:

- Al establecer el número de vectores recibidos que se considerarán durante la reconstrucción, el tamaño de la matriz de autocovarianza está controlado, pero no ocurre lo mismo con la matriz de covarianzas cruzadas. El tamaño de esta matriz dependerá del número de vectores perdidos durante la ráfaga. Esto impide predecir con facilidad los requerimientos máximos de memoria y la complejidad computacional del algoritmo.

- Debido a esta dependencia con el tamaño de la ráfaga, es difícil establecer un valor mínimo y máximo de τ. Los valores extremos de la distancia temporal son necesarios a la hora de definir el conjunto de covarianzas $c(\tau, k_1, k_2)$ que se estimarán sobre la base de datos.

Aunque el primer problema tiene una sencilla solución: trabajar sobre vectores perdidos (en vez de sobre la ráfaga completa); la solución al segundo problema implica realizar suposiciones sobre la longitud máxima de las ráfagas. A fin de evitar estos problemas, en este trabajo se propone la reconstrucción progresiva de éstas.

En esta reconstrucción, la estimación MAP se realiza por cada vector perdido, aplicándose una ventana que incluye los M vectores anteriores y los M posteriores, y de la cual se extraen las observaciones. Así, el vector $\mathbf{X}_m$ se compone únicamente del vector perdido $\mathbf{x}_t$, mientras que el vector $\mathbf{X}_o$ se construye con aquellos vectores recibidos dentro de una ventana de tamaño $2M + 1$ centrada en el vector perdido. De esta forma, es posible predecir los valores extremos de τ ($-2M \leq \tau \leq 2M$), así como la complejidad del algoritmo (como máximo se invierten matrices de $2MK \times 2MK$, siendo K el número de características por vector).

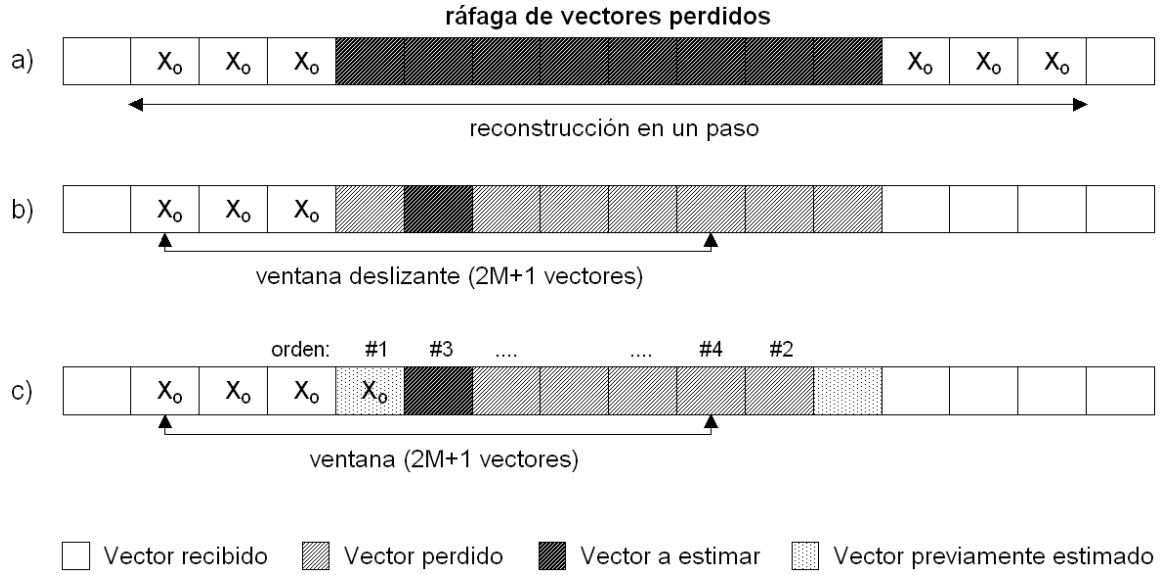


Figura 5.5: Estrategias para la reconstrucción de ráfagas mediante estimación MAP: a) Por ráfagas completas, b) Reconstrucción no progresiva, c) Reconstrucción progresiva desde los extremos al centro.

Puesto que la complejidad computacional depende del valor de M, es deseable establecer ventanas de reducido tamaño. Sin embargo, tamaños de ventana reducidos pueden conducir a ventanas sin vectores recibidos durante las ráfagas. Esto es, si una ráfaga de longitud $2M+1$ o superior aparece, la ventana quedará vacía durante ciertas estimaciones (figura 5.5b). Con el objetivo de evitar esta situación, se considerarán como recibidos aquellos vectores previamente recuperados. Esta suposición conlleva que implícitamente se esté degradando las estimaciones. Para evitar que la degradación se vaya acumulando hasta el final de la ráfaga, se puede aplicar una estrategia progresiva desde los extremos hacia el centro. Es decir, si se supone una ráfaga desde $t = 1$ a $t = T$, primero se reconstruirán los vectores $\mathbf{x}_1$ y $\mathbf{x}_T$, luego los $\mathbf{x}_2$ y $\mathbf{x}_{T-1}$ y así sucesivamente hasta la mitad de ésta (figura 5.5c). De esta forma, la degradación que supone considerar vectores previamente recuperados como recibidos se acumula en el centro de la ráfaga donde, al existir cierta distancia temporal con respecto a los vectores recibidos, ya se espera que la estimación ofrezca peores reemplazos.

5.4.3. Reconstrucción MAP por bandas cepstrales

Durante la estimación MAP, no todas las componentes observadas contribuyen de la misma forma a la estimación de cada componente perdida. Esto puede aprovecharse para

reducir la complejidad de la estimación. Supongamos que en vez de trabajar con vectores completos, operamos componente a componente, la reconstrucción MAP queda entonces definida como,

$$\hat{X}(t,k) = \mu(k) + \mathbf{c}_{mo}(t,k)\Sigma_{oo}^{-1}(\mathbf{X}_o - \boldsymbol{\mu}_o) \tag{5.53}$$

donde $X(t,k)$ es la característica k del vector t que queremos estimar, $\mu(k)$ es el valor esperado de la característica, y $\mathbf{c}_{mo}(t,k)$ es el vector de covarianzas cruzadas entre la componente $X(t,k)$ y las componentes observadas que conforman el vector $\mathbf{X}_o$. La estimación obtenida por la ecuación (5.53) es idéntica a la que se obtendría empleando la ecuación (5.45) con vectores completos. Desgraciadamente, la complejidad no se reduce, ya que la dimensión de la matriz Σ_{oo} (que debe invertirse) tampoco es distinta a la de la ecuación (5.45). Supongamos entonces que sólo una única observación, $X_o(t+\tau, k_1)$, se tiene en cuenta durante la estimación. La expresión (5.53) queda reducida a,

$$\hat{X}(t,k) = \mu(k) + \frac{c(\tau,k,k_1)}{c(0,k_1,k_1)}(X_o(t+\tau,k_1) - \mu(k_1)) \tag{5.54}$$

que puede expresarse en términos de la covarianza relativa entre componentes, $r(\tau,k,k_1)$, como,

$$\hat{X}(t,k) = \mu(k) + r(\tau,k,k_1)\sqrt{\frac{c(0,k,k)}{c(0,k_1,k_1)}}(X_o(t+\tau,k_1) - \mu(k_1)) \tag{5.55}$$

en donde,

$$r(\tau,k,k_1) = \frac{c(\tau,k,k_1)}{\sqrt{c(0,k,k)c(0,k_1,k_1)}} \tag{5.56}$$

En la expresión (5.55) puede observase claramente que, conforme la covarianza relativa entre la componente estimada y la observada disminuye, la contribución de la observación a la estimación decrece linealmente. De esta forma, para valores pequeños de $r(\tau,k,k_1)$ la contribución a la estimación es prácticamente nula. Extendiendo el razonamiento a todas las observaciones, puede concluirse que cada componente recibida da lugar a una corrección sobre la media de la componente estimada, ponderada en base a su covarianza relativa. Puesto que aquellas observaciones que comparten una baja covarianza relativa tienen una contribución a la estimación pequeña, podrían no ser tenidas en cuenta durante la estimación. Esto conduciría a un pequeño error en la estimación, pero permitiría una reducción de la dimensión de Σ_{oo}.

Así pues, una forma de reducir la complejidad del algoritmo consistiría en definir un umbral r_{min} de forma que, durante la estimación de cada componente k, sólo se consideren

aquellas componentes observadas tales que,

$$r(\tau, k, k_1) > r_{min} \tag{5.57}$$

Sin embargo, un análisis de las covarianzas relativas implicadas puede conducir a estrategias más eficientes. La figura 5.6 muestra la covarianza relativa $r(\tau, k_1, k_2)$, para diferentes componentes k_1 del vector de características, en función de k_2 y τ. Se ha considerado que el vector de características cuenta con 14 componentes, las 12 primeras representando los coeficientes cepstrales del orden 1 al 12, y las dos últimas los coeficientes energéticos MFCC0 y logE. Por razones de espacio sólo se muestran la distribución de covarianza relativa del MFCC1, MFCC3, MFCC5, MFCC8, MFCC10 y LogE. De forma general, pueden hacerse las siguientes observaciones:

- Como es de esperar, la covarianza relativa decrece tanto al considerar diferentes características, como al incrementar el lapso temporal entre ellas. Obviamente, los valores máximos se producen en $r(0, k, k)$ y son iguales a 1.
- La covarianza relativa de cada coeficiente cepstral (a excepción del MFFC0) respecto a los restantes coeficientes cepstrales es prácticamente nula. Esto es, $r(\tau, k_1, k_2) \sim 0$, con $k_1, k_2 \leq 12$ y $k_1 \neq k_2$, para cualquier valor de τ.
- Para los coeficientes ceptrales de orden alto, la conclusión anterior es extensible también a los parámetros energéticos. Para estas componentes, las distribuciones están claramente dominadas por la covarianza relativa con la misma componente a distintos desplazamientos temporales (gráficas MFCC5-10).
- Las características energéticas comparten valores altos de covarianza relativa entre ellas (última gráfica). Aunque $r(\tau, logE, MFCC0)$ disminuye conforme τ aumenta, sus valores son próximos a $r(\tau, logE, logE)$.
- Finalmente, las covarianzas relativas para los coeficientes cepstrales de orden bajo con respecto a los coeficientes energéticos, presentan valores relativamente altos, especialmente el MFCC1 (primera gráfica). Lo mismo ocurre, aunque en menor medida, para las covarianzas relativas del LogE (última gráfica) y el MFCC0 con respecto a los coeficientes cepstrales de orden bajo.

Todas estas observaciones resultan coherentes con las operaciones implicadas en la extracción de características. Por un lado, los coeficientes cepstrales se obtienen por medio

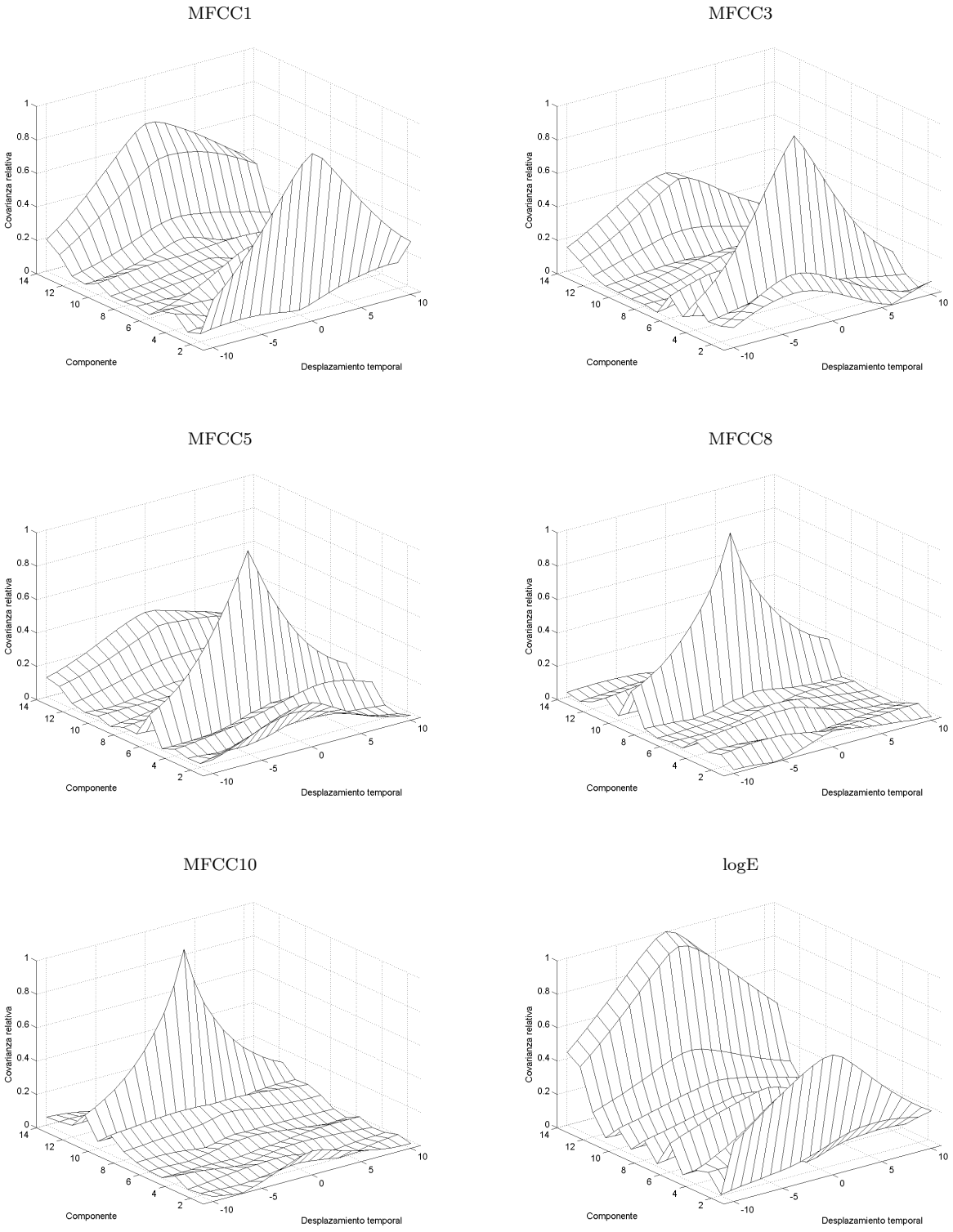


Figura 5.6: Covarianzas relativas para MFCC1, MFCC3, MFCC5, MFCC8, MFCC10 y LogE con respecto a las 14 restantes características, considerando diferentes desplazamientos temporales ($10 \leq \tau \leq 10$).

de una transformada coseno que, justamente, decorrela estos parámetros. Resulta evidente por tanto que presenten una covarianza relativa entre ellos casi nula. Por otro lado, se espera que los coeficientes energéticos presenten una alta covarianza relativa entre ellos, ya que representan de forma distinta una misma característica del segmento de voz, esto es, su energía. Finalmente, también es razonable que se presenten covarianzas relativas significativas entre componentes espectrales y componentes energéticas. La energía del segmento está relacionada con la sonoridad de dicho segmento. Precisamente por ello, estos parámetros se incluyen el vector de características para mejorar la discriminación entre fonemas sordos y sonoros, como se vio en la sección 2.2.4.

Puesto que las componentes espectrales tienen una covarianza relativa casi nula con respecto al resto de componentes espectrales, la contribución de estas últimas a la estimación puede despreciarse. En base a esto proponemos una reconstrucción MAP por bandas cepstrales. En ella, la estimación de una componente espectral del vector de características se realiza teniendo en cuenta sólo aquellas observaciones recibidas en la misma banda de cuefrencia. Esto es, sólo se consideran aquellas componentes que representen a la misma característica en instantes anteriores o posteriores al estimado, descartándose el resto de componentes. Por simplicidad, extenderemos esta estrategia a las componentes energéticas aunque ello conlleve cierto sesgo en la estimación.

Al igual que antes, para cada vector a estimar se definirá una ventana de tamaño $2M+1$ centrada en él, de donde extraeremos las observaciones. Sin embargo, ahora operamos componente a componente, en vez de con vectores completos. Para ello recurriremos a la ecuación (5.53). El vector de observaciones $\mathbf{X}_o$ empleado durante la estimación de una componente $\hat{X}(t, k_1)$, se construye con todas aquellas componentes recibidas $X(t_2, k_2)$ tales que $(k_2 = k_1)$ y $(-M < t_1 - t_2 < M)$. Esto conduce a una reducción significativa de la complejidad computacional, ya que las matrices de observaciones constan ahora de $2M \times 2M$ elementos como máximo.

Reconstrucción no progresiva de las ráfagas

Una de las ventajas asociada al uso de esta estrategia es que se pueden considerar ventanas de mayor tamaño. Gracias a ello, se puede llegar a prescindir de la reconstrucción progresiva descrita con anterioridad.

Supongamos una ráfaga suficientemente larga que se reconstruye de forma no progresiva, esto es, sin considerar vectores previamente estimados como recibidos. Si consideramos

entonces una ventana de tamaño $2M+1$ centrada en el vector a estimar, puede observarse que el vector perdido en el instante de tiempo $t+M+1$, donde t es el instante de tiempo en el que se recibió el último vector antes de la ráfaga, no cuenta con ninguna observación sobre la que ser estimado (ventana vacía). Consideremos entonces el vector recibido en el instante de tiempo justamente anterior $(t+M)$. Durante la estimación de cada componente de este vector, sólo se considerará como observación una única componente, recibida M instantes de tiempo anteriores. En dicho caso, la estimación MAP se reduce a la expresión (5.55), descrita anteriormente.

Como mencionábamos antes, la contribución que hace esta observación a la estimación viene dada por su covarianza relativa, que en nuestro caso será $r(-M,k,k)$. De acuerdo con las gráficas de la figura 5.6, conforme tomamos valores de M mayores, $r(-M,k,k)$ decrece rápidamente. Puede observarse entonces que, para cierto valor M_{max}, la covarianza relativa $r(-M_{max},k,k)$ es suficientemente baja como para que la estimación de la componente $X(k,t+M_{max})$ venga dada prácticamente por $\mu(k)$, esto es, su media global. Estimación que precisamente corresponde a $X(k,t+M_{max}+1)$, en donde emplearíamos una ventana vacía. Exactamente lo mismo ocurre al otro extremo de la ráfaga pero considerando $r(M,k,k)$.

Por tanto, no merece la pena definir una ventana de tamaño mayor a M_{max}, ya que, aunque así consideraríamos al menos una observación para $X(k,t+M_{max}+1)$, la covarianza relativa entre estas componentes sería tan baja que la estimación obtenida sería prácticamente igual a si considerásemos una ventana vacía. Así pues, para cada componente, puede definirse un tamaño máximo de ventana $M_{max}(k)$ en base a la covarianza relativa $c(\tau,k,k)$.

Puesto que ahora el orden en que son recuperados los vectores no afecta a las estimaciones (no se hace una reconstrucción progresiva), no es necesario hacer una reconstrucción de los extremos al centro, pudiéndonos limitar a una reconstrucción hacia delante (figura 5.5b). Así, una ventaja adicional asociada a esta estrategia es que conocemos a priori el retardo que introduce, dado por M_{max}.

5.4.4. Complejidad computacional y requerimientos de memoria

Dependiendo de la estrategia considerada, la complejidad computacional de la estimación es variable, aunque en líneas generales ésta es generalmente alta, debido principalmente a la operación de inversión de matrices implicada. Atendiendo a la expresión (5.45), la estimación MAP requiere la inversión de una matriz además de la multiplicación de una

matriz por la matriz invertida. Suponiendo matrices cuadradas de $n \times n$ elementos, se suele considerar que el orden de complejidad de cada una de estas operaciones viene dado por $O(n^3)$ (aunque aplicando técnicas más avanzadas como el algoritmo de Strassen es posible reducir este orden hasta $O(n^{\log_2 7})$ [150]).

En la reconstrucción progresiva, en donde se procede vector a vector, el vector de componentes perdidas, $\mathbf{X}_m$, consta de K componentes o características de voz, mientras que el vector de observaciones, $\mathbf{X}_o$, consta de como máximo $2MK$ elementos, siendo M la longitud de la ventana considerada. Esto implica que ha de multiplicarse una matriz de $K \times 2MK$ elementos por otra de $2MK \times 2MK$, que además debe ser invertida. Como resultado, el orden de complejidad del algoritmo viene dado por $O(4M^2K^3 + 8M^3K^3) \approx O(K^3M^3)$.

En la reconstrucción por bandas cepstrales se opera componente a componente. Durante la estimación de cada componente es necesario construir un vector de observaciones que consta de como máximo $2M$ elementos, siendo necesaria la multiplicación de una matriz de $1 \times 2M$ elementos por otra de $2M \times 2M$ elementos que, al igual que antes, debe invertirse. El orden de complejidad de la estimación de cada componente viene dado entonces por $O(4M^3 + 8M^3)$. Dado que un vector consta de K componentes, y la estimación ha de repetirse K veces, el orden se incrementa a $O(12KM^3) \approx O(KM^3)$. Como puede observarse, hay una importante reducción de la complejidad. A esta reducción hay que añadir la que se obtiene al no considerar como observaciones los vectores previamente estimados. Esto hace que, en general, la matriz de observaciones sea más pequeña que en la reconstrucción progresiva.

Los requerimientos de memoria dependen también de la estrategia considerada, aunque en general pueden considerarse bajos. La memoria requerida depende del número de covarianzas $c(\tau, k_1, k_2)$ necesarias para la estimación. En la reconstrucción progresiva son necesarias $2MK^2$ covarianzas, esto es, una para cada posible desplazamiento entre cada dos componentes de la ventana, exceptuando el cero ($M \leq \tau \leq M$,$\tau \neq 0$). En la reconstrucción por bandas este número se reduce a $2MK$. Puesto que durante la estimación de una característica sólo se consideran aquellas componentes que representen a la misma característica en instantes anteriores o posteriores, sólo son necesarias las covarianzas $c(\tau, k, k)$.

5.4.5. Resultados experimentales

Las técnicas propuestas basadas en la estimación MAP han sido evaluadas en un entorno experimental idéntico al propuesto en la sección 5.3.6. Por ello, aquí será descrito sólo de forma resumida. Se emplea una arquitectura DSR a fin de que la degradación debida a los vectores perdidos sea la única presente. El canal se simula mediante un modelo de Gilbert, proponiéndose 25 condiciones que ofrecen un muestreo con cierto detalle de la bondad de las técnicas bajo diferentes estados de congestión de la red. Los resultados obtenidos mediante la técnica de mitigación propuesta por el estándar, consistente en la repetición del vector más cercano, se emplearán como referencia. Para facilitar la lectura, estos resultados se han vuelto a reproducir en la tabla 5.8.

La tabla 5.9 muestra los resultados obtenidos empleando una reconstrucción MAP progresiva operando vector a vector. Como se detalló en la sección 5.4.2, esta técnica estima vectores completos a partir de vectores previa y posteriormente recibidos, así como vectores previamente estimados. Debido a su alta complejidad computacional, se ha definido una ventana reducida con $M = 5$ vectores. Como puede observarse, esta técnica mejora notablemente la precisión en el reconocimiento en comparación con la mitigación estándar. Sin embargo, requiere la inversión de matrices de 140×140 observaciones, resultando en una ingente cantidad de cálculo. Valga como detalle que, durante nuestras pruebas, el tiempo empleado para la mitigación superaba varias veces el requerido para el propio reconocimiento.

Como se mostró en la sección 5.4.3, una forma de reducir la matriz de observaciones, y por tanto, la complejidad computacional, consistía en descartar aquellas observaciones que no tuvieran una contribución significativa a la estimación. Para ello se hacía uso de la covarianza relativa entre componentes. Una aproximación ruda a este descarte de observaciones consistía en la definición de un umbral r_{min}, de forma que no se considerasen aquellas componentes cuya covarianza relativa con respecto a la componente estimada fuese inferior a él. La tabla 5.10 muestra la precisión del reconocimiento obtenida aplicando esta estrategia, con un umbral $r_{min} = 0{,}25$ y una ventana $M = 5$. Como puede observarse, esta aproximación da lugar a una ligera reducción general de la precisión en el reconocimiento, pero permite una reducción importante del número de observaciones consideradas y del tamaño de las matrices de autocovarianza.

Otra aproximación, derivada del análisis de los valores de la covarianza relativa, consistía en operar por bandas cepstrales. Por simplicidad, esta aproximación se extendía

Tasa de pérdidas	*Long. media ráfaga*				
	1	*2*	*4*	*8*	*12*
10 %	98.98	98.61	96.50	93.43	91.69
20 %	98.96	98.08	94.03	87.28	84.74
30 %	98.88	97.56	90.92	81.43	77.25
40 %	98.92	96.81	88.18	76.61	69.00
50 %	98.90	96.21	84.57	70.36	63.27

Tabla 5.8: Precisión del reconocimiento (Wacc) empleando la mitigación propuesta por el estándar.

Tasa de pérdidas	*Long. ráfaga*				
	1	*2*	*4*	*8*	*12*
10 %	99.05	98.82	97.40	94.70	93.05
20 %	99.04	98.44	95.75	90.14	87.48
30 %	98.99	98.17	93.57	85.69	80.83
40 %	99.04	97.79	91.54	81.64	74.46
50 %	98.96	97.34	88.69	76.48	69.24

Tabla 5.9: Precisión del reconocimiento (Wacc) empleando la reconstrucción MAP progresiva por vectores.

Tasa de pérdidas	*Long. ráfaga*				
	1	*2*	*4*	*8*	*12*
10 %	99.04	98.82	97.26	94.53	92.94
20 %	99.03	98.45	95.60	89.77	87.32
30 %	98.99	98.06	93.24	85.08	80.47
40 %	99.00	97.75	91.08	80.98	73.82
50 %	98.97	97.29	88.14	75.83	68.91

Tabla 5.10: Precisión del reconocimiento (Wacc) empleando la reconstrucción MAP progresiva con umbral de covarianza relativa.

Tasa de pérdidas	*Long. ráfaga*				
	1	*2*	*4*	*8*	*12*
10 %	99.04	98.73	97.23	94.75	93.46
20 %	99.01	98.31	95.54	90.22	88.02
30 %	99.00	97.96	93.30	85.94	81.41
40 %	99.01	97.54	91.26	81.89	75.30
50 %	98.97	97.04	88.04	76.78	70.19

Tabla 5.11: Precisión del reconocimiento (Wacc) empleando la reconstrucción MAP no progresiva por bandas cepstrales.

también a los parámetros energéticos. La tabla 5.11 muestra la precisión del reconocimiento obtenida empleando la reconstrucción MAP no progresiva por bandas cepstrales. En base a los valores de covarianza relativa obtenidos (en la figura 5.6 se mostraban algunos de ellos) se ha considerado una ventana de tamaño máximo $M_{max} = 10$. De esta forma la técnica tan sólo requiere la inversión de matrices de 20×20 elementos. Este es, además, un valor máximo que sólo se produce cuando se pierde un único vector y todos los adyacentes se han recibido. Por lo general, la matriz cuenta con menos elementos, ya que la reconstrucción se realiza de forma no progresiva (las componentes estimadas no se consideran como recibidas). Debido al descarte de observaciones, esta técnica conduce a una ligera reducción de la precisión frente a ráfagas cortas (longitudes medias de 2 y 4 paquetes) con respecto a la reconstrucción progresiva por vectores. Sin embargo, la capacidad para usar una ventana de mayor tamaño, unida al hecho de que las componentes previamente estimadas no se reutilizan (y no distorsionan la estimación actual), permite que, frente a ráfagas más largas (8 y 12 paquetes), se obtengan incluso mejores resultados que la estimación MAP progresiva con vectores completos.

5.5. Combinación MMSE-MAP

Como hemos visto en las secciones anteriores, las técnicas MMSE y MAP desarrolladas presentan requerimientos computacionales y de memoria contrapuestos. Por un lado, las estimaciones MMSE descritas son muy rápidas gracias a que la estimación de cada vector está precalculada y la mitigación se reduce a una búsqueda de la combinación de referencia en la tabla de registros. Como vimos, esta búsqueda además puede optimizarse ordenando las tablas y empleando un algoritmo de búsqueda binaria cuya complejidad es aún menor. Sin embargo, se requieren grandes cantidades de memoria para almacenar las combinaciones de símbolos junto a las estimaciones correspondientes. Comparativamente, los requerimientos de memoria de la estimación MAP son mucho más bajos. Sin embargo, su complejidad computacional es elevada, ya que se hace necesaria la inversión y multiplicación de matrices de dimensiones significativas.

La existencia de estos requerimientos contrapuestos facilita el diseño de un algoritmo de mitigación basado en estimación MMSE y MAP que suponga un compromiso entre complejidad computacional y requerimientos de memoria.

5.5.1. Descripción de la técnica

La técnica propuesta se basa en una modificación de la reconstrucción MMSE basada en modelo reducido. Como vimos en la sección 5.3.4, esta reconstrucción sólo considera aquellas combinaciones cuya frecuencia de aparición en la base de datos de entrenamiento supera un cierto umbral η. Por esta razón es posible que, durante su aplicación, pueda darse una combinación de símbolos de la que no se tenga una estimación precalculada. Era necesario entonces emplear un algoritmo de mitigación secundario, por ejemplo, la repetición del vector más cercano. Esto nos aseguraba que los resultados serían como mínimo los obtenidos por esta técnica.

Una idea alternativa consiste en aplicar la estimación MAP como técnica de mitigación secundaria en una reconstrucción MMSE basada en modelo reducido. En vista de los resultados experimentales obtenidos con las técnicas MAP preferiremos la reconstrucción por bandas cepstrales. Esto se debe a que ofrece resultados similares al resto de técnicas (incluso superiores), con unos menores requerimientos de memoria y complejidad computacional. Aunque la integración de estos dos algoritmos de mitigación resulta trivial, conduce a las siguientes ventajas:

- Se garantiza que en el peor caso se obtienen los resultados de la estimación MAP, mejores a los de la mitigación estándar.
- Gracias a esto, es posible incrementar el umbral de aceptación. Un umbral más alto implica considerar un menor número de registros (la combinación de símbolos debe aparecer con una mayor frecuencia en la base de datos), conduciendo por tanto a una reducción del número de estimaciones precalculadas. Adicionalmente, es de esperar una mejora en la calidad de estas estimaciones, ya que se garantiza, para todas ellas, un mayor número de muestras sobre las que obtenerse (el registro tiene una frecuencia de aparición más alta).
- Puesto que aquellas combinaciones de símbolos que no tienen registro correspondiente son precisamente las que menos aparecen durante el entrenamiento, es de esperar que igualmente tengan una baja frecuencia de aparición durante las pruebas. Así, la mitigación secundaria, en este caso la estimación MAP, debe aplicarse relativamente poco.

Como técnica resultante, se obtiene una nueva solución que requiere de menos memoria que las técnicas MMSE y menos cálculo que la estimación MAP.

La transferencia de requerimientos (memoria y complejidad computacional) depende del valor η elegido. En principio este valor debería ser más alto que el empleado en la estimación MMSE no combinada. Sin embargo, un umbral excesivamente elevado puede conducir a una aplicación continuada de la estimación MAP, al no disponerse de estimaciones para la mayoría de combinaciones de símbolos. En esta situación se perderían ambas ventajas, es decir, la técnica resultaría computacionalmente costosa y requeriría de una cantidad de memoria elevada. Para la elección de un umbral correcto podemos apoyarnos en la base de datos de entrenamiento. Establecido un valor para η, podemos obtener una probabilidad de rechazo $P(rechazo|\eta)$ como,

$$P(rechazo|\eta) = \frac{n_{dwn}}{n_{up} + n_{dwn}} \tag{5.58}$$

en donde n_{dwn} es el número de apariciones que han sido rechazadas como registros y n_{up} el número de apariciones de los registros, esto es,

$$n_{up} = \sum_{r \in C} \#r \ , \ n_{dwn} = \sum_{r \notin C} \#r \tag{5.59}$$

donde $\#r$ denota el número de veces que aparece una combinación de símbolos r en la base de datos y C es el conjunto de registros o combinaciones que superan el umbral ($\#r > \eta$). Como puede observarse, $P(rechazo|\eta)$ expresa la probabilidad de que una combinación de símbolos no sea un registro, es decir, ésta no disponga de una secuencia de estimaciones precalculada. Esta probabilidad de rechazo puede obtenerse al mismo tiempo que se extraen los registros en la base de datos de entrenamiento y su valor estima el porcentaje de veces que se debe recurrir a la mitigación secundaria, en nuestro caso, la estimación MAP.

5.5.2. Resultados experimentales

La técnica propuesta se ha evaluado en el marco experimental descrito en la sección 5.3.6. La reconstrucción MMSE se realizará mediante un modelo de tercer orden, para el que se han precalculado secuencias de estimaciones hacia delante y atrás de hasta 20 instantes de tiempo para cada una ($L = 20$). Como técnica secundaria se empleará la estimación MAP no progresiva por bandas, definiendo un tamaño máximo de ventana $M_{max} = 10$.

Para establecer un umbral de aceptación adecuado se ha obtenido el número de registros resultante y la probabilidad de rechazo de algunos umbrales, mostrados en la tabla

Umbral (η)	*Nº registros*	$P(rechazo\|\eta)$
10	132498	0.13
50	35123	0.32
100	17455	0.36
500	2137	0.75

Tabla 5.12: Número de registros almacenados y probabilidad de rechazo para distintos valores de η.

Tasa de pérdidas	*Long. ráfaga*				
	1	*2*	*4*	*8*	*12*
10 %	99.04	98.76	97.32	94.88	93.43
20 %	99.01	98.34	95.58	90.32	88.08
30 %	98.95	97.93	93.48	85.90	81.73
40 %	98.94	97.51	91.26	81.97	75.28
50 %	98.92	97.08	88.38	77.24	70.57

Tabla 5.13: Precisión del reconocimiento (Wacc) empleando la técnica combinada MMSE-MAP.

5.12. Como puede apreciarse, un umbral de 50 o 100 repeticiones mantiene la probabilidad de rechazo en torno a un tercio. Por tanto, es de esperar que con estos umbrales la técnica resulte en una complejidad 3 veces inferior a la estimación MAP (sin combinar). Resulta evidente la conveniencia del umbral $\eta = 100$, ya que el número de registros requeridos se reduce casi a una décima parte con respecto a la reconstrucción MMSE basada en modelo reducido de tercer orden con $\eta = 10$. Aunque umbrales más estrictos, como $\eta = 500$, permiten una mayor reducción en el número de registros, implican una alta aplicación de la estimación MAP, siendo sólo aconsejables cuando los requerimientos de memoria sean muy estrictos.

La tabla 5.13 muestra los resultados obtenidos mediante esta técnica combinada con $\eta = 100$. Como puede observarse, la técnica propuesta conduce a un aumento de la precisión del reconocimiento con respecto a una reconstrucción basada en modelo reducido de tercer orden con $\eta = 10$ (tabla 5.5), y con respecto a una reconstrucción MAP por bandas no progresiva (tabla 5.11). Sin embargo, el hecho más significativo es la reducción de memoria y cómputo necesario. Frente a los más de cien mil registros necesarios en un modelo reducido de tercer orden con $\eta = 10$, ahora tan solo se requieren 17455 registros. Por otra parte, se espera que la complejidad computacional con respecto a la estimación MAP se reduzca a un 36 %.

Técnica de mitigación	*Condición*				
	1	*2*	*3*	*4*	*5*
Mitigación estándar	98.98	98.08	90.92	76.61	63.27
FB–MMSE	99.01	98.01	90.43	75.38	61.35
F+B–MMSE	99.00	98.17	91.91	77.89	64.23
MMSE orden 2	99.01	98.23	92.49	79.76	67.38
MMSE orden 3 reducido	99.03	98.25	93.42	81.58	69.75
MMSE orden 4 reducido	98.97	98.21	92.85	80.16	68.07
MAP progresivo	**99.05**	98.44	**93.57**	81.64	69.24
MAP con umbral	99.04	**98.45**	93.24	80.98	68.91
MAP por bandas	99.04	98.31	93.30	81.89	70.19
Combinación MMSE-MAP	99.04	98.34	93.48	**81.97**	**70.57**

Tabla 5.14: Resumen de los resultados obtenidos con las técnicas de mitigación propuestas.

5.6. Resumen de resultados y conclusiones

En este capítulo hemos desarrollado diferentes técnicas de mitigación destinadas a combatir, en el receptor, los efectos de la pérdida de vectores de características. Como se mostró en el capítulo 3, la pérdida de paquetes así como el descarte de tramas da origen a una pérdida de información que tiene un efecto negativo relevante sobre el reconocimiento de la voz. Se hacen por tanto necesarias técnicas que provean de reemplazos para los vectores perdidos. Las técnicas propuestas en este capítulo estiman dichos reemplazos por medio de un modelo estadístico de voz. Dos métodos de estimación bien conocidos, la estimación MMSE y la estimación MAP, han servido de base para su desarrollo.

La tabla 5.14 muestra un resumen de los resultados obtenidos mediante las técnicas propuestas. Esta tabla se compone únicamente de los resultados de la diagonal de las tablas anteriores. Es decir, la condición 1 representa $R_{loss} = 10\,\%, L_{loss} = 1$, la condición 2, $R_{loss} = 20\,\%, L_{loss} = 2$, y así sucesivamente hasta la condición 5 con $R_{loss} = 50\,\%, L_{loss} = 12$. Los resultados obtenidos con la mitigación propuesta por el estándar tambien se incluyen en dicha tabla como punto de referencia.

Las técnicas basadas en estimación MMSE (tabla 5.14, filas 2-6) se han desarrollado a partir de la técnica de mitigación FB–MMSE. Esta técnica ofrece excelentes resultados en canales inalámbricos gracias al eficiente uso que realiza de la información obtenida del modelado de la voz y del modelado de la transmisión. Sin embargo, su aplicación a los canales con pérdidas, donde no puede establecerse de forma explicita un modelo de canal, ofrece un rendimiento inferior a la mitigación estándar. Una variante de esta técnica, la

estimación F+B–MMSE (tabla 5.14, fila 3), que además tiene una menor complejidad computacional, ponía de manifiesto que es posible obtener buenas estimaciones a partir de la información contenida únicamente en la fuente, si se evitaba lo que denominábamos degradación mutua en los extremos (presente en la estimación FB–MMSE).

Gracias a las simplificaciones que conlleva la ausencia de un modelo explicito de canal, es posible desarrollar técnicas de estimación MMSE con modelos más ricos de voz. Se proponen así la técnica MMSE basada en modelos de segundo orden (tabla 5.14, fila 4). Esta técnica hace uso de los dos vectores recibidos anterior y posteriormente a la ráfaga, obteniendo mejores estimaciones y, como puede observarse, mejor precisión en el reconocimiento. Al no disponer de información instantánea esta técnica almacena las estimaciones precalculadas. De esta forma, se obtiene una mitigación de poca complejidad computacional pero altos requerimientos de memoria. Puesto que estos requerimientos crecen de forma exponencial al considerar modelos de órdenes mayores, se propone la técnica MMSE basada modelo reducido de orden M (tabla 5.14, filas 5 y 6). Esta aproximación permite reducir de forma drástica el número de estimaciones precalculadas, haciendo factible el uso de estos modelos de orden superior. Sin embargo, debíamos ser cuidadosos a la hora de elegir el orden del modelo. Como puede observarse en la tabla, la elección de un orden demasiado alto puede provocar una insuficiencia de datos durante el entrenamiento de las estimaciones, conduciendo a peores resultados (tabla 5.14, fila 6). La obtención de un número de registros inferior al obtenido con un modelo reducido de orden inferior era indicativo de este problema.

Las técnicas MAP propuestas (tabla 5.14, filas 7-9) ofrecen una aproximación alternativa. Estas técnicas consideran la secuencia de vectores de características como la salida de un proceso aleatorio estacionario en sentido amplio con una distribución gaussiana. Partiendo de esta hipótesis, los vectores estimados se calculan maximizando su probabilidad condicionada a los valores de los vectores recibidos, también llamados vectores observados. El problema de estas técnicas radica en su elevada complejidad computacional, al requerirse la inversión de grandes matrices de observaciones. Por esta razón, se proponen soluciones que reduzcan el número de observaciones que se tienen en cuenta durante la estimaciones (tabla 5.14, fila 8). Un análisis de las covarianzas relativas entre las componentes de los vectores revela que, debido a la operación DCT aplicada, los coeficientes cepstrales presentan bajas covarianzas relativas entre ellos. Se propone así una estimación MAP realizada por bandas que permite una reducción significativa de la complejidad sin provocar una reducción importante en la precisión del reconocimiento (tabla 5.14, fila 9).

Ambas aproximaciones presentan mejoras importantes en el reconocimiento, aunque cabe destacar que, en general, se obtienen mejores resultados mediante las técnicas MAP. Aunque esto resulte en principio contradictorio, se justifica en que las técnicas MAP propuestas disponen de más información para realizar la estimación. Así, mientras que la estimación MMSE llega a considerar hasta tres vectores previos o posteriores a la ráfaga, la estimación MAP emplea una ventana en la que se pueden llegar a considerar 10 vectores, si estos llegan a recibirse.

Por último, puede observarse que ambas aproximaciones, MMSE y MAP, presentan requerimientos contrapuestos. Por ello, se ha presentando una solución que integra ambos métodos de estimación y cuyo objetivo es el de equilibrar la complejidad computacional y los requerimientos de memoria. Aunque esta combinación se puede llevar a la práctica de una forma casi trivial, permite la obtención de una solución que ofrece mejores resultados que ambas técnicas por separado (tabla 5.14, fila 10) a la vez que una menor complejidad con respecto a las técnicas MAP y con menos requerimientos de memoria que las técnicas MMSE.

Capítulo 6

TÉCNICAS BASADAS EN EL EMISOR Y TRATAMIENTO DE PÉRDIDAS EN EL RECONOCEDOR

6.1. Introducción

Las técnicas de mitigación tienen la ventaja de requerir únicamente la colaboración del receptor, pero con frecuencia están limitadas. Esto se debe a que implícitamente asumen que el segmento de señal a recuperar presenta una evolución estable, suposición que sólo es válida para lapsos cortos de tiempo. El rendimiento del sistema puede mejorarse si el emisor colabora anticipándose a los errores. Así, surgen técnicas preventivas cuyo propósito consiste en evitar que se produzcan las pérdidas, o al menos que éstas queden distribuidas de una forma favorable. A este conjunto de técnicas de recuperación se le conoce como técnicas basadas en emisor.

La recuperación basada en emisor no se realiza de forma aislada, ya que estas técnicas difícilmente llegan a evitar todas las pérdidas. Por esta razón, las técnicas basadas en el emisor se aplican generalmente junto a técnicas de mitigación, considerándose ambos paradigmas complementarios. De hecho, existen técnicas basadas en el emisor que sólo tratan de mejorar el rendimiento final de las técnicas de mitigación. El entrelazado de vectores es el ejemplo más evidente de esto. Esta técnica no permite la recuperación de ningún vector perdido, pero modifica la distribución de las pérdidas de forma que mejore

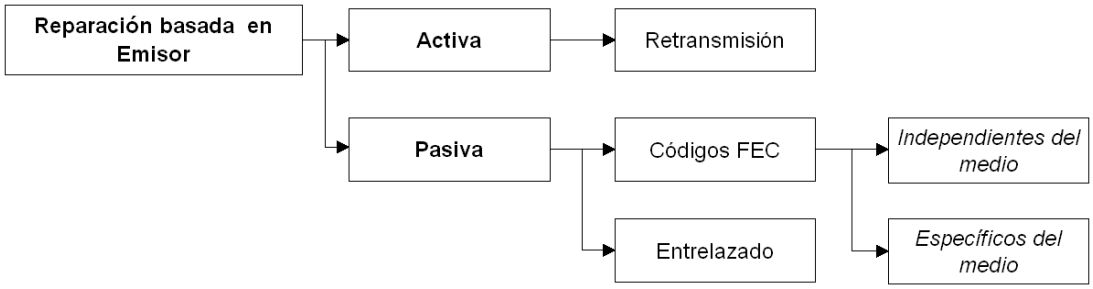


Figura 6.1: Técnicas de reparación basadas en el emisor.

el rendimiento de las técnicas aplicadas en el receptor.

A pesar de sus beneficios, la introducción de estas técnicas queda limitada a aquellas redes y arquitecturas en donde sea factible modificar el emisor, puesto que es necesaria su colaboración. Esto supone un problema para las arquitecturas NSR, ya que la principal ventaja de esta aproximación al RSR consiste, precisamente, en que no es necesaria la modificación del terminal del usuario. Esto no quiere decir que las técnicas basadas en emisor no estén presentes en estas arquitecturas. En GSM, por ejemplo, se utilizan profusamente, debido a la hostilidad del canal de radio. Sin embargo, éstas están basadas en estándares de codificación de voz y canal ya establecidos, diseñados para maximizar la calidad de voz, no el rendimiento del reconocimiento de habla. Así pues, el problema radica en que, a diferencia de las técnicas de mitigación, que pueden sustituirse por técnicas más avanzadas sin requerir cambios mayores, las técnicas basadas en el emisor pueden implicar cambios en todo el sistema (especialmente en el emisor). Ciertas redes, en cambio, como las redes IP, no presentan ningún inconveniente a la hora de introducir este tipo de técnicas, gracias a la flexibilidad que les caracteriza.

Las técnicas de reparación basadas en emisor pueden clasificarse en técnicas activas y técnicas pasivas. La figura 6.1 muestra un diagrama jerárquico de todas ellas. Las primeras se basan en esquemas de retransmisión activa de las pérdidas, esto es, en el reenvío de los paquetes si tras un cierto tiempo no se ha recibido confirmación de su llegada. En transmisiones sin requerimientos temporales estás técnicas resultan muy útiles. Mediante esta técnica el protocolo TCP, por ejemplo, asegura que todos los paquetes son entregados al destinatario. Sin embargo, debido a los estrechos márgenes de latencia disponibles en las aplicaciones interactivas, como es el caso del reconocimiento remoto, las técnicas de retransmisión no resultan apropiadas.

Generalmente, un esquema de retransmisión eficiente implica un retraso excesivo para los paquetes reenviados. Esto se debe a que las pérdidas de paquetes se deben a sobre-

cargas en las colas de los routers. Supuesta una cola saturada, el reenvío constante de paquetes no ayudará a la recuperación de esta sobrecarga sino que, justamente al contrario, prolongará esta situación. Se hace necesario entonces una espera que permita que la cola se descongestione, permitiendo que el paquete no sea descartado sino transmitido de forma efectiva. El retraso requerido usualmente inutiliza los paquetes retransmitidos que, al llegar fuera de los requerimientos temporales establecidos, son equivalentes a una pérdida. Esto podría no resultar problemático sino fuera porque el ancho de banda empleado para la retransmisión podría haberse utilizado para el envío de otro paquete más reciente.

Por esta razón, los esquemas de retransmisión suelen descartarse en las comunicaciones con requerimientos de tiempo real. En su lugar, generalmente se aplican técnicas basadas en la codificación pasiva del canal, que agrupan al entrelazado y la transmisión de códigos de corrección hacia delante. En este capítulo exploraremos estas técnicas y su utilidad en el reconocimiento remoto de la voz, en concreto para las arquitecturas DSR (por las razones expuestas anteriormente). Las dos secciones siguientes están dedicadas a la aplicación de códigos de corrección y al uso de entrelazadores, proponiéndose distintos esquemas que permiten una mejora significativa de la precisión en el reconocimiento.

En este capítulo también exploraremos el tratamiento de las pérdidas en el propio reconocedor, al que se dedican las secciones restantes, especialmente cuando éste se combina con técnicas basadas en el emisor. Como se mencionó al principio del capítulo anterior, en los sistemas de reconocimiento remoto del habla el propio reconocedor de voz puede adaptarse a la presencia de vectores perdidos. Mediante modificaciones en los algoritmos de reconocimiento, es posible realizar una decodificación de las observaciones en texto escrito aún cuando algunas de ellas no sean fiables o estén incompletas. Aunque estas técnicas pertenecen a un paradigma mucho más amplio, cuya utilidad va más allá de su aplicación en los canales con pérdidas, en tales canales pueden considerarse como técnicas de recuperación basadas en el reconocedor. Como las técnicas de mitigación, estas técnicas se aplican en el *back-end* pero, a diferencia de éstas, no obtienen sustituciones para los vectores perdidos, sino que adaptan el reconocimiento a su presencia. En este sentido, son dependientes de la aproximación al RAH elegida.

6.2. Códigos de Corrección hacia Delante

La cantidad de información no disponible para el reconocimiento, debida a la pérdida de vectores causada por el canal, puede reducirse o minimizarse por medio de la introduc-

ción de información redundante en los paquetes transmitidos. Esta información adicional permite al receptor realizar una reconstrucción parcial o incluso total, en el mejor de los casos, de los vectores perdidos. A estos datos redundantes se les conoce como códigos de corrección hacia delante (*FEC–Forward Error Correction*) y su objetivo no es otro que el de anticiparse a los efectos degradantes del canal.

Como contrapartida, la introducción de estos datos redundantes implica la necesidad de un ancho de banda mayor. Esto supone un compromiso, ya que una mayor redundancia de información permitirá recuperar un mayor número de vectores perdidos, pero conlleva un incremento de la tasa de bits transmitidos. Este incremento, además de resultar indeseable, puede contribuir a la congestión de la red, resultando en un empeoramiento del canal. Esto se debe a que al incrementar el tamaño de los paquetes, se hace más probable la saturación de las colas de los routers (véase la sección 3.7.1) dando lugar a más pérdidas. Por ello, se persigue la minimización de la cantidad de datos redundantes por paquete. Adicionalmente puede ligarse la aplicación de las técnicas FECs al uso de protocolos de control de la congestión. De esta forma, puede retardarse el envío de la información redundante para cuando se tenga alguna constancia de pérdidas en el receptor.

6.2.1. FEC independientes y FEC específicos del medio

Los códigos FEC pueden clasificarse, según sean o no independientes de la información que se transmite, en FECs independientes del medio o FECs específicos de éste. Ambos tipos de códigos presentan ventajas e inconvenientes. Sin embargo, como veremos, los FECs específicos del medio pueden resultar especialmente útiles para el reconocimiento distribuido de la voz.

FEC independientes del medio

Los FEC independientes del medio emplean códigos mediante los cuales se generan paquetes adicionales que permiten la reconstrucción de paquetes perdidos. Esto es, cada código FEC se construye a partir de n paquetes de datos, generando k paquetes adicionales, así, un total de $n + k$ paquetes son transmitidos por la red. Estos paquetes de reparación son independientes del contenidos de los paquetes, es decir, no se realiza ninguna suposición acerca de la información que se transmite. Gracias a ello, la reconstrucción del paquete perdido es una copia exacta del paquete original. Debido a esta propiedad ha existido un importante interés en el desarrollo de FEC independientes.

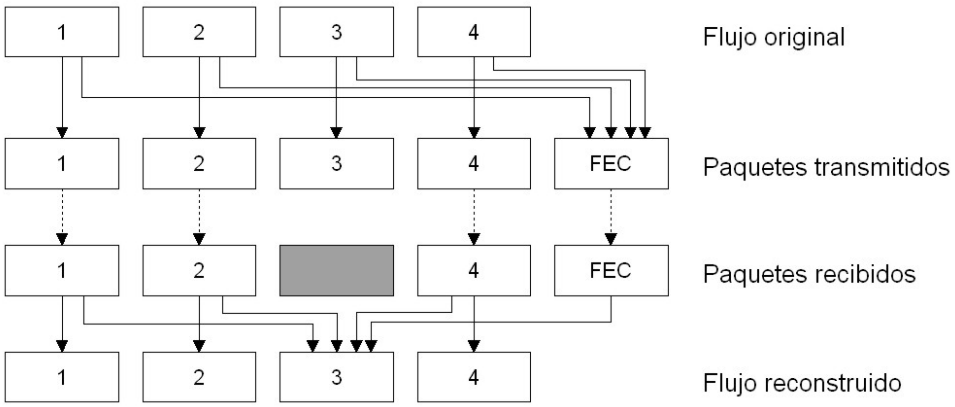


Figura 6.2: Esquema FEC independiente del medio incluyendo un paquete redundante cada cuatro obtenido mediante la operación XOR.

Existen diversos esquemas de codificación FEC independientes propuestos para canales basados en conmutación de paquetes, destacando la codificación por paridad y los códigos Reed-Solomon, para los que existen definiciones de carga RTP [151]. En la codificación por paridad se aplica la operación o-exclusivo (XOR) sobre un grupo de paquetes para generar el correspondiente paquete de paridad. Un esquema común consiste en la transmisión de un paquete de paridad tras n paquetes de datos (figura 6.2). De esta forma es posible recuperar un paquete perdido cada $n+1$ paquetes enviados. Mediante diferentes combinaciones de la operación XOR sobre los paquetes, pueden derivarse otras técnicas de codificación. Desafortunadamente, estos esquemas resultan completamente inútiles frente a pérdidas en forma de ráfaga, donde varios paquetes del bloque (o incluso todo el bloque) pueden perderse. Los códigos Reed-Solomon (R-S) [152], en cambio, son conocidos por sus excelentes propiedades de corrección en presencia de ráfagas. Aunque la complejidad de estos códigos es muy superior a la codificación XOR, lo cierto es que el procedimiento de codificación es relativamente eficiente y existen algoritmos optimizados para realizarlo. En ausencia de errores, el proceso de decodificación tiene el mismo coste computacional que la codificación, incrementándose en presencia de paquetes perdidos. Así, los códigos R-S han sido aplicados con anterioridad al problema del reconocimiento remoto sobre redes de paquetes [50, 153]. En estas soluciones se descarta el uso de la codificación descrita en los estándares DSR en favor de una protección contra errores desigual (*UEP–Unequal Error Protection*) propuesta por los autores, donde los bits de los parámetros de voz son ordenados según su importancia para el reconocimiento (medida a través del WER) y alojados en distintos flujos de códigos RS.

Aparte del aumento de complejidad computacional que suponen, el principal inconveniente para la aplicación de este tipo de códigos FEC a los canales con pérdidas radica en su escasa eficacia bajo ciertas condiciones transitorias con elevadas tasas de pérdidas. Esto se debe a que los códigos FEC independientes del medio requieren que al menos n de los $n + k$ paquetes por bloque sean recibidos.

FEC específicos del medio

A diferencia de los anteriores, los FEC específicos del medio conocen el tipo de medio que se transmite y los esquemas de compresión disponibles para éste. La base de estos esquemas radica en la repetición de la información de cada paquete en otro u otros paquetes. De esta forma, si un paquete se pierde, otro paquete conteniendo esta misma información puede cubrirlo. Evidentemente, si la información fuera replicada tal cual, el ancho de banda sería varias veces el original (dependiendo de cuantas veces se repita la información). Se hace pues necesaria una criba de los datos adicionales que conserve la información más relevante, desechando la que es accesoria. En general, se suele recurrir a una codificación secundaria cuya tasa de bits es significativamente inferior a la de la codificación principal. De ahí que se les considere específicos del medio, ya que la codificación secundaria no se puede emplear sobre un medio distinto a aquel para el que se ha diseñado. Sin embargo, la característica más relevante de los FEC específicos es que no obtienen una paquete reparado idéntico al paquete perdido, sino una versión degradada a la que nos referiremos como *réplica*. La figura 6.3 muestra un ejemplo empleando este esquema. Como puede observarse, los FEC específicos del medio presentan analogías con la codificación adaptativa de AMR. Frente a la pérdida total de un vector de parámetros es preferible la obtención de una réplica, aunque degradada, del vector original.

Como los FEC independientes del medio, estos FECs no aseguran la recuperación de todos los paquetes. Si un paquete y sus réplicas se pierden, los vectores que contuvieran están irremediablemente perdidos. Sin embargo, los FEC específicos del medio presentan, frente a los anteriores, la ventaja de no imponer un número mínimo de paquetes recibidos por bloque para llevar a cabo la reconstrucción. Cuando se recibe un paquete, la información adicional que contiene puede emplearse para la recuperación de otro paquete (en caso de que este último se haya perdido). En este sentido, los FEC específicos pueden resultar especialmente atractivos.

Supongamos un escenario en el que los paquetes contienen réplicas de otros vectores relativamente alejados en el tiempo. Cuando estos paquetes se reciban, la información

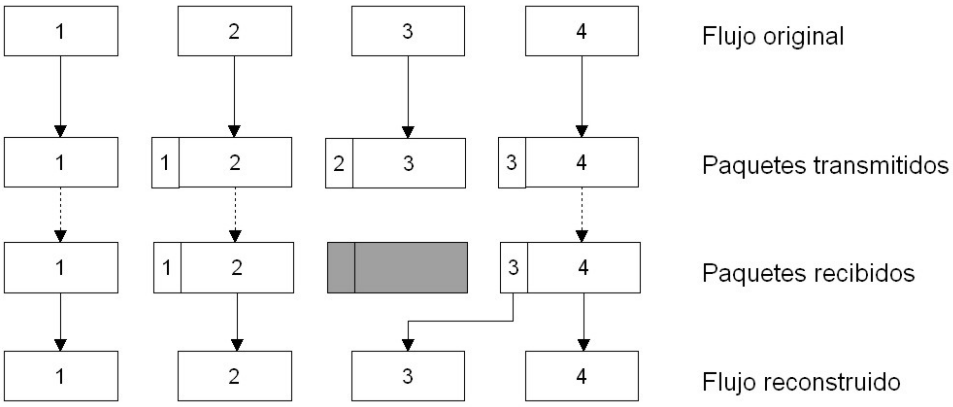


Figura 6.3: Esquema FEC específico del medio en el que cada paquete incluye una réplica del anterior.

que contienen no sólo permite una reconstrucción parcial de un vector perdido, sino que además la recuperación de este vector puede lograr la fragmentación de una ráfaga en dos ráfagas más cortas. En la figura 6.4 se muestra un posible caso.

La fragmentación de ráfagas en otras más cortas da lugar a una mejora en la precisión del reconocimiento. Esto se debe a que las técnicas de mitigación, que implícitamente asumen que el segmento a recuperar se encuentra en un estado estacionario, funcionan mejor frente a ráfagas cortas que frente a ráfagas largas [154]. Experimentalmente se obtiene que, para la misma tasa de pérdidas, las condiciones con ráfagas más largas resultan más degradantes que aquellas con ráfagas más cortas (véase sección 3.7.2). Así, como puede observarse en la figura 3.20, una reducción en la longitud promedio de la ráfaga (equivalente a un desplazamiento longitudinal) conduce a una mejora significativa en el reconocimiento.

Sin embargo, debemos recordar que los FECs específicos no obtienen reemplazos exactos de los vectores perdidos. Las réplicas no son más que copias degradadas y no deberían considerarse de la misma forma que un vector recibido. En este sentido, el éxito del esquema FEC depende de cómo explote el receptor la información parcial contenida en la réplica. De nuevo, las técnicas aplicadas en el receptor juegan un papel relevante.

6.2.2. Reparación de paquetes perdidos mediante réplicas VQ

Basándonos en la hipótesis descrita anteriormente, por la que el uso de réplicas permitiría una fragmentación de las ráfagas en otras más cortas y, por tanto, una mejora del rendimiento, en esta sección proponemos un sencillo esquema FEC específico del medio,

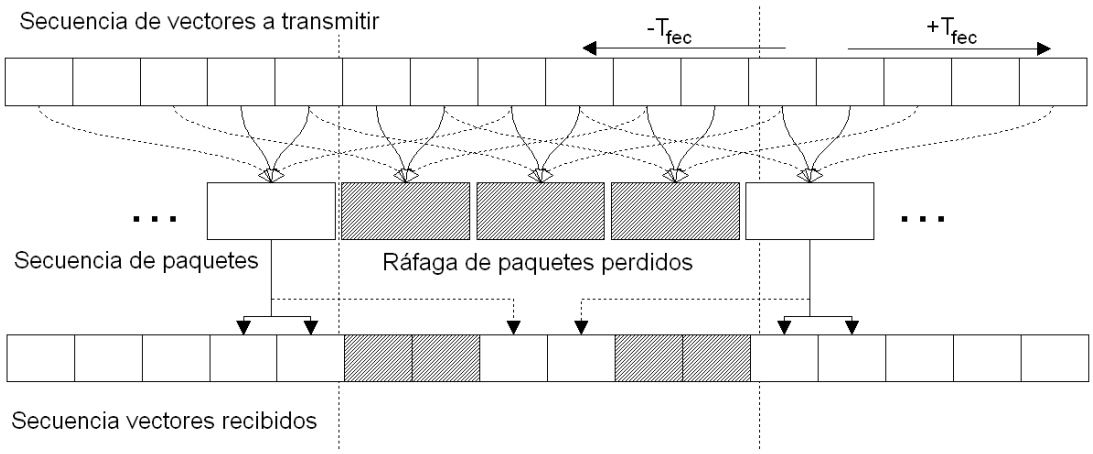


Figura 6.4: Ejemplo de esquema FEC especifico del medio incluyendo vectores a una distancia $T_{fec} = \pm 3$ como vectores redundantes para cada paquete.

en el que en cada paquete se codifican 4 vectores o tramas. De acuerdo con el estándar de DSR [74, 79], dos de estas tramas representarán la pareja actual de vectores que queremos transmitir. Mientras, las otras dos tramas representarán dos vectores redundantes, uno anterior y otro posterior a las tramas actuales, tomados a una distancia temporal T_{fec}. La figura 6.4 muestra un ejemplo de esta codificación redundante. Como puede observarse, cada vector es transmitido dos veces, una vez en el paquete que le corresponde y otra repetido en otro paquete a una distancia de T_{fec} vectores.

De no emplearse una codificación secundaria para la información redundante, el ancho de banda requerido se incrementará justamente al doble. Esta duplicación de la tasa de bits no sólo resulta inadmisible sino que además, al sobrecargar más la red, es de esperar que tenga efectos negativos sobre el estado del canal. Para evitar este crecimiento desproporcionado del ancho de banda, proponemos el uso de una sencilla codificación secundaria en la que los vectores redundantes se representan mediante cuantización vectorial (VQ) con un diccionario de 2^N centros (N bits). Este diccionario puede obtenerse empleando un algoritmo de k-medias sobre la base de datos de entrenamiento, considerando la siguiente medida de distancia pesada:

$$d_W(\mathbf{x}_r, \mathbf{x}_s) = \frac{\sum_{k=1}^{12} \left(c_r(k) - c_s(k)\right)^2}{\bar{\sigma}_c^2} + \frac{\left(c_r(0) - c_s(0)\right)^2}{\sigma_{c_0}^2} + \frac{\left(\log E_r - \log E_s\right)^2}{\sigma_{\log E}^2} \qquad (6.1)$$

en donde $x(c(0), \ldots, c(12), \log E)$ representa el vector de características con 14 dimensiones, $\bar{\sigma}_c^2$ es el promedio de las varianzas de los MFCCs(1-12), y $\sigma_{c_0}^2$ y $\sigma_{\log E}^2$ son las varianzas de $c(0)$ y $\log E$, respectivamente. Esta medida de distancia preserva el uso de la distancia

euclídea para los MFCC de orden 1 al 12 a la vez que ecualiza, en promedio, su contribución con respecto a las características energéticas por medio de la normalización con $\bar{\sigma}_c^2$, $\sigma_{c_0}^2$ y $\sigma_{\log E}^2$ [155].

A fin de asegurar cierta compatibilidad con el estándar de definición de carga RTP para DSR (más adelante retomaremos esta cuestión), supondremos que los datos no redundantes se codifican de acuerdo con dicho estándar. Es decir, siete diccionarios SVQ se emplean para codificar cada pareja de características. Los MFCCs(1-12) se codifican por medio de seis diccionarios de 64 centros (6 bits), mientras que para el MFFC0 y la $\log E$ se utiliza un diccionario de 256 centros (8 bits). De esta forma cada paquete incluiría la siguiente información:

- 2×44 bits que representan los dos vectores cuantizados mediante SVQ.
- $2 \times N$ bits representando las dos réplicas cuantizadas mediante VQ.

Gracias a este esquema FEC específico del medio, los paquetes recibidos antes y posteriormente a una ráfaga consiguen insertar réplicas dentro de ésta (figura 6.4). Puede observarse que, conforme el valor de T_{fec} es mayor, un mayor número de paquetes es capaz de insertar réplicas. Estas réplicas pueden usarse tal cual, de forma que si un vector no está disponible en el receptor pero sí su réplica, entonces ésta sustituirá al vector perdido.

Por lo general, no será posible una recuperación completa de la ráfaga. Por ello se recurre a sencillo algoritmo de mitigación basado en la reconstrucción propuesta por el estándar de Aurora. Como en este último, la ráfaga se divide en dos segmentos, de forma que el último vector recibido antes de la ráfaga se repite hacia delante, mientras que el primero tras ella se repite hacia atrás. Sin embargo, en la reconstrucción propuesta el vector a repetir se actualiza cada vez que se encuentra una réplica, siendo el vector codificado en la réplica el que se repite hacia adelante o hacia atrás, según corresponda.

Finalmente, debe considerarse el retraso introducido en la comunicación al utilizarse este esquema de códigos FECs. Esto se debe a que, para la construcción del paquete actual, es necesario esperar el vector generado T_{fec} instantes de tiempo posteriores a el. Es decir, el paquete no podrá enviarse hasta pasados T_{fec} instantes de tiempo. Igualmente, en el receptor no se puede dar por perdido un paquete que no ha sido recibido en el instante de tiempo actual, ya que pasados T_{fec} instantes podría recibirse su réplica. Como puede observarse, la latencia introducida depende del valor de T_{fec} y viene dada por $l = 2T_{fec}$.

6.2.3. Estimación FB-MMSE con réplicas VQ

En la sección anterior hemos mostrado cómo las réplicas pueden usarse de forma directa en el receptor. Sin embargo, estos códigos específicos no son más que versiones degradadas de los vectores, y buena parte del éxito del esquema dependerá de cómo el algoritmo de mitigación explote su presencia. En esta sección proponemos el uso de una técnica especialmente adaptada para tratar la información contenida en las réplicas, la técnica FB–MMSE con réplicas VQ.

Como ya se comentó en la sección 5.3.2, a pesar de resultar muy potente en el contexto de canales inalámbricos, la técnica FB–MMSE no ofrecía tan excelente rendimiento en canales con pérdidas. Esto se debía a que en dichos canales no se recibe ninguna información durante una ráfaga, dando lugar a una degradación mutua de las probabilidades de transición hacia los extremos. Este escenario cambia con la introducción de réplicas VQ, ya que éstas pueden ofrecer cierta información sobre algunos vectores perdidos. Ahora, podemos suponer que el canal con pérdidas se comporta de forma similar a un canal inalámbrico, en el sentido de que, durante las ráfagas, se *"reciben"* vectores degradados. Sería posible entonces aplicar la estimación FB–MMSE en su forma completa, esto es, considerando de forma explícita un modelo de canal.

Este modelo de canal debe construirse teniendo en cuenta la naturaleza de la degradación presente en las tramas recibidas artificialmente durante una ráfaga. Cuando se emplean réplicas VQ, la degradación no viene dada por una distorsión propia del canal (por ejemplo, la modificación de bits transmitidos), sino por el fuerte proceso de cuantización aplicado durante la codificación de la réplica. El modelo de voz, en cambio, sigue siendo el mismo y las expresiones descritas en la sección 5.3.1 (ecuaciones (5.10),(5.10, (5.11), (5.16) y (5.19)) pueden reutilizarse. Resta entonces por determinar las probabilidades de observación ($b_i(\mathbf{y}_t) = P(\mathbf{y}_t|\mathbf{x}^{(i)})$) que modelan la degradación (artificial) del canal.

Siguiendo un esquema similar al descrito originalmente para la técnica FB–MMSE en canales inalámbricos [142, 144, 145], operaremos sobre pares de características ya que ésta es la unidad de codificación empleada por el estándar DSR. De acuerdo con esto, tras la cuantización SVQ, cada par de características queda representado por un subvector $\mathbf{c}$ ($\mathbf{c} \in \{\mathbf{c}^{(i)}; i = 0, \ldots, 2^M - 1\}$). Esta modificación no supone cambios drásticos en la formulación descrita en la sección 5.3.1, ya que basta con considerar que los centroides $\mathbf{c}^{(i)}$ y los subvectores observados $\hat{\mathbf{c}}_t$ se corresponden, respectivamente, con $\mathbf{x}^{(i)}$ e $\mathbf{y_t}$ en las

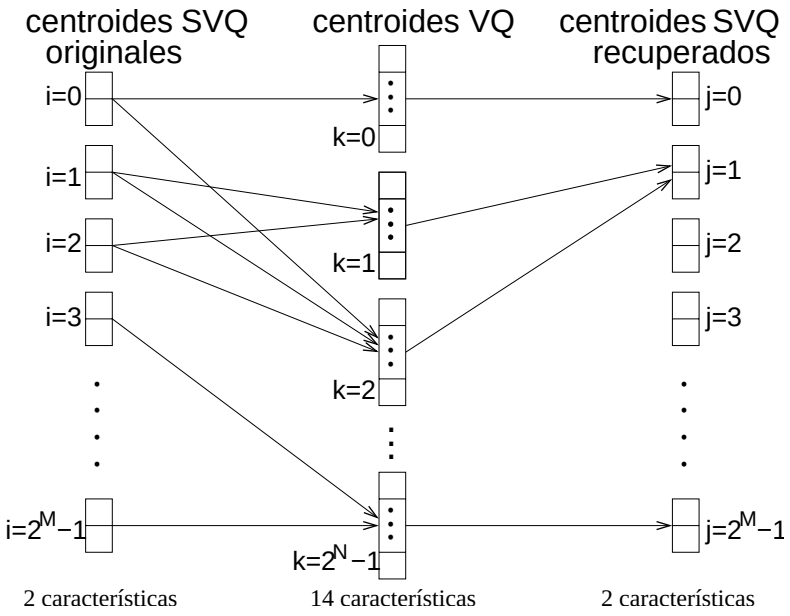


Figura 6.5: Ejemplo de la secuencia de cuantizaciones aplicadas a las réplicas correspondientes a un par de características SVQ.

ecuaciones (5.10-5.19). Obtenida la estimación sobre un par de características, operaremos de igual forma sobre el resto.

Las probabilidades de observación $b_i(\hat{\mathbf{c}}_t) = P(\hat{\mathbf{c}}_t|\mathbf{c}^{(i)})$ ($b_i(\hat{\mathbf{y}}_t)$ según la formulación de la sección 5.3.1) se determinan dependiendo del tipo de vector de características considerado. Así, distinguiremos entre los siguientes casos:

- En el instante t se dispone de un vector correcto, es decir, un vector recibido que no es una réplica. Para cada subvector, la probabilidad de observación viene dada por,

$$b_i(\hat{\mathbf{c}}_t) = \begin{cases} 0 & \mathbf{c}^{(i)} \neq \hat{\mathbf{c}}_t \\ 1 & \mathbf{c}^{(i)} = \hat{\mathbf{c}}_t \end{cases} \tag{6.2}$$

  Estos vectores se encuentran únicamente en los extremos de la ráfaga ($t = 0$ o $t = T+1$) y si la ráfaga se sitúa al principio o al final de una frase sólo dispondremos de uno de ellos. En este último caso, al igual que durante la ráfaga, los vectores no correctos se deberán tratar como réplicas o vectores perdidos.

- En el instante de tiempo t se dispone de una réplica, es decir, una versión degradada del vector original. En este caso, la réplica se divide en pares de características que son nuevamente cuantizadas mediante SVQ. Cada par queda representando por un centro SVQ, $\mathbf{c}_t^{(j)}$, que identificaremos como *centro SVQ recuperado*. La figura 6.5

muestra un ejemplo de correspondencias entre los distintos tipos de centros considerados. Se observa que un centro SVQ recuperado puede corresponder a varios centroides VQ, lo cuales, a su vez, pueden corresponder a distintos centros originales. Esto se debe a que la cuantización VQ considera todas las características mientras que la cuantización SVQ tan sólo un par. Por tanto, tras el doble proceso de cuantización, dado un centro SVQ original $\mathbf{c}_t^{(i)}$, es posible observar diferentes centros SVQ recuperados $\mathbf{c}_t^{(j)}$. Entonces, la probabilidad de que se observe uno u otro puede obtenerse a partir de la base de datos de entrenamiento, mediante un triple proceso de cuantización (SVQ-VQ-SVQ), como,

$$b_i(\hat{\mathbf{c}}_t = \mathbf{c}^{(j)}) = P(\mathbf{c}^{(j)}|\mathbf{c}^{(i)}) = \frac{\text{nº centros recuperados } j \text{ dado centro original } i}{\text{nº centro original } i}. \tag{6.3}$$

Si $\mathbf{c}^{(i)}$ representa un centro vacío (ningún vector de entrenamiento es cuantizado en dicho centro), un umbral ϵ de entrenamiento insuficiente se asigna a $b_i(\mathbf{c}^{(j)})(\forall j = 0, \ldots, 2^M - 1)$.

- Finalmente, si en el instante t no se ha recibido ni un vector ni una réplica, se supondrá una cuantización VQ degenerada con 0 bits. De esta forma, todos los centros SVQ originales se corresponden con un único centro VQ, el vector medio de toda la base de datos, que a su vez corresponde a uno sólo de los posibles centros SVQ recuperados. Entonces, las probabilidades de observación serán asignadas como,

$$b_i(\hat{\mathbf{c}}_t) = \begin{cases} 1 & \text{si } \mathbf{c}^{(i)} \text{ no es un centro vacío} \\ \epsilon & \text{si } \mathbf{c}^{(i)} \text{ es un centro vacío} \end{cases} \tag{6.4}$$

donde, como antes, un umbral ϵ de entrenamiento insuficiente se asigna a aquellos centros en los que ningún vector de entrenamiento ha sido cuantizado. Como puede observarse, en este caso el algoritmo progresa guiado principalmente por las probabilidades de transición del modelo del reconocedor.

Este esquema implícitamente supone el uso de un modelo HMM discreto en la estimación FB–MMSE, en donde se emplea un conjunto discreto de observaciones. También sería posible modelar las probabilidades de observación mediante funciones de densidad de probabilidad dadas por alguna función paramétrica apropiada. Esto permitiría obtener un modelo HMM continuo, haciéndose innecesario el segundo proceso SVQ. Sin embargo, dado que este proceso de cuantización no representa ninguna reducción en el rendimiento del reconocedor [51, 142], hemos preferido la versión discreta por simplicidad.

6.2.4. Resultados experimentales

Las técnicas descritas anteriormente han sido evaluadas en una arquitectura DSR operando sobre un canal con pérdidas. El uso de una arquitectura DSR se justifica en base a dos razones. La primera es que, al igual que en el capítulo anterior, nuestro interés se centra ahora en la pérdida de información y no en otros ruidos derivados. Por otro lado, como adelantábamos en la introducción, las técnicas de recuperación basadas en el emisor se contraponen al interés de las arquitecturas NSR, ya que requieren de la modificación del terminal del cliente.

El marco experimental en el que se realizan las pruebas es prácticamente idéntico al propuesto para la evaluación de las técnicas de mitigación del capítulo 5. Así, la extracción de características se realizará de acuerdo con el front-end básico estandarizado por ETSI [74]. Los vectores de características obtenidos (vectores SVQ) serán empaquetados conforme al formato de carga útil propuesto para DSR sobre IP (RFC 3557 [79]), con la salvedad de que ahora se incluirán bits adicionales para la transmisión de las réplicas. De acuerdo con el esquema FEC descrito y con las recomendaciones del estándar, un único par de vectores SVQ se transmite en cada paquete.

La simulación de las pérdidas de paquetes se realiza a través del modelo de tres estados descrito en la sección 3.7.1. El rendimiento de las técnicas basadas en emisor depende de la forma en que se producen las recepciones, ya que dependen de ellas para introducir los mecanismos de recuperación (en el caso de los FECs, las réplicas). Por tanto, durante la evaluación de estas técnicas, será necesario emplear un modelo que considere no sólo distribución de las pérdidas sino también de las recepciones. El modelo de Gilbert, empleado en el capítulo anterior, no permite especificar una longitud de recepciones consecutivas entre ráfagas, ILPL, independientemente de la longitud de las ráfagas, LPL, (véase sección 3.7.1). En cambio, en el modelo de tres estados es posible establecer ambos parámetros por separado.

Ya que las técnicas serán evaluadas para distintas latencias y tamaños de diccionario, a fin de evitar un uso (aún más) abusivo de las tablas, únicamente consideraremos 5 condiciones de canal. La tabla 6.1 muestra los parámetros empleados para generar las condiciones. Dos de ellos, R_{loss} y L_{loss}, ya han sido descritos con anterioridad, indicando L_{recep} la longitud de las recepciones entre ráfagas durante los periodos de congestión. En todas las condiciones se considerará una separación entre los periodos con congestión y libres de ella de 50 paquetes. En esta misma tabla se muestran, como referencia, los resultados obtenidos por la técnica de mitigación del estándar de Aurora, por MMSE

Condición	*Parámetros*			*Mitigación*		
	R_{loss}	L_{loss}	L_{recep}	Aur	MMSE	MAP
1	10 %	2	4	98.47	98.53	98.57
2	20 %	4	4	93.57	95.06	95.05
3	30 %	6	3	85.47	88.20	88.45
4	40 %	8	3	76.51	80.58	81.12
5	50 %	10	2	65.71	70.24	70.70

Tabla 6.1: Parámetros del modelo de 3 estados para cada condición y resultados de referencia obtenidos por Aurora (Aur), MMSE con un modelo de orden 3 reducido (MMSE), y MAP aplicado en bandas cepstrales (MAP).

con un modelo de orden 3 reducido, y por MAP aplicado en bandas cepstrales. Como puede observarse, el modelado de las recepciones tiene escaso interés para las técnicas de mitigación, obteniéndose resultados similares a los correspondientes empleando un modelo de Gilbert (véase tabla 5.14).

Las tablas 6.2, 6.3 y 6.4 muestran los resultados obtenidos mediante la aplicación directa de réplicas, descrita en la sección 6.2.2, cuantizadas con diccionarios VQ de 4, 16 y 256 centros. Se han considerado latencias que van desde los 120 ms a los 600 ms, con T_{fec}= 6, 8, 12, 16, 20, 24 y 30, aunque debe notarse que las latencias más altas, especialmente la de 600 ms, pueden no resultar viables en la práctica (a las restantes latencias debe añadirse el retraso introducido por la red) ya que degradarían la naturalidad de la interacción oral. En general, puede observarse una mejora de los resultados conforme se aumenta la latencia. Esto se debe a que un mayor número de paquetes, alejados temporalmente de la ráfaga, pueden introducir réplicas en ella, permitiendo una mayor fragmentación.

Atendiendo al tamaño de las réplicas, puede observarse que si el grano de cuantización es suficientemente fino (tablas 6.3 y 6.4), la recuperación de ciertas tramas, ligada a la ruptura de ráfagas en otras menores, da lugar a una mejora notable de la precisión en el reconocimiento, en relación a la técnica estándar. Con réplicas de 8 bits se obtienen resultados superiores, incluso a bajas latencias, a los alcanzados con técnicas de mitigación avanzadas (tabla 6.1). Lo mismo sucede para diccionarios de 4 bits, pero asumiendo latencias mayores (a partir de 160 ms). Sin embargo, si las réplicas están fuertemente cuantizadas, puede llegar a obtenerse el efecto contrario, esto es, una reducción del rendimiento. Así, para el caso en que las réplicas están cuantizadas con sólo dos bits (tabla 6.2), a bajas latencias se obtienen incluso peores resultados que la técnica estándar y, en general, resultados inferiores a los alcanzados con técnicas de mitigación avanzadas.

Cond.	*Latencia (ms)*						
	120	160	240	320	400	480	600
1	98.58	98.67	98.57	98.71	98.59	98.64	98.47
2	93.17	93.21	93.79	93.81	94.05	93.67	93.46
3	86.47	86.89	87.61	87.34	87.90	88.03	88.11
4	77.11	78.30	79.43	79.50	80.56	80.64	80.55
5	63.76	65.11	66.63	67.25	67.08	66.90	67.46

Tabla 6.2: Resultados obtenidos mediante reparación con réplicas VQ de 2 bits.

Cond.	*Latencia (ms)*						
	120	160	240	320	400	480	600
1	98.76	98.85	98.73	98.86	98.83	98.78	98.72
2	94.97	95.11	95.93	95.58	95.92	95.82	95.93
3	89.68	90.48	91.75	92.06	92.89	92.90	93.07
4	81.85	83.58	85.25	86.32	87.11	88.00	88.52
5	69.39	71.61	73.85	74.87	75.79	76.02	77.27

Tabla 6.3: Resultados obtenidos mediante reparación con réplicas VQ de 4 bits.

Cond.	*Latencia (ms)*						
	120	160	240	320	400	480	600
1	98.88	98.92	98.93	98.95	98.90	98.83	98.85
2	95.76	95.89	96.59	96.53	96.80	96.79	96.66
3	91.27	92.54	93.75	94.42	94.89	95.08	95.27
4	84.11	85.93	88.45	89.55	90.51	91.51	91.63
5	72.35	75.40	77.71	79.69	80.44	81.26	82.25

Tabla 6.4: Resultados obtenidos mediante reparación con réplicas VQ de 8 bits.

La razón por la que las réplicas fuertemente cuantizadas resultan dañinas para el reconocimiento radica en el uso que hace de ellas el algoritmo de mitigación, considerándolas iguales a vectores recibidos. Si en vez de recurrir a una aplicación directa, se considera la estimación FB–MMSE con réplicas VQ descrita en la sección 6.2.3, la información presente en estas réplicas, aunque escasa y degradada, puede contribuir a una mejora del rendimiento. Las tablas 6.5, 6.6 y 6.7 muestran los resultados cuando se aplica una estimación FB–MMSE basada en réplicas VQ con diccionarios de 4, 16 y 256 centros. Como puede observarse, la estimación FB–MMSE con réplicas da lugar a una mejora notable de la precisión con respecto a la aplicación directa de las réplicas, en todos los casos. En general y a pesar de estar aplicando cuantizaciones VQ tan extremas, se alcanzan resultados significativamente mejores a los de la técnica estándar y, salvo para los casos más

	Latencia (ms)						
Cond.	120	160	240	320	400	480	600
1	98.83	98.75	98.71	98.78	98.79	98.69	98.72
2	95.28	95.60	96.07	95.79	96.26	95.81	95.97
3	88.88	89.49	90.48	90.40	90.74	90.87	90.86
4	80.33	81.50	83.22	83.59	84.18	84.68	84.16
5	69.40	71.39	73.43	73.61	73.59	73.56	74.30

Tabla 6.5: Resultados obtenidos mediante estimación MMSE con réplicas VQ de 2 bits.

	Latencia (ms)						
Cond.	120	160	240	320	400	480	600
1	98.96	98.94	98.86	98.93	98.97	98.85	98.85
2	96.93	97.12	97.86	97.70	98.19	97.97	98.21
3	92.03	92.88	94.36	95.03	95.57	95.60	96.13
4	84.15	86.58	88.58	89.99	90.83	91.61	91.96
5	74.32	77.03	79.61	81.02	81.45	81.87	83.13

Tabla 6.6: Resultados obtenidos mediante estimación MMSE con réplicas VQ de 4 bits.

	Latencia (ms)						
Cond.	120	160	240	320	400	480	600
1	99.00	98.92	98.93	98.99	99.00	98.91	98.88
2	97.52	97.79	98.37	98.42	98.68	98.63	98.64
3	93.33	94.20	96.17	96.49	97.05	97.24	97.66
4	86.50	88.79	91.56	92.53	93.69	94.23	94.68
5	77.44	80.54	83.39	84.94	85.58	86.17	87.44

Tabla 6.7: Resultados obtenidos mediante estimación MMSE con réplicas VQ de 8 bits.

restrictivos (2 bits y 120 ms de latencia), a los obtenidos por MMSE con modelo de orden 3 reducido y MAP aplicado por bandas.

6.2.5. Formato de carga útil e implementación en IP

Incluso con tan sólo unos pocos bits adicionales, la estimación FB–MMSE con réplicas VQ puede resultar muy efectiva para la mitigación de los efectos de las pérdidas debidas al canal IP en una arquitectura DSR. Sin embargo, estos bits destinados a representar las réplicas deben introducirse de alguna forma en los paquetes. En esta sección proponemos varias alternativas. Para ello recurriremos a la especificación del formato de carga útil DSR para el protocolo RTP [79], descrita con anterioridad en la sección 3.4.2.

Recordemos que, según las recomendaciones de esta especificación, los paquetes se alinean en palabras de 8 bits, como se muestra en la figura 3.10. Este alineamiento permite a los routers un fácil acceso a las cabeceras, agilizando la comunicación. Sin embargo, dado que la longitud original de un FP (pareja de tramas) es de 92 bits y ésta cantidad no puede descomponerse de forma entera en octetos, es necesario añadir al paquete un relleno de cuatro bits todos iguales a cero. Inicialmente estos bits están libres, lo que sugiere algunas formas de introducir los códigos FEC propuestos.

- Los 4 bits de relleno podrían emplease para introducir dos réplicas VQ representadas mediante un diccionario de 4 centros (2 bits por réplica). En este caso, las réplicas están fuertemente cuantizadas. Como se desprende de la tabla 6.2, un uso directo de las réplicas conduce a unos resultados inferiores a los obtenidos por el estándar Aurora. Por ello, con esta configuración se hace necesario aplicar una técnica de post-procesamiento para las réplicas. Por ejemplo, la estimación FB–MMSE propuesta.

- Adicionalmente, los 4 bits dedicados al CRC podrían emplearse para introducir códigos FEC de mayor tamaño. Los bits de CRC se mantienen en la recomendación de carga por razones de compatibilidad, ya que los 16 bits de comprobación del protocolo UDP, junto con los mecanismos de seguridad aplicados usualmente en la capa física, así como la prueba de coherencia aplicada en el receptor, deberían ser suficientes para asegurar la integridad de los datos. En este caso se podrían transmitir dos réplicas cuantizadas con un diccionario de 16 centros (4 bits/réplica), obteniéndose los resultados de la tabla 6.3 si las réplicas se aplican directamente, o los de la tabla 6.6 si se recurre a la estimación FB–MMSE.

- Finalmente, cualquier otro incremento en el tamaño de los códigos FEC requiere la introducción de una nueva palabra de 8 bits en el paquete. En este caso, los 8 bits dedicados al CRC y al relleno, junto con la nueva palabra, permitirían la codificación de 2 réplicas cuantizadas con diccionarios de 256 centros (8 bits/réplica), obteniéndose una importante mejora del reconocimiento (tablas 6.4 y 6.7).

Como puede observarse, las dos primeras implementaciones permiten, mediante el uso de réplicas VQ, una mejora significativa en la precisión del reconocimiento evitando un incremento efectivo del ancho de banda necesario. En el último caso, en cambio, es necesario la introducción de una nueva palabra de 8 bits. Sin embargo, este incremento resulta ínfimo en comparación con el tamaño total del paquete, con 96 bits de carga útil más una cabecera IP de 160 bits, UDP de 64 bits y RTP de, al menos, 96 bits.

6.3. Entrelazado de Tramas

El entrelazado es una sencilla pero efectiva técnica de prevención de errores usada en múltiples sistemas de comunicaciones. Tradicionalmente, el entrelazado ha sido aplicado a nivel de bits, con el fin de aleatorizar la aparición de errores y permitir a las técnicas de corrección un mayor éxito en su desempeño. Aunque el entrelazado de bits puede incrementar la robustez frente a cierto tipo de canales [156], carece de utilidad en los canales con pérdidas, en donde se pierden vectores consecutivos completos. Sin embargo, en estos canales es posible recurrir al entrelazado de unidades de transmisión de mayor tamaño, como las tramas o vectores de características. Para diferenciar a este último tipo de entrelazado, nos referiremos a él como *entrelazado de tramas*.

El entrelazado de tramas permuta el orden en que los vectores de características son transmitidos. En la figura 6.6 se muestra un ejemplo de entrelazado. Como puede observarse, una vez que las tramas son restauradas en su orden original, las pérdidas consecutivas se perciben en el receptor como ráfagas más cortas. Esta dispersión de las pérdidas resulta ventajosa ya que las técnicas de mitigación, al depender de la similitud a corto plazo de la señal de voz, funcionan significativamente mejor recuperando lapsos de tiempo cortos.

Sin embargo, si bien el entrelazado de tramas puede lograr una mejora del rendimiento del reconocedor, ésta se consigue a costa de un incremento en la latencia de la comunicación. Mientras que en los entrelazadores de bits el entrelazado se realiza generalmente dentro de la propia trama, el entrelazado de tramas involucra a varias de ellas, las cuales deben ser temporalmente almacenadas y reordenadas antes de la transmisión. Esto necesariamente implica un retraso en la comunicación. Igualmente se hace necesario un incremento de la memoria en el receptor pero, teniendo en cuenta la capacidad de los dispositivos actuales, esto es un problema menor. En reconocimiento, la latencia no es tan crítica como en transmisión de voz. Aunque sería deseable, no se espera una respuesta inmediata del reconocedor, por lo que un incremento de la latencia en unos pocos cientos de milisegundos no resulta significativo de cara a la calidad global del servicio. Sin embargo, un retraso excesivo en la respuesta puede degradar la naturalidad de la conversación. Algunos autores establecen este limite en torno a los 500 ms [50].

A causa del coste en latencia que supone su uso, una gran variedad de entrelazadores han sido propuestos y analizados. Andrews et al. [157] da un formalismo matemático a la técnica de entrelazado, facilitando la comparación y el análisis de los entrelazadores con respecto a su capacidad de dispersar pérdidas y sus requerimientos de memoria y latencia.

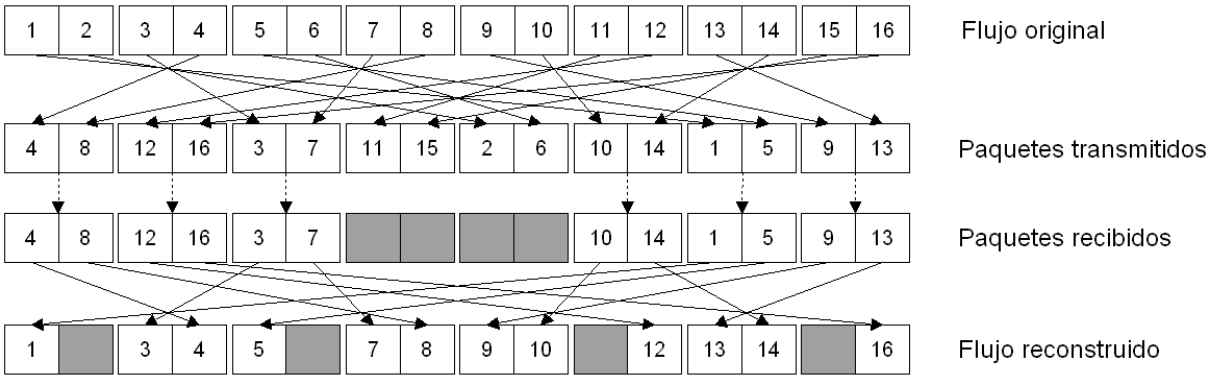


Figura 6.6: Ejemplo de entrelazado de vectores empleando el entrelazador de bloque de latencia mínima ($s = 4$) sobre paquetes de dos tramas.

6.3.1. Análisis matemático de los entrelazadores

Formalmente, un entrelazador se define como una máquina de estados de una entrada y una salida, que toma una secuencia de símbolos de un alfabeto fijo y produce una secuencia de salida sobre el mismo alfabeto idéntica a la de entrada excepto por el orden. Sea $\ldots, a_{-2}, a_{-1}, a_0, a_1, a_2, \ldots$ la secuencia de entrada y $\ldots, b_{-2}, b_{-1}, b_0, b_1, b_2, \ldots$ la secuencia de salida, entonces un entrelazador es una permutación $\pi : \mathcal{Z} \rightarrow \mathcal{Z}$ tal que $a_i = b_{\pi(i)}$. El entrelazador es una máquina de estados finita únicamente si la permutación π es periódica, esto es,

$$\pi(i + p) = \pi(i) + p \quad \forall i \tag{6.5}$$

donde p es el periodo de π. Para cada entrelazador puede definirse un *desentrelazador* o *entrelazador inverso*, π^{-1}. El desentrelazador se aplica sobre la salida del entrelazador, devolviendo, con un posible retraso temporal, los elementos a su orden original, es decir,

$$\pi^{-1}(\pi(t)) = t + l \quad \forall t \tag{6.6}$$

donde l es la latencia que introduce el proceso completo de entrelazado/desentrelazado expresada en número de símbolos. Esta latencia viene dada por la suma de los retardos introducidos por el entrelazador y su inverso, esto es,

$$l_\pi = l_\pi^+ + l_{\pi^{-1}}^+ \tag{6.7}$$

donde,

$$l_{\pi}^{+} = max\{\pi(x) - x\}, \tag{6.8}$$

$$l_{\pi^{-1}}^{+} = max\{\pi^{-1}(x) - x\}. \tag{6.9}$$

Un entrelazador es físicamente realizable si es *causal*, esto es, si para todo i, $\pi(i) \geq i$. Un entrelazador causal verifica que todos los símbolos salen posteriormente a su entrada y no antes. Si, además, el entrelazador verifica que $\pi(i) = i$ para algún i, entonces se considera *mínimo causal*. Puede comprobarse que,

$$l_{\pi^{-1}}^{+} = -min\{\pi(x) - x\} = -l_{\pi}^{-}, \tag{6.10}$$

$$l_{\pi} = l_{\pi}^{+} - l_{\pi}^{-} \tag{6.11}$$

donde l_{π}^{-} es el retraso necesario (positivo o negativo) que convierte al entrelazador π en mínimo causal [157]. Puede observarse que para un entrelazador mínimo causal la latencia viene dada por $l_{\pi} = l_{\pi}^{+}$, ya que $l_{\pi}^{-} = 0$.

La capacidad de un entrelazador para dispersar pérdidas viene dada por su *dispersión*. Se dice que un entrelazador tiene una dispersión (s, t) si verifica que,

$$|\pi(i) - \pi(j)| \geq t \quad \text{siempre que,} \quad |i - j| < s. \tag{6.12}$$

Es decir, un entrelazador de dispersión (s, t) reordena la secuencia de entrada de forma que ninguna secuencia consecutiva de t símbolos contenga ningún símbolo que estuviera separado por menos de s símbolos en el orden original. Puede demostrarse que si un entrelazador tiene una dispersión (s, t) su inverso tiene una dispersión (t, s) [158], es decir,

$$|\pi^{-1}(i) - \pi^{-1}(j)| \geq s \quad \text{siempre que,} \quad |i - j| < t. \tag{6.13}$$

Cuando $s = t$, es bastante común abreviar dispersión (s, s) por s. Una mayor dispersión suele implicar una mayor latencia. Por ello, se buscan aquellas permutaciones π que ofrezcan la mayor dispersión con la menor latencia posible.

6.3.2. Entrelazadores de bloque de latencia mínima

Un entrelazador que ha sido aplicado con éxito a DSR en canales con pérdidas es el entrelazador de bloque de latencia mínima (*MLBI–Minimum Latency Block Interleaver*)

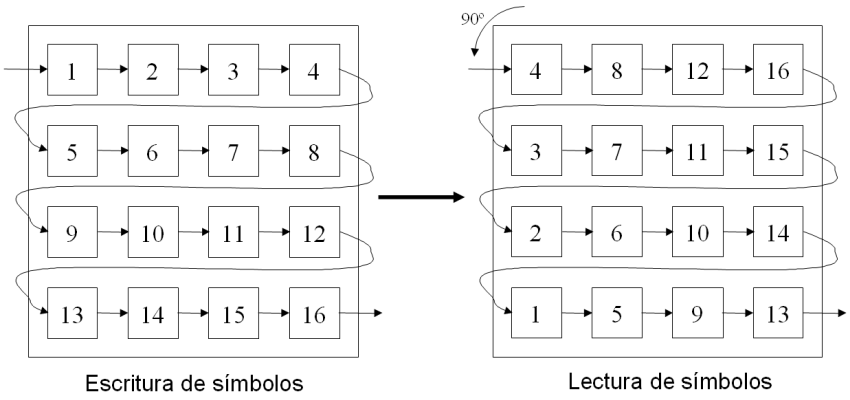


Figura 6.7: Entrelazado de bloque de latencia mínima mediante rotación de matrices de 4×4.

[108, 109, 154]. Este entrelazador pertenece a una conocida clase de entrelazadores denominada *entrelazadores de bloque*, ampliamente usada en una gran variedad de sistemas de comunicación. Formalmente, un entrelazador π es un entrelazador de bloque de tamaño p si p es el periodo de π y existe un intervalo de longitud p cuya imagen en π es también un intervalo de longitud p. De forma intuitiva, un entrelazador de bloque es aquel que permuta entre si los p elementos que conforman su periodo. Puede demostrarse que, entre todos los entrelazadores de bloque con dispersión s, existen únicamente dos entrelazadores que resultan óptimos en términos de latencia [157]. Estos dos entrelazadores vienen dados por,

$$\pi_1(is+j) = (s-1-j)s+i \quad 0 \leq i,j \leq s-1, \tag{6.14}$$

y

$$\pi_2(is+j) = js+(s-1-i) \quad 0 \leq i,j \leq s-1, \tag{6.15}$$

Ambos entrelazadores forman un *par invertible*, esto es, $\pi_1 = \pi_2^{-1}$ y $\pi_2 = \pi_1^{-1}$. El retraso introducido por cada uno de ellos viene dado por $l^+ = s(s-1)$, de donde se deduce que la latencia introducida en la transmisión viene dada por $l = 2s^2 - 2s$. Esta latencia es la mínima posible entre todos los entrelazadores de bloque con dispersión s. Por ello, Andrew et al. [157] los denominaron entrelazadores de bloque de latencia mínima.

La implementación de estos entrelazadores puede llevarse a cabo de forma sencilla mediante una matriz de dimensión $s \times s$. Los símbolos entrantes (vectores de características) se organizan por filas en la matriz. Así, el símbolo correspondiente al instante de tiempo $t = is+j$ ($i = 0,\ldots,s-1$,$j = 0,\ldots,s-1$) se introduce en la posición (i,j). Tras esto, la matriz se rota 90º, y los símbolos son extraídos por filas (véase figura 6.7). En el

ejemplo de la figura 6.6, el orden de los vectores durante la transmisión viene dado por un entrelazador de bloque de latencia mínima y dispersión $s = 4$.

6.3.3. Entrelazadores de Ramsey aplicados al reconocimiento distribuido

Aunque los entrelazadores anteriores resultan óptimos en términos de latencia, estos asumen que t es igual a s, es decir, una dispersión (s, s). Una clase alternativa de entrelazadores son los *entrelazadores convolucionales*, constituida por aquellos entrelazadores que no son de bloque. Ramsey probó que es posible construir entrelazadores pertenecientes a esta clase con dispersión (s, t), en donde $s \neq t$, y que, bajo ciertas condiciones de primalidad, también implican una latencia mínima [158].

Como se puede deducir de la ecuación 6.13, si una ráfaga de pérdidas consecutivas aparece desde i hasta j con una longitud inferior a t, entonces un entrelazador con dispersión (s, t) dispersará dicha ráfaga en pérdidas aisladas separadas por, al menos, $s - 1$ tramas. Por tanto, un entrelazador MLBI sólo puede dispersar completamente ráfagas de longitud igual a la distancia a la que las pérdidas aisladas son dispersadas $(t = s)$. Por el contrario, los entrelazadores de Ramsey no requieren de la condición $s = t$, de tal forma que los incrementos de s posibilitan una mayor dispersión de las pérdidas aisladas, mientras que los incrementos de t hacen posible contrarrestar ráfagas más largas.

En este punto debe considerarse la técnica de mitigación aplicada en el receptor. Por ejemplo, la técnica propuesta por el estándar de Aurora utiliza sólo una trama recibida para obtener los reemplazos de los vectores perdidos. Así, la primera mitad de la ráfaga se reconstruye repitiendo el último vector recibido antes de ésta, y la segunda mitad repitiendo el primero recibido tras ella. Como puede observarse, la inserción de más de una trama entre las pérdidas no permitiría una mejora del rendimiento de la técnica de mitigación, ya que ésta no puede aprovechar dichas tramas. Sin embargo, este innecesario aumento de la dispersión tendrá su coste en latencia. Suponiendo entonces una técnica de mitigación que sólo requiera una recepción entre las pérdidas, un entrelazador con dispersión $(s = 2, t)$ podría proporcionar un rendimiento similar a otro con dispersión $(s > 2, t)$, pero con una latencia menor.

Basándonos en este análisis, en este trabajo proponemos un entrelazador convolucional de latencia mínima derivado del *desentrelazador (t, s) de tipo III* propuesto por Ramsey

[158]. Este desentrelazador puede formularse como,

$$\pi_{III}^{-1}(i) = (i \text{ div } s) \cdot s - (i \text{ mod } s) \cdot t. \tag{6.16}$$

Cuando s y t son primos relativos y $t > s$, el entrelazador correspondiente es un entrelazador con dispersión (s, t) de latencia mínima, dada por $(s-1)(t+1)$ [159]. Entonces, podemos suponer $s = 2$ y $t = 2B + 1$ $(B \geq 1)$, obteniendo el siguiente par invertible de entrelazadores:

$$\pi(i) \quad = i + (i \text{ div } 2) \cdot 2(B+1) \tag{6.17}$$

$$\pi^{-1}(i) \quad = (i \text{ div } 2) \cdot 2 - (i \text{ mod } 2) \cdot (2B+1). \tag{6.18}$$

Este entrelazador proporciona una dispersión $(2, 2B+1)$, pudiendo fragmentar ráfagas de pérdidas de longitud igual o menor a $2B$ en pérdidas aisladas separadas entre si por una única trama. Además, dado que s y t son siempre primos relativos y $t > s$, el entrelazador tiene latencia mínima, igual a $l = 2(B+1)$.

Una ventaja adicional que presenta este entrelazado es su flexibilidad cuando una latencia máxima ha sido establecida. En el caso de los entrelazadores MLBI, la latencia depende del parámetro s, creciendo cuadráticamente respecto a él $(l = 2s^2 - 2s)$. Por el contrario, en los entrelazadores propuestos, la latencia depende linealmente del parámetro B $(l = 2(B+1))$.

6.3.4. Resultados experimentales

A fin de evaluar el rendimiento del entrelazado propuesto en comparación con los entrelazadores MLBI descritos anteriormente, ambos esquemas han sido aplicados a una arquitectura DSR sobre un canal con pérdidas. El marco experimental de las pruebas coincide con el descrito en la sección 6.2.4. La única salvedad con respecto al marco anterior radica en el orden en que los vectores son introducidos en los paquetes. Este orden viene dado por la función de entrelazado a evaluar. Como es de esperar, los vectores son restaurados en su orden original por el desentrelazador correspondiente, previamente al reconocimiento.

La tabla 6.8 muestra los resultados de precisión en el reconocimiento obtenidos con el entrelazador MLBI en comparación con un sistema DSR sin entrelazado. La técnica de mitigación empleada es la propuesta por el estándar de Aurora. Se han considerado dispersiones de s=3, 4, 5 y 6, que causan latencias de 120, 240, 400 y 600 ms. Como puede

		MLBI				*Ramsey* $(2,t)$			
Cond.	Aur.	120 ms	240 ms	400 ms	600 ms	120 ms	240 ms	400 ms	600 ms
1	98.47	98.93	98.89	99.03	99.04	99.01	98.95	99.00	98.98
2	93.57	96.03	97.21	97.94	98.26	97.42	98.25	98.35	98.52
3	85.47	89.68	92.50	94.65	95.77	93.39	95.20	95.64	96.51
4	76.51	81.79	86.02	89.08	91.10	86.69	89.76	91.35	92.06
5	65.71	71.75	75.20	79.64	81.44	77.74	80.27	81.96	83.15

Tabla 6.8: Resultados obtenidos empleando entrelazadores convolucionales de Ramsey $(2,t)$ en comparación con el entrelazado de bloque de latencia mínima (MLBI) y un sistema sin entrelazado, todos con mitigación estándar (Aur.).

		MLBI				*Ramsey* $(2,t)$			
Cond.	MMSE	120 ms	240 ms	400 ms	600 ms	120 ms	240 ms	400 ms	600 ms
1	98.53	98.92	98.92	98.99	99.02	98.96	98.94	99.04	99.00
2	95.06	96.97	97.60	98.22	98.45	97.80	98.38	98.51	98.55
3	88.20	92.33	93.92	95.29	96.58	94.45	95.58	96.35	97.01
4	80.58	85.96	88.93	90.82	92.83	88.90	91.28	92.69	93.10
5	70.70	76.78	79.06	82.68	84.63	81.00	82.14	84.18	85.61

Tabla 6.9: Resultados obtenidos empleando entrelazadores $(2,t)$ en comparación con el entrelazado MLBI y un sistema sin entrelazado, todos con mitigación MMSE con modelo de orden 3 reducido (MMSE).

		MLBI				*Ramsey* $(2,t)$			
Cond.	MAP	120 ms	240 ms	400 ms	600 ms	120 ms	240 ms	400 ms	600 ms
1	98.57	98.95	98.91	99.00	98.95	99.03	98.97	99.00	99.01
2	95.05	97.04	97.72	98.25	98.41	97.93	98.38	98.60	98.62
3	88.45	92.49	94.11	95.39	96.69	94.72	95.92	96.41	97.00
4	81.12	85.91	89.07	91.02	93.29	89.62	91.60	92.85	93.45
5	70.70	77.04	79.50	82.68	84.87	81.24	82.86	84.28	86.04

Tabla 6.10: Resultados obtenidos empleando entrelazadores $(2,t)$ y MLBI en comparación con un sistema sin entrelazado, todos con mitigación MAP por bandas (MAP).

observarse, mediante incrementos en la dispersión del entrelazador pueden obtenerse mejores resultados, pero en dicho caso deben asumirse mayores latencias. Esta tabla también muestra los resultados obtenidos por un entrelazador basado en el entrelazado convolucional de Ramsey con dispersión (s, t) bajo las mismas condiciones y latencias. Como puede apreciarse, este entrelazador mejora significativamente la precisión en el reconocimiento en comparación con el MLBI, especialmente a bajas latencias.

Aunque el entrelazador propuesto ha sido diseñado teniendo en cuenta un algoritmo de mitigación que sólo requiere una recepción entre las pérdidas, hemos realizado pruebas de reconocimiento empleando técnicas de mitigación más avanzadas que exploten más de un vector previo o posterior a la ráfaga. Éste es el caso de la estimación MMSE con modelo de orden 3 reducido y la estimación MAP por bandas. Las tablas 6.9 y 6.10 muestran los resultados obtenidos empleando como mitigación estas técnicas, con los dos esquemas de entrelazado. Como puede observarse, los resultados obtenidos por el entrelazador propuesto son superiores a los obtenidos por el entrelazado MLBI. Esto parece indicar que, aunque se empleen técnicas de mitigación más avanzadas que pudieran requerir una separación mayor entre las pérdidas, suponer $s = t$ resulta excesivo y claramente contraproducente, al menos, para el reconocimiento distribuido.

6.4. Tratamiento de pérdidas en reconocedor

Usualmente, durante el proceso de reconocimiento de voz, se asume que todas las observaciones suministradas al reconocedor son igualmente relevantes. Sin embargo, pueden encontrarse situaciones en las que resultaría conveniente considerarlas de forma diferenciada. Esto ocurre cuando no tenemos certeza acerca de los valores de ciertas observaciones y nuestra confianza no es la misma para todas ellas. Es el caso, por ejemplo, del reconocimiento remoto a través de un canal digital ruidoso. Debido a los errores de transmisión, algunas características de la voz pueden estar afectadas por el ruido y otras ser recibidas sin errores. En este caso, la fiabilidad de las características decodificadas depende del estado del canal ya que, conforme éste se degrada, menos pueden garantizarse los valores recibidos. Otra situación similar ocurre durante el reconocimiento de voz en entornos acústicamente ruidosos. Por diversas razones, ciertas características de la señal de voz pueden quedar más contaminadas por el ruido que otras. Al igual que antes, las observaciones presentarán distintos grados de incertidumbre.

Para tratar estos casos, diversos autores han propuesto modificaciones sobre los algoritmos de reconocimiento que permiten la introducción de valores de confianza o certeza

junto con las observaciones [160, 161, 162, 163]. Estas técnicas se engloban dentro del paradigma del reconocimiento con datos no fiables e incompletos (*MTD–Missing Data Techniques*). Lo que se persigue es que las observaciones con mayor confianza tengan un peso específico durante el reconocimiento superior a aquellas observaciones para cuyos valores tenemos una menor certeza.

En el contexto de canales con pérdidas, podemos considerar estas modificaciones sobre los algoritmos de reconocimiento como técnicas de recuperación basadas en el reconocedor. Éstas operan de forma distinta a las técnicas de mitigación, ya que no pretenden una reconstrucción de los vectores perdidos, sino que adaptan el reconocimiento a la presencia de éstos. De esta forma, la distinción en el receptor entre la decodificación de los datos recibidos (y mitigación de errores) y el reconocimiento propiamente dicho, como dos etapas diferenciadas de procesamiento, deja de tener sentido. Ahora, el reconocedor pasa a formar parte del decodificador, cuya meta consiste en la decodificación del mensaje de texto original, creado por el locutor remoto, a partir de los datos recibidos del canal.

La mayor ventaja de este tipo de técnicas radica en el uso implícito de un potente modelo estadístico de voz, el modelo empleado por el propio reconocedor. Este modelo de voz es superior a aquel aplicado por las técnicas de mitigación estadísticas que, como se mostró en el capítulo anterior, ofrecían los mejores resultados. Por ello, las técnicas basadas en el reconocedor suelen ofrecer resultados superiores a los de las técnicas de mitigación. Como contrapartida, estas técnicas suelen ser específicas del algoritmo de reconocimiento implementado.

En general, estas técnicas se apoyan en la asignación de un valor de confianza o fiabilidad para cada vector de características (o incluso para cada componente de estos). Estos valores de confianza se procesan en el reconocedor junto con las observaciones. Se pueden definir, entonces, dos problemas a resolver para la aplicación de estas técnicas:

- La definición de un criterio por el cual se asigna un valor de confianza o fiabilidad a un determinado vector (o componente).

- La aplicación de este valor de confianza durante el proceso de reconocimiento. Generalmente consiste en una modificación sobre el algoritmo de Viterbi, por su extendido uso en los sistemas de reconocimiento de habla continua.

En las siguientes dos secciones describiremos brevemente las soluciones disponibles para estos problemas.

6.4.1. Modificaciones al algoritmo de Viterbi

Como ya se comentó en la sección 2.3.3, resulta muy común agrupar los problemas derivados del reconocimiento basado en HMMs en tres clases, a saber, la evaluación del modelo, la búsqueda del camino óptimo en el modelo y el entrenamiento. La resolución al segundo problema, esto es, la búsqueda de la secuencia de estados óptima en el macromodelo dados unos valores observados, resulta especialmente relevante para el reconocimiento de voz continua. Decíamos entonces que este problema se resuelve por medio del algoritmo de Viterbi (*VA*). El algoritmo de Viterbi es una solución de programación dinámica que opera de forma similar a como lo hiciera el algoritmo adelante-atrás empleado en la estimación FB–MMSE. El objetivo del algoritmo radica en el cálculo de la función:

$$\delta_t(i) = \max_{q_1,q_2,\ldots,q_{t-1}} P\left(q_1, q_2, \ldots, q_{t-1}, q_t = i, x_1, x_2, \ldots, x_t|\lambda\right). \tag{6.19}$$

Esta función viene dada por la mejor secuencia $q_1, q_2, \ldots, q_{t-1}$ que hace posible la generación de las observaciones $x_1, x_2, \ldots, x_t$ siendo q_t el estado actual ($q_t = s_i$). Para obtenerla, el algoritmo de Viterbi calcula la probabilidad para cada estado j en cada instante t, multiplicando las probabilidades de transición a_{ij} entre estados del modelo y la probabilidad de observación $b_j(x_t)$ a lo largo de todo el camino. Adicionalmente se emplea una función auxiliar $\phi_t(j)$ que permite recuperar la secuencia de estados una vez que la recursión acaba. El procedimiento puede resumirse en los siguientes pasos:

1. Inicialización:

$$\delta_1(i) = \pi_i b_i(x_1) \qquad (1 \leq i \leq N) \tag{6.20}$$

$$\phi_1(i) = 0 \tag{6.21}$$

2. Recursión:

$$\delta_t(j) = \max_{1\leq i\leq N} \left[\delta_{t-1}(i)a_{ij}\right] b_j(\mathbf{x}_t) \quad (1 \leq j \leq N; 2 \leq t \leq T) \tag{6.22}$$

$$\phi_t(j) = \max_{1\leq i\leq N}{}^{-1} \left[\delta_{t-1}(i)a_{ij}\right] \quad (1 \leq j \leq N; 2 \leq t \leq T) \tag{6.23}$$

3. Finalización:

$$P^* = P[Q^*, O|\lambda] = \max_{1\leq i\leq N} \left[\delta_T(i)\right] \tag{6.24}$$

$$q_T^* = \max_{1\leq i\leq N}{}^{-1} \left[\delta_T(i)\right] \tag{6.25}$$

4. Recuperación del camino óptimo (secuencia de estados optima $q_1^*, q_2^*, \ldots, q_T^*$):

$$q_t^* = \phi_{t+1}(q_{t+1}^*) \qquad (T-1, T-2, \ldots, 1) \tag{6.26}$$

La complejidad computacional del algoritmo es significativa, del orden de $O(N^2T)$ (donde N es el número total de estados y T el de instantes de tiempo). Así, en sistemas de reconocimiento continuo con cierto nivel de complejidad, la carga computacional puede llegar a hacerse inmanejable. Por ello se recurre a *estrategias de poda* que limitan el número de estados candidatos considerados durante la recursión. Una estrategia muy común consiste en el establecimiento de un umbral por debajo del cual los estados candidatos son descartados [164].

Varios autores han propuesto modificaciones a este extendido algoritmo que permiten la introducción de valores de fiabilidad junto con las observaciones, de forma que sea posible expresar también la confianza en ellas.

Decodificación de Viterbi ponderada

Una aproximación directa es la propuesta por Bernard y Alwan [160, 161], y consiste en la ponderación de las probabilidades de observación de cada vector de características en el algoritmo de Viterbi. El algoritmo resultante se conoce como algoritmo de Viterbi ponderado (*WVA–Weighted Viterbi Algorithm*) e implica cambios mínimos con respecto al algoritmo VA. Estos consisten únicamente en la introducción de pesos en la ecuación (6.22), modificándola de la forma:

$$\delta_t(j) = \max_{1 \le i \le N} \left[\delta_{t-1}(i) a_{ij}\right] \left[b_j(\mathbf{x}_t)\right]^{\gamma_t} \tag{6.27}$$

donde γ_t toma valores en el intervalo $[0, 1]$ y representa la confianza en la observación $\mathbf{x}_t$. Cuando la observación es completamente fiable, se asigna $\gamma_t = 1$ y la ecuación (6.27) se hace equivalente a la ecuación (6.22). Por el contrario, si se desconfía completamente de la observación, se asigna $\gamma_t = 0$. De esta forma, el algoritmo VA ignora las observaciones, progresando guiado únicamente por las probabilidades de transición.

Esta elegante modificación tiene la ventaja de ser independiente tanto de la composición del vector de características como de la forma de las probabilidades de observación $b_j(\mathbf{x}_t)$. Gracias a esto, la técnica WVA puede aplicarse tanto a modelos HMM discretos como continuos. Sin embargo, puede darse el caso de un modelo continuo en el que la densidad de probabilidad para la producción de observaciones viene dada por una mezcla de

gaussianas multivariadas, es decir, la ecuación (2.20) (véase sección 2.3.1) tiene la forma,

$$b_j(\mathbf{x}_t) = \sum_{m=1}^{M} c_{j,m} \frac{1}{\sqrt{(2\pi)^K |\Sigma|}} \exp\left[-\frac{1}{2}(\mathbf{x}_t - \mu)(\Sigma)^{-1}(\mathbf{x}_t - \mu)^T\right] \tag{6.28}$$

donde $\mathbf{x}_t$ es un vector de características K-dimensional, $c_{j,m}$ son los pesos de la mezcla de M gaussianas para el estado j, y μ y Σ son, respectivamente, el vector medio y la matriz de covarianza de la gaussiana. En el caso particular en que la matriz de covarianza sea diagonal, es posible aplicar el algoritmo WVA con pesos independientes para cada componente del vector de observaciones $x_t(k)$ $(k = 1, \ldots, K)$. Es decir, para cada característica puede indicarse un valor de confianza $\gamma_{k,t}$ independiente, modificando la expresión 6.28 como:

$$b_j(\mathbf{x}_t) = \sum_{m=1}^{M} c_{j,m} \prod_{k=1}^{K} \mathcal{N}\left(x(k); \mu_{j,m}(k), \sigma^2_{j,m}(k)\right)^{\gamma_{k,t}} \tag{6.29}$$

donde $\mathcal{N}\left(x(k); \mu_{j,m}(k), \sigma^2_{j,m}(k)\right)$ representa la función de distribución gaussiana para la k-ésima característica con media $\mu_{j,m}(k)$ y varianza $\sigma^2_{j,m}(k)$.

Pesado de la matriz de covarianzas

Cardenal-Lopez et al. [162, 163] proponen una modificación alternativa en donde, en lugar de ponderar la probabilidad de observación, se ponderan las matrices de covarianza de las gaussianas que la modelan. Es decir, la función de densidad de probabilidad de cada gaussiana multivariada de la mezcla se modifica de la forma,

$$b_j(\mathbf{x}_t) = \sum_{m=1}^{M} c_{j,m} \frac{1}{\sqrt{(2\pi\gamma_t)^K |\Sigma|}} \exp\left[-\frac{1}{2}(\mathbf{x}_t - \mu)(\gamma_t\Sigma)^{-1}(\mathbf{x}_t - \mu)^T\right] \tag{6.30}$$

donde μ y Σ son, respectivamente, el vector medio y la matriz de covarianza de la gaussiana, y γ_t es un vector de confianzas. Si los valores del vector de confianzas son iguales a 1 (plena confianza), la ecuación (6.30) es equivalente a la (6.28). Por el contrario, conforme los valores de este vector decrecen hasta 0, las gaussianas se ensanchan (incrementa la varianza) de forma que la distribución se transforma gradualmente en una función probabilidad uniforme. Como en la aproximación anterior, durante la recepción de vectores completamente no fiables, el algoritmo VA progresa únicamente guiado por las probabilidades de transición.

Aunque comparte similitudes con el WVA, esta técnica tiene como ventaja adicional la posibilidad de aplicar un peso independiente a cada componente aún sin la suposición

de una matriz de covarianza diagonal.

6.4.2. Asignación de valores de confianza

Uno de los mayores problemas que presentan las técnicas basadas en el reconocedor consiste, precisamente, en obtener una función de fiabilidad para los vectores de características. Esto se debe a que definir una fiabilidad para cada vector (o cada característica que lo compone) implícitamente supone cuantificar cómo de útil es dicho vector (o característica) al reconocimiento. En los canales con pérdidas, este problema aparenta tener una fácil solución, ya que los vectores o bien se reciben sin errores, o bien se pierden. Así, una aproximación sencilla consiste en considerar totalmente fiables aquellos vectores que se han recibido y completamente no fiables aquellos que se han perdido, esto es,

$$\gamma_t = \begin{cases} 0 & \text{si } \mathbf{x}_t \text{ se ha perdido} \\ 1 & \text{si } \mathbf{x}_t \text{ se ha recibido} \end{cases} \tag{6.31}$$

Puede observarse que las dos modificaciones del algoritmo VA descritas en la sección anterior tienen el mismo comportamiento frente a este tipo de asignación binaria. Es decir, el algoritmo VA progresa guiado tanto por las probabilidades de transición como las de observación durante las recepciones, y únicamente por las probabilidades de transición durante las ráfagas.

Al explotar un modelo de voz de mayor potencia, es de esperar que las técnicas basadas en el reconocedor ofrezcan mejores resultados que las técnicas de mitigación. Sin embargo, en la práctica, este esquema de asignación binario ofrece peores resultados cuando se producen ráfagas cortas que, por ejemplo, la mitigación del estándar de Aurora (sustitución por el vector recibido más próximo). La tabla 6.11 muestra la precisión en el reconocimiento obtenida aplicando WVA con asignación binaria de confianzas. A fin de permitir comparaciones, se ha empleado el mismo marco experimental y las mismas condiciones de canal que para la evaluación de las técnicas de mitigación del capítulo 5 (tablas 5.1-5.13). Como puede observarse, la técnica WVA obtiene mejores resultados con ráfagas largas, pero, en presencia de ráfagas cortas, incluso las técnicas de mitigación más simples la superan.

Resulta evidente que los resultados obtenidos pueden mejorarse por medio de una aplicación conjunta de técnicas basadas en el receptor y en el reconocedor. En tal caso, no debe aplicarse una asignación binaria ya que, aunque los vectores perdidos pueden

Tasa de pérdidas	*Long. ráfaga*				
	1	*2*	*4*	*8*	*12*
10 %	98.13	97.75	96.69	95.08	94.24
20 %	97.21	96.54	94.40	90.62	89.39
30 %	96.64	95.27	92.13	86.90	83.73
40 %	96.36	94.07	89.54	83.47	78.49
50 %	96.31	92.96	86.80	80.18	74.60

Tabla 6.11: Precisión del reconocimiento (Wacc) empleando WVA con asignación binaria (0 o 1) de confianzas.

considerarse completamente no fiables, no ocurre lo mismo con los vectores que los sustituyen. Debido a la alta correlación de la voz a corto plazo, estas sustituciones pueden resultar útiles para el reconocimiento. Regresamos así al problema original, dar un valor de confianza para estas sustituciones o, lo que es lo mismo, cuantificar como de útil es un vector reemplazado para el reconocimiento.

Por lo general, se está de acuerdo en que resulta beneficioso reducir la fiabilidad de los vectores reemplazados conforme avanzamos en la ráfaga. Esto se debe a que las técnicas de mitigación generalmente asumen que el segmento a recuperar está en un estado estacionario. Sin embargo, cuanto más nos alejamos de la última recepción, más probable es que la señal haya evolucionado a otro sonido y la suposición anterior deje de ser válida. En el caso de que la sustitución esté basada en tramas recibidas antes y tras la ráfaga, puede decirse lo mismo pero conforme nos acercamos al centro de la ráfaga.

Una posible solución, propuesta por Cardenal-Lopez [162], consiste en el uso de una función paramétrica que controle la variación de las fiabilidades. Esta función puede ser lineal,

$$\gamma_t = \begin{cases} 1-(1-\alpha)n, & n = 1,\ldots,B \\ 1-(1-\alpha)(2B-n+1), & n = B+1,\ldots,2B \end{cases} \tag{6.32}$$

o exponencial,

$$\gamma_t = \begin{cases} \alpha^n, & n = 1,\ldots,B \\ 1-\alpha^{(2B-n+1)}, & n = B+1,\ldots,2B \end{cases} \tag{6.33}$$

donde $2B$ es el tamaño de la ráfaga, n es la posición en la ráfaga y α es un valor experimental en el intervalo $[0,1]$. En el caso de obtener valores γ_t negativos, estos se asumen iguales a 0. Mediante pruebas de reconocimiento se muestra que los valores óptimos de α se encuentran en torno a $\alpha = 0{,}1$ para la expresión lineal, y $\alpha = 0{,}7$ para la exponencial [162, 165]. Ambas expresiones no ofrecen diferencias sustanciales cuando se utiliza pesado de la matriz de covarianzas. Sin embargo, cuando se aplica WVA, la expresión (6.33) suele

ofrecer mejores resultados [162].

Bernard y Alwaan proponen para su aproximación WVA un estimador alternativo no basado en funciones paramétricas experimentales [161]. En el esquema propuesto, se asume la repetición de vectores recibidos como técnica de mitigación, estableciéndose las confianzas a partir de la correlación estadística, ya que ésta puede interpretarse como una medida de la dependencia lineal entre variables aleatorias. Para ello, la secuencia de vectores se considera un proceso estocástico con todas las características mutuamente decorreladas (lo que equivale a considerar matrices de covarianza diagonales). Así, la correlación estadística de la secuencia se reduce a un conjunto de K funciones de autocorrelación, $\rho_k(\tau)$, de la forma,

$$\rho_k(\tau) = R_k(\tau)/R_k(0) \tag{6.34}$$

$$R_k(\tau) = E[x_t(k)x_{t+\tau}(k)] \tag{6.35}$$

Estas funciones pueden estimarse asumiendo que cada componente es estacionaria y ergódica (aproximación ruda, pero suficiente para los objetivos que se persiguen), obteniéndose finalmente la fiabilidad de cada componente como,

$$\gamma_{k,t} = \begin{cases} \sqrt{\rho_k(t-t_s)}, & t_s < t \leq (t_e - t_s)/2 \\ \sqrt{\rho_k(t_e - t)}, & (t_e - t_s)/2 < t < t_e \end{cases} \tag{6.36}$$

donde t_s indica el instante de tiempo del último vector recibido antes de la ráfaga, y t_e el del primero tras ella.

Esta aproximación estadística al cálculo de la fiabilidad de los vectores mitigados mediante repetición es la que ofrece mejores resultados, superando a las funciones de fiabilidad paramétricas [163].

Fiabilidad de las componentes dinámicas

El cálculo de la fiabilidad de las componentes dinámicas suele presentar un problema adicional. Generalmente, los sistemas de reconocimiento distribuido no transmiten las características dinámicas, ya que estas pueden calcularse a partir de las estáticas. Como se mostró en la sección 2.2.4, cada derivada temporal se obtiene mediante una ventana deslizante aplicada sobre la derivada temporal de orden inmediatamente inferior. Es decir, las delta-delta características (aceleraciones) se calculan aplicando una ventana sobre las delta características (velocidades) que, a su vez, se obtienen aplicando otra ventana sobre las estáticas. Entonces, si un vector se ha recibido próximo a una ráfaga de vectores

perdidos, las componentes dinámicas asociadas a dicho vector pueden estar afectadas por la pérdida de vectores vecinos. Por esta razón, los valores de confianza de las características dinámicas deben calcularse de forma separada a los de las estáticas [165], obteniéndose los primeros como función aplicada de los valores de confianza de estos últimos. De esta forma, la confianza en las delta componentes vendría dada por,

$$\gamma_t^{\Delta} = f\left(\gamma_{t-W_\Delta}, \gamma_{t-W_\Delta+1}, \ldots, \gamma_t, \ldots, \gamma_{t+W_\Delta-1}, \gamma_{t+W_\Delta}\right) \tag{6.37}$$

mientras que para las delta-delta componentes, por la expresión,

$$\gamma_t^{\Delta\Delta} = f\left(\gamma_{t-W_{\Delta\Delta}}^{\Delta}, \gamma_{t-W_{\Delta\Delta}+1}^{\Delta}, \ldots, \gamma_t^{\Delta}, \ldots, \gamma_{t+W_{\Delta\Delta}-1}^{\Delta}, \gamma_{t+W_{\Delta\Delta}}^{\Delta}\right) \tag{6.38}$$

donde W_Δ es la ventana empleada para las velocidades y $W_{\Delta\Delta}$ la correspondiente a las aceleraciones. La función f puede definirse de múltiples formas.

En el caso en que no se recurra a ninguna técnica de mitigación (se supone, por tanto, una asignación binaria de confianzas), se aplica una asignación *"hard"* en donde la confianza en la componente dinámica se hace 0 si alguna de las componentes estáticas sobre las que se calcula se ha perdido. Sólo en el que caso en que todas se hayan recibido se considera a la componente dinámica totalmente fiable (confianza igual a 1). Esto se debe a que, si no se emplea ninguna técnica de mitigación, no es posible calcular las componentes dinámicas, considerándose perdidas y completamente no fiables [109].

Si por el contrario, se aplica una mitigación sobre los vectores perdidos, pueden emplearse estrategias de asignación *"soft"*. Bernard y Alwaan emplean una asignación muy sencilla, similar a la asignación binaria pero aplicada sobre las componentes dinámicas,

$$\gamma_t^{\Delta}, \gamma_t^{\Delta\Delta} = \begin{cases} 0 & \text{si } \mathbf{x}_t \text{ se ha perdido} \\ 1 & \text{si } \mathbf{x}_t \text{ se ha recibido} \end{cases} \tag{6.39}$$

A pesar de lo rudo del esquema, ya que se consideran totalmente fiables las componentes dinámicas cercanas a las pérdidas (calculadas sobre algunos vectores reemplazados), así como totalmente no fiables las componentes en los extremos de las ráfagas (calculadas sobre algunos vectores recibidos), proporciona buenos resultados, mejorando significativamente al esquema binario anterior [161].

James et al., proponen dos esquemas *soft* con una evolución más suave. El primero de ellos obtiene las confianzas de las componentes dinámicas como el productorio de las

confianzas de las componentes estáticas en la ventana considerada, esto es,

$$\gamma_t^{\Delta} = \prod_{w=-W_{\Delta}}^{-W_{\Delta}} \gamma_{t+w}, \qquad \text{y} \qquad \gamma_t^{\Delta\Delta} = \prod_{w=-W_{\Delta\Delta}}^{-W_{\Delta\Delta}} \gamma_{t+w}^{\Delta}. \tag{6.40}$$

En el segundo esquema, la confianza en la componente dinámica se obtiene aplicando la expresión (2.13) (sección 2.2.4) a las confianzas en las componentes estáticas. Esto es,

$$\gamma_t^{\Delta} = \frac{\sum_{w=-W_{\Delta}}^{-W_{\Delta}} w\gamma_{t+w}}{\sum_{w=-W_{\Delta}}^{-W_{\Delta}} w^2} \qquad \text{y} \qquad \gamma_t^{\Delta\Delta} = \frac{\sum_{w=-W_{\Delta\Delta}}^{-W_{\Delta\Delta}} w\gamma_{t+w}^{\Delta}}{\sum_{w=-W_{\Delta\Delta}}^{-W_{\Delta\Delta}} w^2}. \tag{6.41}$$

Debe notarse que la fórmula regresiva anterior aproxima la derivada de las confianzas en las componentes, no la confianza en la derivada de las componentes. A pesar de esto, en los resultados experimentales presentados en [165] se observa una leve mejora con respecto a los esquemas anteriores.

6.4.3. Reconocimiento ponderado con réplicas VQ

La introducción en el reconocedor de valores de confianza que representen nuestra certeza en una característica decodificada resulta especialmente atractiva para la aplicación de códigos FEC específicos del medio. Mediante estos valores de confianza, sería posible indicar al reconocedor que se dispone de información parcial sobre la trama perdida, esto es, la réplica, pero que debe tener un peso inferior durante el reconocimiento, ya que al estar degradada por un fuerte proceso de cuantización, la confianza en ella debe ser menor. Como se mostró en la sección 6.2.3, incluso con códigos FEC muy cortos, la técnica FB–MMSE proporcionaba una importante mejora de la robustez gracias al uso de un modelo estadístico de voz. Ahora, al tratar las pérdidas durante el propio reconocimiento, un modelo más potente puede ser aprovechado, permitiendo mejores resultados.

Con este objetivo, en esta sección proponemos un esquema mixto de recuperación en el que, junto con los vectores de características, se transmiten códigos FEC específicos que posteriormente son tratados en el reconocedor de forma diferencial mediante valores de confianza. Para ello, recurriremos al esquema de réplicas VQ propuesto en la sección 6.2.2 y al algoritmo de Viterbi ponderado (WVA) descrito en la sección 6.4.1. De esta forma, en el emisor se introducen réplicas VQ de los vectores correspondientes a los instantes de tiempo $t \pm T_{fec}$. En el receptor, estas réplicas permitirán la reconstrucción de algunos vectores perdidos, a los que será necesario asignar un valor de confianza.

Como se ha descrito en la sección 6.4.2, existen diversas heurísticas para la asignación de valores de confianza, siendo la más exitosa la función de autorrelación $\rho_k(\tau)$. Sin embargo, la función de autocorrelación es incapaz de dar valores de confianza coherentes cuando se aplican réplicas VQ. Por ello, proponemos la generalización de la función de autocorrelación hacia la covarianza cruzada normalizada entre la característica original perdida, x, y la estimada, $\tilde{x}$, definida como,

$$\bar{C}[x, \tilde{x}] = C_{x\tilde{x}}/\sigma_x^2 \tag{6.42}$$

donde σ_x^2 es la varianza de la característica y $C_{x\tilde{x}}$ es la varianza cruzada entre x y $\tilde{x}$, definida como,

$$C_{x\tilde{x}} = E[(x - \mu)(\tilde{x} - \tilde{\mu})] \tag{6.43}$$

donde $\mu = E[x]$ y $\tilde{\mu} = E[\tilde{x}]$.

Cuando se define un método de reconstrucción y se asume independencia temporal, la covarianza cruzada puede pre-estimarse empleando la base de datos de entrenamiento. Así, si el método de estimación para una característica, $\tilde{x}_t(k)$, consiste en la repetición de la característica de otro instante de tiempo, esto es, $x_{t+\tau}(k)$, (como en la reconstrucción del estándar Aurora) puede derivarse que la covarianza cruzada normalizada es una función de la distancia temporal τ, obtenida como,

$$\bar{C}[x_t(k), \tilde{x}_t(k)] = \bar{C}[x_t(k), x_{t+\tau}(k)] = \frac{R_k(\tau) - \mu_k^2}{R_k(0) - \mu_k^2} \tag{6.44}$$

Cuando las variaciones globales de una característica son pequeñas en comparación con su valor medio, la función de autocorrelación (ecuación (6.35)) normalmente devuelve valores altos de confianza. Esto es particularmente cierto para los coeficientes energéticos (MFCC0 y LogE), cuyos valores son siempre positivos y su media tiende a ser alta. Esta dependencia se puede evitar por medio de la covarianza normalizada, que elimina la contribución de la media de la característica durante el cálculo de la confianza.

Fiabilidad de las réplicas VQ

Una vez que ciertos vectores son recuperados por medio de las réplicas VQ contenidas en los códigos FEC, los vectores definitivamente perdidos son reemplazados por el vector recibido más próximo (tanto si es un vector SVQ como una réplica VQ). Con este esquema de recuperación la ráfaga reconstruida queda compuesta de réplicas VQ, repeticiones de

réplicas VQ y repeticiones de vectores SVQ. La fiabilidad para todas estas características puede obtenerse como casos particulares de la ecuación (6.42) como sigue:

- Cuando las características SVQ se repiten, sus fiabilidades se obtienenen de igual forma que en la ecuación (6.44),

$$\bar{C}[x_t(k), \tilde{x}_t(k)] = \bar{C}[x_t(k), x_{t+\tau}(k)] \equiv \bar{C}_{SVQ}(\tau; k) \tag{6.45}$$

- En cualquier otro caso, la característica recuperada $\tilde{x}_t(k)$ es la repetición de una réplica VQ τ instantes de tiempo anteriores o posteriores, o es la propia réplica ($\tau = 0$), esto es,

$$\tilde{x}_t(k) = VQ[x_{t+\tau}(k)] \tag{6.46}$$

  de forma que,

$$\bar{C}[x_t(k), \tilde{x}_t(k)] = \bar{C}[x_t(k), VQ[x_{t+\tau}(k)]] \equiv \bar{C}_{VQ}(\tau; k) \tag{6.47}$$

  donde $VQ[]$ representa la cuantización vectorial.

Finalmente, un estimador de la función de fiabilidad puede definirse de forma análoga a la expresión (6.36) como,

$$\gamma_{k,t} = \begin{cases} \sqrt{\bar{C}_{SVQ}(\tau_1; k)} & \text{cuando } \tilde{x}_t(k) = x_{t+\tau_1}(k) \\ \sqrt{\bar{C}_{VQ}(\tau_2; k)} & \text{cuando } \tilde{x}_t(k) = VQ[x_{t+\tau_2}(k)] \end{cases} \tag{6.48}$$

Generalmente, la covarianza cruzada normalizada decae rápidamente. Por ello, sólo es necesario calcular y almacenar su valor para unos pocos valores de τ, los restantes pueden asumirse iguales a 0.

La figura 6.8 muestra un ejemplo de la evolución de la función de fiabilidad, para el MFCC de orden 1, durante una ráfaga que comienza en S_b y acaba en E_b y en la que se reciben algunas réplicas VQ de 8 bits (instantes marcados por R). Como puede observarse, la fiabilidad en la observación decrece conforme nos alejamos del último y primer vector recibido, respectivamente, antes y tras la ráfaga. Sin embargo, previamente a la aparición de una réplica, cuando se emplean repeticiones de ésta, la fiabilidad se recupera, alcanzando un pico en la propia réplica. El valor de este pico estará en consonancia con el tamaño del diccionario empleado para la cuantización de la réplica, siendo más o menos próximo a 1 dependiendo el número de bits considerado.

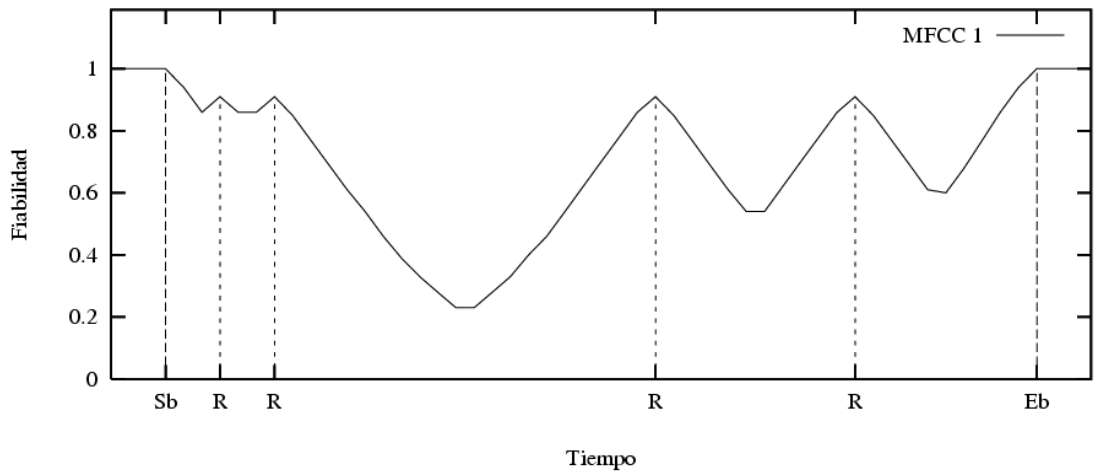


Figura 6.8: Ejemplo de la evolución de la fiabilidad del MFCC(1) en el tiempo cuando se aplican réplicas VQ (R) dentro de una ráfaga (desde S_b a E_b).

Por último, es posible que reemplazar una pérdida con una copia del vector SVQ más cercano sea mejor que sustituirla por una copia de una réplica VQ, o que incluso por la propia réplica. Esto es particularmente cierto cuando las réplicas están fuertemente cuantizadas y están disponibles al principio o al final de una ráfaga. En esos instantes, dada la alta correlación de la señal a corto plazo, la repetición del vector SVQ más próximo puede ser más fiable que el reemplazo por una réplica. Puesto que disponemos de un mecanismo que nos permite conocer la bondad de cada opción, esto es, la función de fiabilidad, podemos seleccionar aquel método que arroje una mayor fiabilidad al reemplazar la característica perdida. Es decir,

- si la sustitución de una pérdida por la característica SVQ más próxima arroja una mayor fiabilidad que la sustitución por la correspondiente réplica (en caso de que esté disponible), se aplicará la repetición SVQ.

- si la réplica no esta disponible y la sustitución de una pérdida por la característica SVQ más cercana tiene mayor fiabilidad que la sustitución por la réplica más próxima, se aplicará la repetición SVQ.

- en cualquier otro caso se empleará la réplica correspondiente, si está disponible, o la repetición de la réplica VQ más próxima, en su defecto.

Como puede concluirse, la aplicación de este esquema de selección permite la maximización de la fiabilidad de las observaciones,

$$\gamma_{k,t} = max\left\{\sqrt{\bar{C}_{SVQ}(\tau_1;k)}, \sqrt{\bar{C}_{VQ}(\tau_2;k)}\right\} \tag{6.49}$$

6.4.4. Resultados experimentales

La técnica de reconocimiento ponderado con réplicas VQ, descrita en la sección anterior, ha sido evaluada en una arquitectura DSR sobre un canal con pérdidas. El marco experimental en el que se realizan las pruebas coincide con el descrito en la sección 6.2.4. Esto es, mediante el front-end básico estandarizado por ETSI [74] se extraen los vectores de características que posteriormente serán empaquetados. Este empaquetado se realiza conforme al RFC 3557 (formato de carga útil DSR [79]), pero añadiendo ahora 8 bits adicionales por paquete. Estos bits adicionales permiten, junto con la reutilización de los 4 bits dedicados al CRC y los 4 bits de relleno, la introducción de dos réplicas de hasta 8 bits. De acuerdo con las recomendaciones del estándar, sólo se transmitirá un único par de vectores SVQ por paquete.

La simulación de las pérdidas de paquetes se realiza a través del modelo de tres estados y bajo las 5 condiciones de canal ya descritas en la sección 6.2.4. A fin de facilitar la lectura, la tabla 6.12 muestra de nuevo los parámetros empleados para generar las condiciones, así como los resultados obtenidos, en estas condiciones, por la técnica de mitigación del estándar y el algoritmo de reconocimiento WVA empleando la covarianza normalizada como estimación de la fiabilidad de las características estáticas (para las dinámicas se emplea la aproximación de Alwaan descrita en la ecuación (6.39)).

Las tablas 6.13, 6.14 y 6.15 muestran los resultados obtenidos por el esquema propuesto de reconocimiento ponderado (algoritmo WVA con fiabilidad basada en covarianza normalizada) con réplicas VQ de 2, 4 y 8 bits, respectivamente. Puede describirse una evolución similar a la observada cuando las réplicas se aplican de forma directa o con estimación FB–MMSE (tablas 6.2-6.7), esto es, una mejora de los resultados tanto al incrementar la latencia (un mayor número de paquetes puede insertar réplicas en la ráfaga) como al emplear diccionarios mayores, aunque en este caso los resultados son claramente mejores. En todas las condiciones y para todas las latencias, los resultados obtenidos mejoran también a aquellos correspondientes a la técnica WVA sin réplicas, y superan ampliamente los proporcionados por la técnica estándar.

	Parámetros			*Mitigación*	
Condición	R_{loss}	L_{loss}	L_{recep}	Aur	WVA
1	10 %	2	4	98.47	98.60
2	20 %	4	4	93.57	96.30
3	30 %	6	3	85.47	91.97
4	40 %	8	3	76.51	86.28
5	50 %	10	2	65.71	78.87

Tabla 6.12: Parámetros del modelo de 3 estados para cada condición y resultados de referencia obtenidos por Aurora (Aur.) y WVA con covarianza normalizada (WVA).

	Latencia (ms)						
Cond.	120	160	240	320	400	480	600
1	98.67	98.67	98.66	98.75	98.74	98.78	98.65
2	96.85	96.60	96.99	96.81	97.16	96.93	96.83
3	93.06	93.48	93.69	93.46	93.92	94.12	93.78
4	87.82	88.60	88.81	88.98	89.18	89.79	89.89
5	80.52	81.53	82.23	82.68	82.29	82.00	83.10

Tabla 6.13: Resultados obtenidos mediante la aplicación conjunta de réplicas VQ de 2 bits y reconocimiento ponderado WVA.

	Latencia (ms)						
Cond.	120	160	240	320	400	480	600
1	98.75	98.74	98.73	98.75	98.80	98.81	98.69
2	97.57	97.51	97.83	97.73	97.86	97.93	97.79
3	94.63	95.18	95.69	95.75	96.11	96.50	96.56
4	90.06	90.96	91.83	92.67	93.24	93.61	94.11
5	83.88	84.91	86.26	87.43	87.65	87.76	89.17

Tabla 6.14: Resultados obtenidos mediante la aplicación conjunta de réplicas VQ de 4 bits y reconocimiento ponderado WVA.

	Latencia (ms)						
Cond.	120	160	240	320	400	480	600
1	98.80	98.81	98.78	98.84	98.88	98.84	98.79
2	97.85	97.92	98.24	98.18	98.34	98.32	98.22
3	95.52	96.29	96.70	97.12	97.34	97.29	97.68
4	91.86	92.70	93.92	94.54	95.24	95.51	96.04
5	86.33	87.77	89.14	90.50	90.96	91.21	92.42

Tabla 6.15: Resultados obtenidos mediante la aplicación conjunta de réplicas VQ de 8 bits y reconocimiento ponderado WVA.

6.5. Combinación de las técnicas en el emisor. Esquema de doble flujo

Como se ha mostrado a lo largo de este capítulo, la aplicación combinada de técnicas basadas en el emisor, en el receptor y en el reconocedor permite un aumento significativo de la robustez en el reconocimiento frente a canales con pérdidas. En este punto cabría preguntarse qué técnica propuesta basada el en emisor resulta más apropiada. Por un lado, el uso de réplicas implica un ligero incremento del ancho de banda necesario (que puede no llegar a ser efectivo) así como un aumento en la latencia de la comunicación. Por otro lado, el entrelazado únicamente causa un incremento de la latencia, sin modificar la tasa de bits a transmitir.

Sin embargo, esta comparación entre técnicas puede resultar innecesaria ya que, como técnicas independientes, no hay razón por la cual los códigos FEC y el entrelazado propuestos no pudieran aplicarse al mismo tiempo. El problema asociado radica en que, si las operaciones de entrelazado y FEC se componen de forma directa, los retrasos (y las latencias) introducidos por ambas técnicas se suman. Así, si los paquetes se entrelazan después de que las réplicas VQ hayan sido calculadas, al retraso introducido por los códigos FEC se le añade el correspondiente al entrelazado. Lo mismo ocurre en el caso contrario, ya que se tendrían que tomar réplicas a $\pi(t) \pm T_{fec}$ tramas anteriores y posteriores a la pareja de vectores entrelazados.

En esta sección proponemos un esquema alternativo por medio del cual se consigue evitar la suma de los retrasos de ambas operaciones. En este sencillo esquema se consideran dos flujos independientes de vectores que, posteriormente, se multiplexarán en los paquetes. El primer flujo está compuesto por vectores de características cuantizados con SVQ, de forma análoga al estándar DSR. El segundo flujo, en cambio, se compone de los mismos vectores pero cuantizados con VQ. Como puede observarse, el segundo flujo no es más que un flujo virtual redundante insertado en los paquetes gracias a las réplicas VQ. Inicialmente, los vectores de ambos flujos (SVQ y VQ) están ordenados de acuerdo a su correspondiente instante temporal. Entonces, una función de entrelazado diferente se aplica sobre cada uno de los flujos. La figura 6.9 muestra el esquema propuesto. Al ser flujos independientes, ambos entrelazados se puede aplicar paralelamente, resultando en que la latencia total introducida en la transmisión es igual a la máxima de sus latencias.

Este esquema abre un amplio abanico de posibilidades, ya que ahora dos entrelazadores, uno primario y otro secundario, deben escogerse de forma que se maximice la

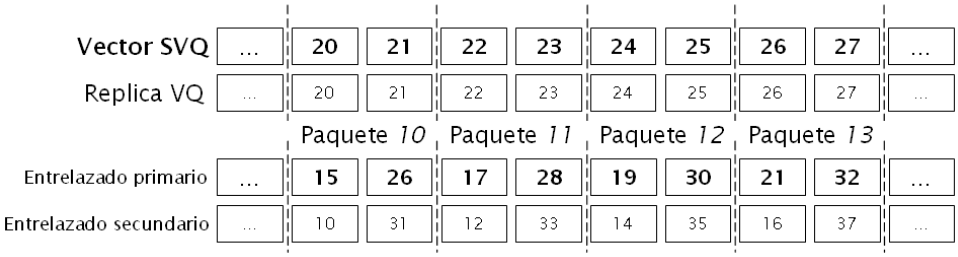


Figura 6.9: Diagrama de ejemplo de transmisión con doble flujo con vectores SVQ y réplicas VQ multiplexadas en los paquetes como códigos FEC.

dispersión de las pérdidas, no sólo entre los vectores recibidos, sino también entre las réplicas. Un problema relevante es el de la aparición de *réplicas solapadas*, es decir, ciertas combinaciones de entrelazadores dan lugar a que algunas réplicas acaben siendo transmitidas en el mismo paquete que los vectores a los que protegen, resultando completamente inútiles (si se pierde el vector, también se pierde la réplica). Adicionalmente, a causa de la naturaleza redundante de las réplicas, la función de entrelazado definida para éstas no tiene por que ser ser estrictamente una permutación. Podrían entonces descartarse ciertas réplicas en favor de que otras se introduzcan varias veces.

Como puede observarse, la elección de la pareja de entrelazadores no es una tarea trivial. En este trabajo hemos preferido no profundizar en este problema, dejando estas cuestiones como posible línea de investigación futura. Así, para la reordenación de los vectores SVQ recurriremos al entrelazado más potente de que disponemos, es decir, el entrelazado $(2, t)$ de tipo Ramsey (sección 6.3.3), mientras que para las réplicas escogeremos uno que tan sólo evite que queden solapadas.

La organización propuesta en el esquema VQ inicial (véase sección 6.2.2) no resulta válida para esta tarea ya que, de aplicarse, todas las réplicas quedarían solapadas. Esto se debe a que dicho esquema es un caso particular del esquema de doble flujo, en donde los vectores SVQ no están entrelazados y la organización de las réplicas (tomadas a $\pm T_{fec}$ tramas del paquete actual) es equivalente a la obtenida empleando un entrelazador $(2, 2B+1)$ de Ramsey, en donde $B = T_{fec}$. Es decir, estaríamos utilizando el mismo entrelazador para los vectores SVQ y VQ. Otro candidato podría ser el entrelazado MLBI, pero una evaluación detallada de este entrelazador revela la aparición de réplicas solapadas en ciertos instantes de tiempo.

Por la simplicidad que ello conlleva, hemos optado por emplear un entrelazador convolucional de Ramsey de baja dispersión para las réplicas VQ. Diversas pruebas experimen-

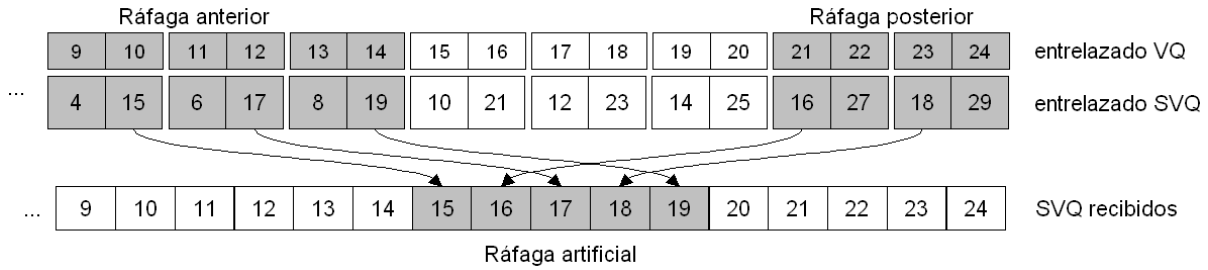


Figura 6.10: Ejemplo de ráfaga artificial sobre el flujo principal causada por el entrelazado convolucional $(2, t)$. Las réplicas VQ no entrelazadas permiten su recuperación.

tales indican que no entrelazar las réplicas, es decir, transmitirlas en su orden original, podría resultar beneficioso. Esto se debe a que, aunque suele ser poco habitual, la misma reordenación que hace que se dispersen las pérdidas, puede reconstruir una ráfaga a partir de otras distantes. Como ejemplo, la figura 6.10 muestra cómo un entrelazador primario da lugar a una ráfaga artificial en el flujo de vectores SVQ a causa de dos ráfagas distintas. Puede observarse que si las réplicas no están entrelazadas, la ráfaga puede reconstruirse completamente. Un entrelazado de baja dispersión puede resultar aún más útil, ya que permite la fragmentación de ráfagas artificiales moderadamente más largas.

6.5.1. Resultados experimentales

Tomando como técnica de reconstrucción el reconocimiento ponderado WVA, podemos evaluar y comparar los resultados alcanzados mediante la técnica de entrelazado más potente propuesta, el entrelazado convolucional de Ramsey $(2, t)$ (sección 6.3.3), y el esquema FEC desarrollado basado en réplicas VQ. Los resultados de la primera técnica se muestran en la tabla 6.16, mientras que los obtenidos por los esquemas FEC pueden consultarse en las tablas 6.13, 6.14 y 6.15 de la sección 6.4.4. Como puede observarse, la técnica de entrelazado arroja mejores resultados que el esquema FEC específico con 2 y 4 bits. En cambio, es superada por este esquema cuando las réplicas VQ están codificadas con, al menos, 8 bits. En vista de estos resultados, el uso de réplicas VQ de 2 y 4 bits resultaría inútil, ya que el entrelazado, que además no supone un incremento del ancho de banda, proporciona un mayor rendimiento.

Sin embargo, ambas técnicas no tienen porqué competir, ya que pueden combinarse y proporcionar mejores resultados que los obtenidos mediante su aplicación independiente. Con este propósito se recurrirá a un esquema de doble flujo, evitando un incremento en la latencia. En dicho esquema se aplicará el entrelazado convolucional de Ramsey $(2, t)$

	Latencia (ms)						
Cond.	120	160	240	320	400	480	600
1	98.90	98.99	98.96	98.98	98.96	98.95	98.93
2	98.15	98.24	98.47	98.55	98.65	98.59	98.62
3	95.86	96.37	96.93	97.18	97.31	97.36	97.66
4	92.11	93.02	93.93	94.37	94.92	95.10	95.20
5	86.35	87.00	87.57	88.94	89.42	89.88	90.03

Tabla 6.16: Resultados obtenidos empleando entrelazado convolucional de Ramsey $(2, t)$ y reconocimiento ponderado WVA con covarianza normalizada.

	Latencia (ms)						
Cond.	120	160	240	320	400	480	600
1	98.90	98.99	98.96	98.98	98.96	98.95	98.93
2	98.16	98.24	98.54	98.64	98.69	98.64	98.71
3	95.93	96.56	97.14	97.52	97.61	97.75	97.83
4	92.21	93.50	94.49	95.14	95.60	95.96	95.63
5	86.46	87.84	88.94	90.48	90.89	91.44	90.95

Tabla 6.17: Resultados obtenidos empleando entrelazado convolucional de Ramsey $(2, t)$, réplicas VQ de 2 bits y reconocimiento ponderado WVA con covarianza normalizada.

	Latencia (ms)						
Cond.	120	160	240	320	400	480	600
1	98.90	98.98	98.96	98.97	98.96	98.96	98.93
2	98.24	98.37	98.63	98.72	98.84	98.70	98.75
3	96.02	96.81	97.47	97.78	98.00	98.12	98.09
4	92.44	93.82	95.10	95.89	96.43	96.90	96.24
5	86.91	88.61	90.18	91.82	92.66	93.42	92.19

Tabla 6.18: Resultados obtenidos empleando entrelazado convolucional de Ramsey $(2, t)$, réplicas VQ de 4 bits y reconocimiento ponderado WVA con covarianza normalizada.

	Latencia (ms)						
Cond.	120	160	240	320	400	480	600
1	98.90	98.98	98.94	99.00	98.97	98.93	98.93
2	98.38	98.42	98.69	98.83	98.88	98.79	98.80
3	96.20	96.99	97.60	98.00	98.35	98.32	98.23
4	92.78	94.19	95.44	96.41	96.85	97.31	96.68
5	87.70	89.44	91.28	92.97	93.61	94.33	93.20

Tabla 6.19: Resultados obtenidos empleando entrelazado convolucional de Ramsey $(2, t)$, réplicas VQ de 8 bits y reconocimiento ponderado WVA con covarianza normalizada.

tanto para los vectores SVQ como para las réplicas VQ. La única diferencia entre ambos entrelazadores será el parámetro t elegido. Para el caso del entrelazador primario, será el máximo permitido por la latencia, es decir, $t_1 = l - 1$, donde l está expresado en número de tramas. Para el segundo, en cambio, se establecerá en $t_2 = l/4 - 1$, proporcionando una dispersión significativamente inferior. El objetivo, como se justificó anteriormente, es que el entrelazador secundario fragmente las posibles ráfagas generadas por el entrelazado primario. Puesto que la latencia del entrelazador secundario es siempre inferior, la latencia total viene dada por la latencia del primario (el máximo de ambas).

Las tablas 6.17, 6.18 y 6.19 muestran los resultados obtenidos según este esquema. Como puede observarse, la aplicación conjunta de ambas técnicas proporciona mejores resultados que las mismas técnicas por separado. En general, se repite el mismo patrón observado en sección 6.2.4, es decir, cuanto mayor sea el tamaño de la réplica y/o la latencia permitida, mayor es la precisión alcanzada durante el reconocimiento. Si bien, en este caso, cabe notar que las réplicas no dan lugar a un fuerte incremento del rendimiento. Esto se debe a que, en los esquemas anteriores, las réplicas no sólo permitían recuperar tramas sino que además causaban una fragmentación de las ráfagas. Ahora, dicha fragmentación la realiza el entrelazador principal (esto, además, podría sustentar la hipótesis de que no es necesario un entrelazador de alta dispersión para el flujo secundario). Así, la mejora obtenida radica principalmente en la recuperación de información perdida por parte de las réplicas.

6.6. Resumen de resultados y conclusiones

En este capítulo hemos propuesto diversas técnicas de recuperación basadas en el emisor y en el reconocedor cuyo objetivo es la mejora de la robustez frente a canales con pérdidas. La figura 6.11 muestra un resumen de los resultados obtenidos por medio de estas técnicas. Para su construcción se ha establecido una latencia máxima de 240 ms. Aunque esté lejos de los 500 ms permitidos antes de que se degrade la interacción oral, debe tenerse en cuenta la latencia propia que puede introducir la red. De cara a los algoritmos propuestos, esta latencia implica un T_{fec} de 12 tramas, en caso de usar un esquema FEC, y un valor para t igual a 23 tramas, para entrelazadores $(2, t)$ de Ramsey. Adicionalmente, se ha considerado una reserva de espacio adicional en cada paquete para alojar códigos FEC de hasta 4 bits. Como se mostró en la sección 6.2.5, este espacio puede obtenerse reutilizando los bits dedicados al CRC y al relleno, sin implicar un incremento efectivo en la tasa de bits.

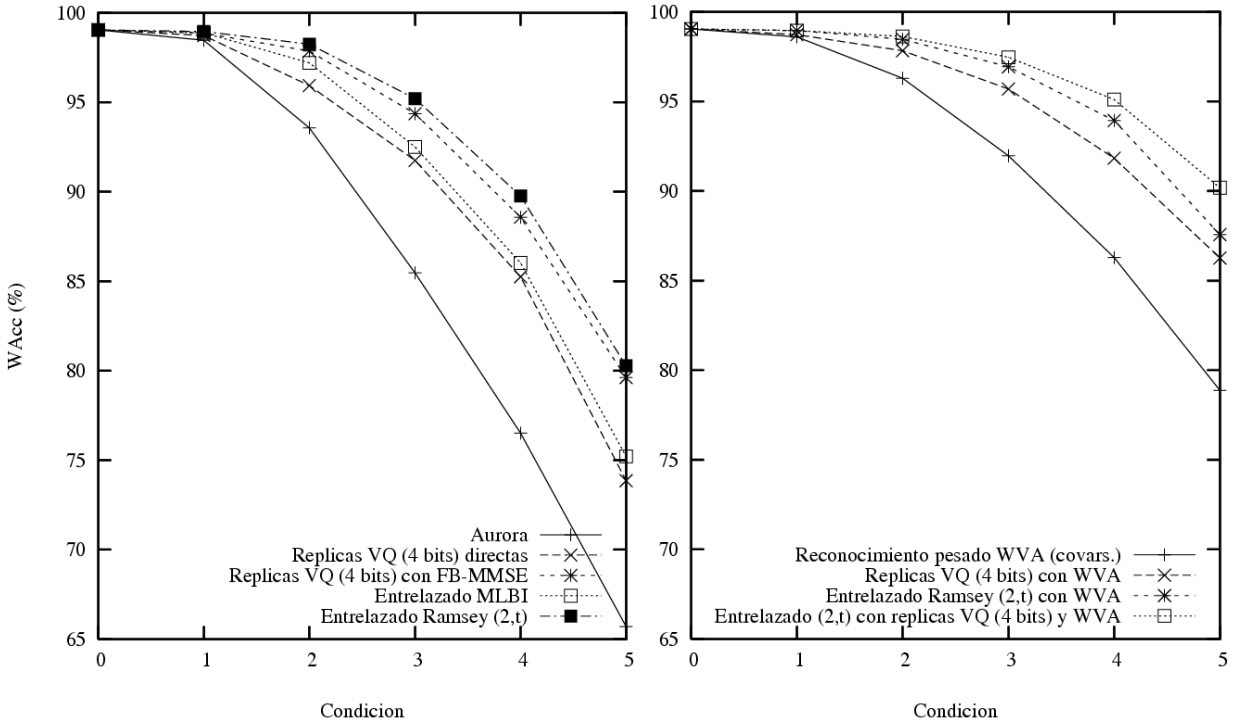


Figura 6.11: Comparación de las técnicas basadas en emisor y en el reconocedor propuestas considerando una latencia máxima permitida de 240 ms y 8 bits adicionales por paquete.

Aunque las técnicas de mitigación tienen la ventaja de aplicarse en el receptor, presentan limitaciones cuando se enfrentan a ráfagas largas, en donde es probable que la voz haya evolucionado de un fonema a otro. Se hace entonces necesario la ayuda de técnicas aplicadas en otras etapas del sistema de reconocimiento remoto. Las técnicas basadas en el emisor son técnicas fundamentalmente preventivas que, como su nombre indica, implican la colaboración del emisor. En este sentido su aplicación queda limitada a aquellas arquitecturas y redes en las que sea posible la modificación de este elemento de la transmisión. En este capítulo examinamos dos tipos de técnicas basadas en el emisor: los códigos FEC específicos del medio y el entrelazado de tramas. En ambos casos se persigue la fragmentación de las ráfagas en otras más cortas, en el primero por medio de la introducción de información redundante en los paquetes (que además permite una reducción del porcentaje de vectores perdidos), mientras que en el segundo mediante una reordenación de los vectores antes de ser transmitidos. Como se ha justificado en múltiples ocasiones a lo largo de este capítulo, esto se debe a que una reducción en la longitud promedio de las ráfagas tiene un notable efecto positivo en el reconocimiento.

Mediante un sencillo esquema FEC específico del medio en el que, junto a los dos

vectores SVQ correspondientes a cada paquete, se envían dos réplicas VQ de vectores distanciados en el tiempo por $\pm T_{fec}$ tramas, es posible mejorar sustancialmente el rendimiento del reconocedor. Sin embargo, a fin de minimizar el incremento en la tasa de bits, estas réplicas han de codificarse con muy pocos bits. Esta fuerte codificación puede ocasionar que la información transmitida llegue muy degradada, y derivar en que una aplicación directa de los códigos FEC resulte incluso perjudicial para el proceso de reconocimiento. Por ello, en este capítulo proponemos una técnica específicamente adaptada para su uso con réplicas VQ, la estimación FB–MMSE. Mediante esta estimación, un sencillo modelo estadístico de la voz permite mejorar la información contenida en las réplicas, aún estando muy degradada. En la figura 6.11 se muestran las sucesivas mejoras obtenidas al aplicar réplicas VQ y estimación FB–MMSE con réplicas VQ.

En relación al entrelazado, este capítulo propone la aplicación de entrelazadores convolucionales de Ramsey al reconocimiento distribuido. Un análisis matemático de los entrelazadores evidencia que dos parámetros, s y t, controlan el comportamiento del entrelazador frente a las ráfagas de vectores perdidos. Mientras que s permite dispersar más las pérdidas aisladas, t permite contrarrestar ráfagas más largas. Desafortunadamente, cuanto más altos sean estos dos parámetros, más latencia en la comunicación causa el entrelazado, por lo que ha de establecerse un compromiso entre ambos. Los entrelazadores de bloque con latencia mínima (MLBI) asumen $s = t$, sin embargo, atendiendo a las técnicas de mitigación aplicadas en el receptor, tal dispersión de las pérdidas aisladas resulta excesiva. Por el contrario, los entrelazadores descritos por Ramsey permiten especificar ambos parámetros de forma independiente, aunque bajo cierta condición de primalidad. En este trabajo se propone un entrelazador de este tipo en donde $s = 2$ y t se establece tan alto como permita la latencia ($t = l - 1$). Como resultado, incurriendo en la misma latencia de comunicación, el entrelazador propuesto obtiene mejores resultados que los proporcionados por el entrelazado MLBI (figura 6.11). Esto se verifica incluso cuando se aplican técnicas más avanzadas que la propuesta por el estándar de Aurora, como las estimaciones MMSE de modelo reducido y MAP, las cuales pueden hacer uso de más de un vector entre las ráfagas ($s > 2$).

Además de contar con la colaboración del emisor para prevenir las pérdidas, o al menos, para distribuirlas de forma favorable, una aproximación alternativa a la recuperación consiste en el tratamiento de las pérdidas en el propio reconocedor, aprovechando así el potente modelo de voz presente en él. Para ello, se recurre a modificaciones en los algoritmos de reconocimiento tales que permitan especificar valores de confianza junto a las observaciones. De esta forma, es posible llevar a cabo el reconocimiento sobre datos

no fiables o, como en nuestro caso, incompletos. Este tipo de técnicas resultan especialmente atractivas para la aplicación de réplicas VQ. Mediante estos valores de confianza, es posible indicar al reconocedor que se dispone de información sobre la trama perdida pero nuestra confianza en ella es menor, ya que está degradada por un fuerte proceso de cuantización.

En este capítulo hemos propuesto un esquema de asignación de confianzas basado en la covarianza cruzada normalizada. Gracias a este esquema, es posible informar a un reconocedor basado en el algoritmo ponderado de Viterbi (WVA) de nuestra confianza en las réplicas de una forma coherente. Es decir, conforme un vector recibido (o una réplica) es repetido como sustitución a un vector perdido, nuestra confianza en él decrece, ya que la voz puede haber evolucionado a otro sonido. Igualmente, la confianza en una réplica decrece conforme ésta es cuantizada con menos bits (aumenta la distorsión de codificación). Por medio de este esquema, es posible mejorar sustancialmente la precisión en el reconocimiento con muy pocos bits por réplica (la figura 6.11 muestra los resultados con 4 bits/réplica).

Además de combinarse con esquemas FEC, el tratamiento mediante WVA puede aplicarse conjuntamente con el entrelazado. En el caso de usarse un entrelazador de Ramsey con dispersión $(2, t)$ los resultados obtenidos de esta combinación mejoran incluso a los del esquema anterior con réplicas VQ y WVA (figura 6.11). Sin embargo, los esquemas FEC y el entrelazado son técnicas independientes, siendo posible combinarlos y alcanzar un mayor rendimiento en el reconocimiento. Así, en este capítulo proponemos un esquema de doble flujo en el que ambas técnicas son aplicadas conjuntamente sin provocar un incremento de las latencias (una composición directa de ambas operaciones ocasiona una latencia total igual a la suma de la latencia de ambas operaciones). Este esquema proporciona los mejores resultados obtenidos hasta el momento. Como puede observarse en la figura 6.11, para una condición de canal con una tasa de pérdidas del 50 % y ráfagas de 10 paquetes podría alcanzarse una precisión en el reconocimiento del 90.18 % asumiendo tan sólo una pequeña latencia en la comunicación de 240 ms y un incremento del ancho de banda de 8 bits por paquete (que puede evitarse reutilizando los bits de CRC y de relleno).

Capítulo 7

CONCLUSIONES

El interés del presente trabajo se centra en el análisis y tratamiento de las degradaciones causadas por el canal de transmisión sobre el rendimiento de un sistema de reconocimiento remoto de voz, concretamente cuando éste se realiza sobre redes móviles GSM y redes de conmutación de paquetes basadas en IP. En ambos casos la voz debe atravesar un canal propenso a errores, dando lugar a degradaciones que afectan a la precisión en el reconocimiento del habla. En este trabajo se proponen diferentes técnicas orientadas a evitar, reducir y compensar dichas degradaciones.

7.1. Conclusiones

- Hemos descrito el estado actual del reconocimiento remoto de la voz, indicando las soluciones disponibles para llevarlo a cabo y sus correspondientes ventajas y limitaciones. Aunque DSR es la arquitectura más potente, ya que está diseñada específicamente para el reconocimiento remoto del habla, su aplicación en GSM requiere cambios hardware en los actuales terminales móviles y modificaciones en la red. Por contra, las redes IP no presentan este problema gracias a su flexibilidad para la transmisión de medios diversos. Así, dos soluciones resultan completamente factibles actualmente, NSR operando sobre GSM y DSR en redes IP.

- Ambas soluciones pueden analizarse de forma unificada asumiendo un canal subyacente con pérdidas, en donde fragmentos completos de información o bien se pierden por congestión de la red (en IP), o bien se descartan por estar seriamente dañados (en GSM). Dependiendo de la arquitectura empleada, DSR o NSR, y por ende, de la codificación de voz aplicada, predictiva o no, además de la consecuente pérdida de

información puede aparecer una degradación adicional a la que nos referimos como ruido de memoria.

- Las memorias de largo retardo, presentes en los codificadores de voz de tipo CELP (como EFR y AMR, en GSM), son el origen de un ruido adicional y posterior a la pérdida de tramas. Este ruido de memoria tiene un efecto negativo importante sobre la precisión en el reconocimiento y es inherente al propio proceso de síntesis de voz, en donde, para la reconstrucción de la trama actual, se requieren muestras de la trama anterior. Si la trama anterior se ha descartado, la síntesis de varias tramas posteriores a ella conducirá a errores, aunque dichas tramas se reciban correctamente.

- El efecto de este ruido sobre el dominio del cepstrum y el espectro logarítmico de la señal puede modelarse mediante factores aditivos (de forma similar al ruido acústico). Gracias a esto, el ruido de memoria puede tratarse directamente sobre los vectores de características obtenidos a partir de la señal de voz decodificada. Puesto que podemos generar cuanta información en estéreo necesitemos, hemos propuesto una adaptación de la técnica FCDCN para la compensación de este ruido (A-FCDCN). Esta técnica no solo resulta efectiva para mitigación del ruido de memoria, sino que además puede aliviar ligeramente la degradación debida a la codificación de la voz.

- Alternativamente, el ruido de memoria puede suprimirse casi en su totalidad a través de la transparametrización de la voz codificada. En este trabajo hemos propuesto dos nuevos transparametrizadores, uno para codec EFR y otro para el conjunto de codecs adaptativo de AMR. Gracias a la transparametrización, los MFCCs de orden 1 al 12 pueden obtenerse sin errores en las tramas posteriores a las perdidas, evitando el ruido de memoria y mejorando la precisión en el reconocimiento. Los restantes parámetros (MFCC0 y LogE), al depender de la energía de la excitación, están inevitablemente contaminados con ruido de memoria. Sobre ellos puede aplicarse la técnica A-FCDCN.

- Además de resultar muy efectiva para evitar la aparición del ruido de memoria, la transparametrización puede resulta beneficiosa contra la distorsión introducida por AMR al reducir la tasa de bits.

- El protocolo TFO permite realizar la transparametrización de la voz sin realizar cambios ni en el terminal de usuario ni en la red GSM. Una actualización con este protocolo, además de mejorar la calidad de la voz (al evitar el funcionamiento en tandem), habilitaría un reconocimiento remoto de calidad con una tasa de reconocimiento cercana e incluso superior a DSR sin necesidad de añadir hardware nuevo al terminar móvil del usuario. Además, esta actualización puede realizarse en aquellas partes terminales de la red donde, por su localización, sea interesante ofrecer servicios de reconocimiento.

- Por otra parte, la degradación derivada de la propia pérdida de información (pérdida de vectores de características) puede tratarse en el receptor mediante técnicas basadas en la estimación MMSE. Las estimaciones obtenidas por estas técnicas pueden mejorarse si se consideran modelos más ricos de evolución temporal de la voz. Sin embargo, los requerimientos de memoria de estos modelos crecen de forma exponencial al considerar órdenes mayores. Mediante el establecimiento de un umbral de frecuencia mínima de aparición es posible reducir de forma drástica estos requerimientos, a la vez que proporcionar una mejora significativa en la precisión del reconocimiento. Como contrapartida es necesaria una técnica auxiliar para ciertas reconstrucciones.

- Alternativamente, el efecto de las pérdidas puede mitigarse suponiendo que la secuencia completa de vectores de características es la salida de un proceso aleatorio estacionario con una distribución gaussiana, siendo los vectores perdidos estimados maximizando su probabilidad condicionada a los valores de los vectores observados (recibidos). La enorme complejidad computacional de esta estimación basada en máximo a posteriori, puede reducirse significativamente limitando las observaciones a una ventana centrada sobre la pérdida en la misma banda cepstral. Gracias a ello, es posible realizar la estimación MAP de forma eficiente y proporcionar una mejora significativa en la precisión del reconocimiento.

- Las técnicas basadas en estimación MMSE y MAP proporcionan resultados similares pero tienen requerimientos contrapuestos. Mediante la introducción de la estimación MAP, como técnica auxiliar, en las técnicas de estimación MMSE con umbral de frecuencia (modelo reducido), es posible balancear los requerimientos computacionales y de memoria entre ambas técnicas.

- Las técnicas de mitigación tienen la ventaja de requerir únicamente la colaboración del receptor, siendo especialmente útiles para la arquitectura NSR. Estas técnicas ofrecen buenos resultados gracias a la fuerte similitud que presenta la voz consigo misma a corto plazo. Sin embargo, por esta misma razón, están limitadas a pérdidas consecutivas (o ráfagas) cortas. En este sentido el rendimiento del sistema puede mejorarse aplicando conjuntamente técnicas basadas en emisor.

- La pérdida consecutiva de vectores de características es significativamente más degradante para el reconocimiento que la pérdida de vectores aislados. Por ello, una reducción en la longitud promedio de la ráfaga conduce a una mejora significativa en el reconocimiento. A fin de lograr una fragmentación de las ráfagas se puede recurrir a la introducción en el emisor de códigos FEC específicos, o réplicas, sobre el bitstream.

- Sin embargo, la introducción de réplicas resulta contraproducente si están fuertemente cuantizadas. En dicho caso se hace necesario aplicar una técnica de post-procesado que maneje correctamente la información contenida en las réplicas. Así, parte del éxito de los esquemas basados en réplicas depende del tratamiento que se aplique sobre éstas en el receptor.

- La estimación FB-MMSE resulta muy efectiva para tratar las réplicas degradas en el receptor. En este caso puede establecerse un modelo de canal en el que los vectores "recibidos", esto es, aquellos recuperados gracias a las réplicas, se consideran afectados por ruido de cuantización. Con la ayuda de un modelo estadístico de voz, la estimación FB-MMSE permite mejorar significativamente la información contenida en las réplicas, aún cuando esté muy degradada.

- Mediante la asignación de valores de confianza que posteriormente se tengan en cuenta durante el reconocimiento, es posible tratar las pérdidas en el propio reconocedor. Esto tiene la ventaja de que un modelo de voz más potente, el incluido en el propio reconocedor, se puede explotar de forma implícita. La asignación de valores de confianza puede realizarse de acuerdo con la covarianza normalizada entre la componente perdida y la componente estimada. Este esquema asigna confianzas a las réplicas (y a los vectores perdidos) de forma coherente, permitiendo que éstas sean post-procesadas por el propio reconocedor, obteniéndose mejores resultados.

- Alternativamente, el entrelazado de tramas resulta especialmente útil para la fragmentación de ráfagas, con la ventaja de no suponer un incremento del ancho de banda requerido. Sin embargo, el entrelazado de bloque de latencia mínima empleado por otros autores no permite controlar la dispersión de las pérdidas aisladas, resultando ésta excesiva. Los entrelazadores convolucionales descritos por Ramsey, en cambio, permiten controlar la distancia a la que se dispersan las pérdidas aisladas. Reduciendo esta distancia es posible, con la misma latencia, contrarrestar ráfagas mayores y aumentar la precisión durante el reconocimiento.

- Por último, las técnicas propuestas basadas en réplicas y el entrelazado de vectores, como técnicas independientes, pueden combinarse y proporcionar mejores resultados que los obtenidos por separado. Un esquema de doble flujo, en donde las réplicas constituyen un flujo virtual adicional que se entrelazada de distinta forma a como lo hacen los vectores de características, permite que las latencias introducidas por ambas técnicas no se sumen entre si, sino que la latencia total sea el máximo de ambas. Este esquema, combinado con el reconocimiento ponderado, ofrece los mejores resultados mostrados en este trabajo.

7.2. Contribuciones

Las principales aportaciones de esta tesis se pueden resumir en:

- Estudio de la influencia del ruido del canal durante el reconocimiento automático de la voz en un entorno de telefonía móvil [166, 167], centrado en el codec EFR (Enhanced Full Rate) y su canal de voz asociado TCH/EFS (Traffic Channel/Enhanced Full Speech).

- Mitigación de los errores de canal en un sistema de reconocimiento remoto basado en voz codificada con EFR [167]. El análisis de los errores del canal permite identificar tres tipos de ruido presentes en la señal de voz: background, de sustitución y apagado, y de memoria, siendo estos dos últimos los principales responsables de la reducción de precisión durante el reconocimiento. Mediante una sencilla interpolación es posible mitigar el ruido de sustitución y apagado, mientras que una adaptación de la técnica FCDCN permite compensar el ruido de memoria. Gracias a este esquema es posible habilitar un reconocimiento remoto de calidad (con tasas de reconocimiento cercanas a DSR) con el codec EFR.

- El desarrollo y aplicación de transparametrizadores para el reconocimiento de voz codificada en canales inalámbricos. Mediante el desarrollo de transparametrizadores para EFR [168] y el conjunto de codecs AMR es posible evitar, en su mayoría, el ruido de memoria, obteniéndose un rendimiento del reconocedor igual o superior a DSR en canales adversos. Los transparametrizadores desarrollados son además compatibles con DSR, no siendo necesario re-entrenar o modificar el backend. Además, se incluyen algunas propuestas para la aplicación de la transparametrización en la red GSM, realizando cambios centralizados (modificando sólo la TRAU) o, incluso, evitándolos (protocolo TFO) [168].
- El desarrollo de técnicas de mitigación basadas en estimación MMSE y MAP para reconocimiento distribuido sobre canales con pérdidas [109, 169], así como la comparación y combinación de ambos tipos de técnicas [170, 171].
- Mitigación de los efectos de la pérdida de paquetes en DSR sobre IP por medio de réplicas VQ y estimación MMSE [172, 173] y propuestas para la introducción de estas réplicas en el formato de carga útil RTP para DSR sin afectar al tamaño efectivo de los paquetes.
- El desarrollo de un esquema combinado que permite la aplicación de réplicas VQ y reconocimiento ponderado de Viterbi [173]. Gracias al desarrollo de una función de asignación de confianzas basada en la covarianza normalizada se indica al reconocedor que cierta información esta disponible, pero que debe tener menos peso durante el reconocimiento pues está degradada por un fuerte proceso de cuantización.
- La aplicación de la clase de entrelazadores de Ramsey para el reconocimiento robusto de la voz en IP [174].
- Comparación y aplicación conjunta de la estimación MMSE con réplicas VQ y el entrelazado de tramas a través de una estrategia de doble flujo [174].

7.3. Trabajo Futuro

Además de la transmisión por un canal propenso a errores, otro problema del reconocimiento remoto es una mayor presencia de ruido acústico, ya que se puede operar prácticamente en cualquier lugar. Dependiendo de la arquitectura, el ruido acústico puede potenciar, en mayor o menor medida, los efectos degradantes del canal. La arquitectura NSR

podría resultar sensible a este problema, ya que depende del rendimiento del codificador de voz en estos entornos. Por ello, sería conveniente evaluar el rendimiento de la transparetrización y la técnica A-FCDCN cuando la voz está contaminada con ruido acústico. Igualmente, la extensión de la técnica A-FCDCN para un tratamiento conjunto de ruido acústico y de canal podría resultar interesante.

Por otra parte, aunque en la arquitectura DSR pueden aplicarse front-ends que traten el ruido en el propio emisor (AFE y XAFE), reduciendo la presencia de este, sería interesante comprobar el rendimiento de las técnicas de recuperación propuestas en este trabajo cuando se aplican en ambientes ruidosos. Igualmente, un área interesante de investigación podría ser el traslado de los métodos de reducción de ruido aplicados en el front-end avanzado de ETSI (AFE) a la aproximación por transparametrización.

En lo referente al tratamiento de las pérdidas, el entrelazado de vectores de características se ha mostrado muy prometedor. El entrelazado aún no ha sido completamente explotado. En este trabajo hemos mostrado que los entrelazadores de bloque propuestos por otros autores no aprovechan completamente las particularidades de la transmisión de voz orientada a reconocimiento. Así, una línea que actualmente estamos investigando es el desarrollo de entrelazadores, basados por ejemplo en los convolucionales de Ramsey, que exploten completamente estas particularidades. En esta misma línea, el esquema de doble flujo deja muchas puertas abiertas, como la combinación óptima de entrelazadores y el uso de funciones de entrelazado para las réplicas que no correspondan a permutaciones.

Igualmente, sería interesante la aplicación de técnicas de codificación más avanzadas para las réplicas que la sencilla cuantización VQ. Si bien esto podría dificultar el modelado del canal en la técnica FB–MMSE con réplicas o la asignación de valores de confianza durante el reconocimiento ponderado, incrementaría la eficiencia de las réplicas (al transmitir más información) y podría mejorar el reconocimiento.

Por último, el tratamiento de errores durante el propio reconocimiento de voz, constituye un área de investigación muy prometedora. El desarrollo de heurísticas y técnicas más elaboradas para la asignación de valores de confianza, no sólo para las componentes estáticas sino también para las dinámicas, resultaría de gran utilidad en este ámbito.

Apéndice A

RESUMEN EN INGLES

Introduction

The ubiquitous and pervasive access to information services has become not only desirable but almost necessary. However, the new portable devices, which are getting smaller and smaller, make difficult the access to these services which claim for improved user interfaces. Information retrieval through speech recognition arises as a solution, however there are serious constraints to introduce a speech recognition subsystem within those devices. This has promoted a novel paradigm where speech recognition is performed outside the portable device in a distant server: remote speech recognition.

Under this approximation, the user device has to send coded speech, or a suitable parametrization of it for speech recognition, through a communication channel. Nowadays, two are the most important types of telecommunication networks worldwide, mobile networks (as GSM) and IP based networks (and Internet), which have constantly grown in the last years. These two networks have the advantage of an almost global deployment that allows remote speech recognition and access to information services whenever and wherever it is required. However, as disadvantage, both networks exhibit an error-prone channel. In mobile telephony, the radio channel is highly exposed to noise, while packet losses are usual in IP networks, since they were not designed for real time communications. As can be expected, these channel errors have a very negative impact on recognition performance.

In this dissertation, the influence of the aforementioned errors on speech recognition is

analyzed and different solutions to prevent, reduce and conceal their effects are developed. To this end, the channels of both networks are unified under the concept of lossy channels, wherein consecutive blocks of information can be erased or discarded. As a result of these losses, two different types of degradation can appear depending on how and what information is transmitted (complete coded speech or recognition-oriented speech feature vectors), severely reducing speech recognition accuracy.

If speech is coded through a predictive speech codec, long term memories usually involved in these codecs cause an additional distortion after the loss that we have called memory noise. In this thesis we evaluate the impact of this type of noise on the recognizer and propose two techniques to treat it. The first one tries to conceal the memory noise by modeling it in the cepstrum domain through additive factors, in a similar way to acoustic noise is often concealed. On the contrary, the second one directly tries to avoid it by transcoding the speech codec parameters to recognition-oriented feature vectors.

On the other hand, the distortion caused by the very loss of information can be concealed at the receiver through mitigation techniques. The statistical techniques proposed in this dissertation, MMSE and MAP estimation, offer good results since speech features exhibit a high short-term self-similarity. However, for this very reason, they are limited to short bursts of losses, where speech is unlikely to have evolved towards another sound. Thus, techniques which fragments the bursts into shorter ones are required. In order to do so, in this thesis the use of specific FEC codes as well as frame interleaving are proposed. In addition, the treatment of losses by the recognizer itself and the combination of this treatment with the aforementioned FEC codes and interleaving are explored.

A Source Model Mitigation Technique for Distributed Speech Recognition over Lossy Packet Channels

Ángel M. Gómez, Antonio M. Peinado, Victoria Sánchez, Antonio J. Rubio

Departamento de Electrónica y Tecnología de Computadores
Universidad de Granada
amgg@ugr.es

Abstract

In this paper, we develop a new mitigation technique for a distributed speech recognition system over IP. We have designed and tested several methods to improve the interpolation used in the Aurora DSR ETSI standard without any significant increase of computational cost at the decoder. These methods make use of the information contained in the data-source, because, in IP networks, unlike in cellular networks, no information is received during packet losses.

When a packet loss occurs, the lost information can be reconstructed through estimations from the N nearest received packets. Due to the enormous amount of combinations from previous and next received speech vector sequences, we have developed a methodology that drastically reduces the amount of required estimations.

1. Introduction

Since its beginning, Internet has been growing in size, incorporating many new networks, as well as in functionality, adding new services. As many other features have been integrated into Internet services, such as shopping, marketing and so on, Internet telephony services have also been incorporated. Nowadays, new services, such as Voice over IP (VoIP), offer an alternative to traditional speech transmission systems.

Researchers and developers have been trying to integrate speech recognition services into VoIP. However, two major VoIP distortion sources, speech codec and packet losses, decrease appreciably recognition accuracy [1],[2].

Speech codec distortion can be avoided under an Internet-based Distributed Speech Recognition (DSR). As many other services over Internet, it is based on a client-server architecture. On one hand, a simple and low power client, called front-end, analyzes, quantizes and packetizes speech data and sends it over the communication channel. On the other hand, a remote server, called back-end, receives the speech data and performs speech recognition. Thus recognition is not performed over encoded speech, because only those parameters which are relevant to the recognition process are transmitted through the channel. This scheme was proposed in [3] and it is an adaptation from the cellular ETSI DSR standard [4] to IP networks. The corresponding block diagram is shown in Figure 1.

Under this scheme, speech codec distortion is avoided because the parameters being used are directly extracted from the original speech, although the distortion due to packet losses is still present.

Work supported by the Spanish CICYT Project TIC-2001-3323

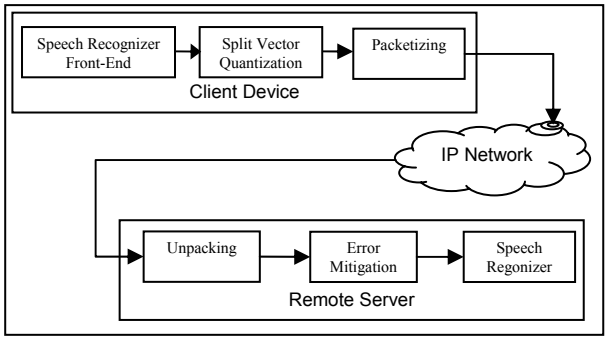


Figure 1: Block diagram of system architecture.

Packet losses are caused by the unsuitability of IP networks to offer a reliable and quality packet delivering service, since they were designed to offer a best effort service. In fact, on congested IP networks, routers will discard packets if their input flow exceeds their output flow for a given data route. In this scenario, a packet loss usually occurs in bursts, where multiple consecutive packets are lost.

As a consequence, the recognition accuracy is diminished and an error mitigation technique is needed. According to the taxonomy of error concealment and reparation techniques described on [5], we will focus on concealment techniques performed by the receiver.

In this paper, we first describe the experimental framework under which our techniques were run. Afterwards, we explain several methods of using the information contained in the data-source to obtain recognition accuracy improvements over lossy packet channels. And finally, we show the obtained results using our algorithms.

2. Experimental Framework

The experimental framework is very similar to that proposed in [3],[6]. At the user side, we will have a thin client that extracts, codes and sends only those parameters relevant to the recognition process. At the server side, the back-end receives the speech parameters, applies some kind of mitigation over them, and performs recognition. Speech parameters are transmitted over IP networks, so they should be packed and sent according to this kind of networks.

2.1. Database, Front-end and Recognizer.

To evaluate and compare the mitigation techniques proposed in this paper, the ETSI STQ-AURORA Project Database 2.0 experimental framework was adopted [7],[8]. The speech data has been extracted from clean sentences of the Aurora-2

database (connected digits spoken by American English speakers). Training is performed from a set of 8440 utterances containing a total of 55 male and 55 female adult speakers, and test is carried out over the clean sentences of set A, with 4004 utterances distributed into 4 subsets.

The front-end used in this work is the one proposed in the ETSI standard [4]. This front-end provides a 14-dimension feature vector containing 13 MFCCs (including the 0th order one) plus log-energy. These features are grouped into pairs and quantized by means of seven Split Vector Quantizers (SVQ). All codebooks have a 64-center size (6 bits), except the one for MFCC-0 and log-Energy, which has 256 centers (8 bits).

The recognizer is the one provided by Aurora and uses eleven 16-state continuous HMM word models, (plus silence and pause, that have 3 and 1 states, respectively) with 3 Gaussians per state (except silence, with 6 Gaussians per state).

2.2. Transmission and Channel Model.

After the SVQ quantization, each feature pair at time t is represented by a vector $\boldsymbol{c_t}$ $(\boldsymbol{c_t} \in \{\boldsymbol{c^{(i)}};\ (i = 0,\ldots,2^M - 1)\})$ (M=6, 8 in this work). The corresponding SVQ indices i_t (at time t) are encoded and sent over an IP network.

IP networks send data through datagrams or packets. In order to reduce the transmission overhead, we will consider that each packet contains speech features from two short-time analysis frames. Then, when a packet is lost we lose two feature vectors. This has been intentionally made, since sending one frame per packet could overload the network (packet header and payload would be similar in size). Furthermore, we want to focus over mitigation capabilities upon large burst losses.

Bolot [9] studied the distribution of packet loss in Internet and concluded that this could be approximated by a Markovian loss model such as a Gilbert model. We use this model to simulate packet losses. The Gilbert model is a two-state Markov model, as shown in figure 2.

In state 1 the packet is lost, so that, the packet loss rate (PLR) can be calculated as:

$$PLR = p/(p+q) \tag{1}$$

The tests were run under the loss condition probabilities reported in Table 1. These conditions are the same as the ones used in other works over IP [3], plus a new condition (5) with even larger bursts.

3. Mitigation Techniques

3.1. Aurora Mitigation Algorithm

Although the ETSI standard over DSR was developed in the context of digital cellular telephony, the same error mitigation algorithm can be used in speech recognition over IP. Frames received with errors can be associated to lost packets in the context of IP transmission.

In that form, if there are B consecutive lost packets, corresponding to 2B speech feature vectors, then the first B speech vectors are reconstructed with a copy of the last received speech vector before the burst and the last B speech vectors are reconstructed with a copy of the first received speech vector after the burst.

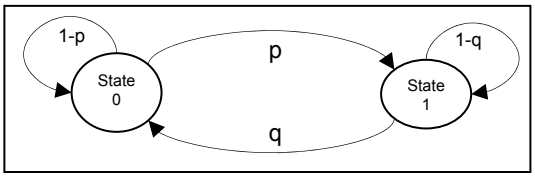


Figure 2: Gilbert Model. State 0 is the normal state and state 1 causes packet loss.

Condition	p	q	PLR
0	0	1	0
1	0.005	0.853	0.006
2	0.006	0.670	0.090
3	0.200	0.500	0.286
4	0.250	0.340	0.424
5	0.300	0.300	0.500

Table 1: Probability values and PLR for each condition.

We will use this 0th order interpolation as reference in our experiments.

3.2. First order Data-Source model.

Implicitly, 0th order interpolation uses information about the data-source. That is, since the speech features change quite slowly, an acceptable estimation of lost speech vectors is the nearest available speech vector. However, the information contained in the source can be exploited to get better estimations of the lost vectors. The problem is how to model this information in an efficient way.

From now on, we will focus on the mitigation of a given feature pair. The rest of feature pairs are processed in the same way. When a packet loss of 2B length occurs at time $t=1$, we will use the last received SVQ index before the burst (i_0) and the first one after it (i_{2B+1}), to build an estimation of the lost packets.

For each SVQ index $(i = 0,\ldots,2^M - 1)$, we obtain an estimation for the sequence of next feature pairs, called *forward estimation,* and for the sequence of previous feature pairs, called *backward estimation*, both of them with a certain fixed length L. In order to build the *forward estimation*, we search each index in the training database and average the sequences of feature pairs which follow it. Similarly, for the *backward estimation*, we search each SVQ index in the database and average sequences of feature pairs previous to it. Equation (2) expresses this. It is an adaptation of the minimum mean square estimation formula proposed in [10].

$$\hat{\vec{c}}_l(i) = \sum_{j=0}^{2^M-1} \vec{c}^{(j)} P\left(i_{t+l} = j \mid i_t = i\right) \quad (-L \le l \le -1),\ (1 \le l \le L) \tag{2}$$

For each index we will have a forward and backward estimation:

$$\begin{aligned} E_F(i) &= \left(\hat{\vec{c}}_1(i), \hat{\vec{c}}_2(i), \ldots, \hat{\vec{c}}_L(i)\right) \\ E_B(i) &= \left(\hat{\vec{c}}_{-L}(i), \hat{\vec{c}}_{-(L-1)}(i) \ldots, \hat{\vec{c}}_{-1}(i)\right) \end{aligned} \tag{3}$$

These estimations are precalculated from the training database and will not involve any computational cost in the decoder.

If we consider a burst starting at *t=1* and finishing at *t=2B*, then the first *B* speech feature pairs will be reconstructed with the *forward estimation* $E_F(i_0)$ where i_0 is the last received SVQ index before the burst, and the last *B* speech feature pairs will be reconstructed with the *backward estimation* $E_B(i_{2B+1})$, where i_{2B+1} is the first received SVQ index after the burst. If *B* is greater than *L*, then we will simply repeat the estimations from time *t=L* and *t=2B-L+1* towards the middle of the burst.

The memory requirements of this technique are small. It is necessary to store six tables of size 64x2x2xL (forward, backward and two speech features per index), plus one of size 256x2x2xL.

3.3. Second order Data-Source model.

The previous scheme can be extended to an estimation based on the two nearest correct frames. In this case, we will have to calculate a *forward* or *backward estimation* for every combination of two indices.

The reconstruction algorithm is similar to the first-order one. The first *B* speech feature pairs are reconstructed through the *forward estimation* from the two last received indices before the burst $E_F(i_{-1},i_0)$, and the last *B* speech feature pairs are reconstructed through the *backward estimation* from the two first received indices after the burst $E_B(i_{2B+1},i_{2B+2})$. As before, if *B* is greater than *L*, then we will repeat the estimations at time *t=L* and *t=2B-L+1* to the middle of the burst.

For this method, it is necessary to store six tables of size 64x64x4L, plus one of size 256x256x4L. It is a big amount of memory but it is still affordable for computers available nowadays.

3.4. Compressed N-order Data-Source model.

The extension of the above mitigation scheme to N-order data-source model is not affordable in the previous form. As we can see, the amount of required memory increases exponentially: we would need six tables of size 64^Nx4L plus one of size 256^Nx4L.

In the previous scheme we had exhaustive tables with each combination of two SVQ indices. However, this is not an efficient memory organization since some combinations do not even appear. Furthermore, not all the combinations appear with equal frequency in the database. Instead of that, we propose a different organization that considerably reduces the required amount of precalculated data.

We associate a register with a sequence of indices with a certain length *N*. Initially, there are as many registers as sequences or combinations of length *N* (64^N or 256^N depending on the SVQ quantizer). However, we only record those registers that appear more than a certain number μ of times in the training database. This can be easily done through an *"N-to-N"* comparison algorithm applied to a quantized training database.

Eliminating those registers which do not appear in the database or do not appear enough also has a beneficial side effect because these registers usually have bad *forward*/*backward estimations*. The amount of eliminated registers is notable and depends on threshold μ. Therefore, we can control the size of the register table through it.

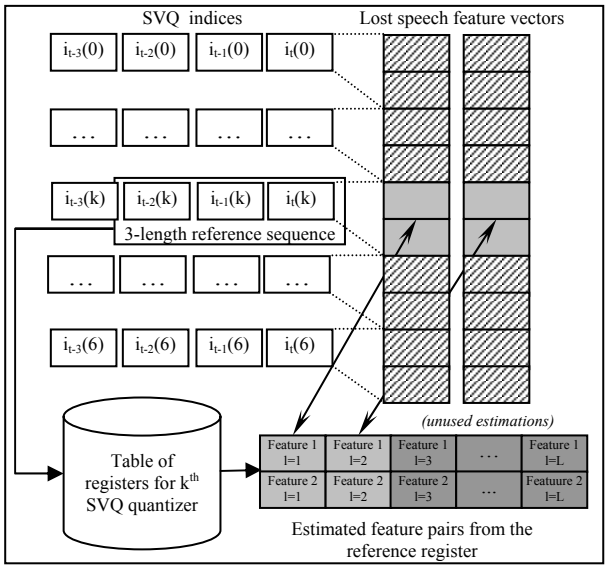


Figure 3: Forward reconstruction example of 2 feature pairs using compressed 3th static order estimation.

When a burst appears, we build a *previous reference register* of length *N* for each feature pair taking the *N* indices previous to the burst. If the first lost vector is at time *t=1*, then we will take the SVQ indices $(i_{-(N-1)},i_{-(N-2)},\ldots,i_0)$. At this step, it is possible that we can not build the reference register because there are not previous speech vectors at certain time *t=-r*, due to a previous packet loss to this burst or an utterance beginning. We can follow two different approaches:

- Since a 0^{th} order interpolation works well as reconstruction technique, we can suppose that unknown indices $(i_{-(N-1)},\ldots,i_{-r})$ are copies of the known index i_{-r+1}.
- Also, we can reduce the length of the reference register from *N* to *R* and recalculate the forward estimation as an average of all forward estimations of the registers which contain the same *R*-length reference register at the end.

When the previous reference register has been built, we search each reference register in the table of registers, getting the corresponding forward estimation or re-estimating it (with R-length reference registers), and replacing the first *B* speech feature pairs with them. An example diagram of a forward reconstruction with 3 length registers can be found in figure 3 (where $i_t(k)$ means the index *i* at time *t* of codebook *k*).

In order to perform the backward reconstruction, we build a *next reference register* for each feature pair in an analogous way to the *previous reference register*. If the last lost vector is at time *t=2B*, then we will take the SVQ indices (i_{2B+1}, $i_{2B+2},\ldots, i_{2B+N}$). Similarly to the *previous reference register*, we can take two ways when there are not next speech vectors from a given time *t=2B+r*: 1) To replace the unknown indices $(i_{2B+r},\ldots, i_{2B+N})$ with copies of the known index i_{2B+r-1} or 2) to reduce the length of the reference register and recalculate the backward estimation as average of all the backward estimations of the registers which contain the same *R*-length reference register at the beginning. After this, we search in the table and replace the last *B* feature pairs with the corresponding backward estimation.

If a reference register which is not in the table of registers is found, we will use a 0^{th} order interpolation as mitigation technique.

Compared to exhaustive tables, this method offers a trade-off between memory resources and complexity requirements, as it is necessary to carry out a search over seven tables of registers. The sequential search complexity is of the order *O(n)*, where *n* is the size of the register table. However, arranging the registers in the table we can use a binary search whose complexity is of the order *O(log(n))*. It must be taken into account that the binary search can not be applied under R-length reference registers.

4. Experimental Results

We have limited the compressed N-order method to N=3. Using exhaustive tables we would have more than 18 million estimations. However, using the most frequent registers with a μ threshold equal to 10 times, we have only 131.462 registers, with its corresponding estimations. As it can be seen, the required memory reduction is impressive.

The estimation length (*L*) has been set equal to 10, so the algorithm will be able to solve burst lengths up to 10 lost packets (20 lost vectors) without estimation repetition. Also, we have quantized the estimations to reduce memory requirements, so a pair of features (usually two floats of 4 bytes) is represented by an index (coded with only 1 byte). This means an 8:1 size reduction in the estimation tables.

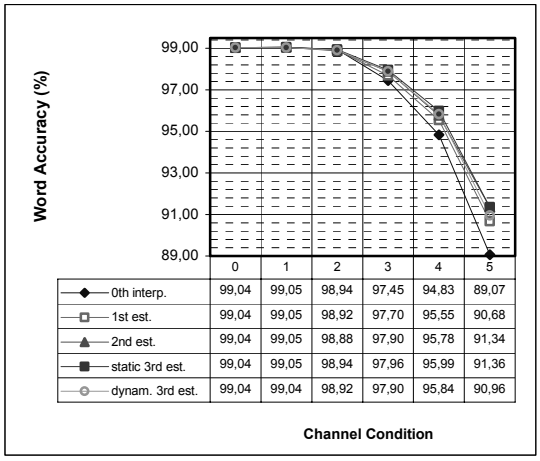
We have also tested the two possible ways of building registers when there are not *N* received indices available. These experiments are called *static 3rd est.* and *dynamic 3rd est*. In the first one, we use copied values repeating the nearest received vector for the unknown ones. In the second one, we reduced the 3 length register to 2 or 1 indices.

Figure 4 shows the results obtained by 0^{th} order interpolation (*0th interp.*), 1^{st} and 2^{nd} order source-model estimations (*1st est. and 2nd est.*), and compressed 3^{rd} order source-model estimations with static (*static 3rd est.*) and dynamic length (*dynamic 3rd est.*). We got the best results in the *static 3rd est.* experiment, where improvements over speech recognition accuracy were up to 2.29% for condition 5 and 1.16% and 0.5% for condition 4 and 3 respectively. That is, we got a relative improvement of 20-22% as regards Aurora mitigation algorithm.

Finally, a binary search over 132K registers take less than 18 steps and, besides, the estimations are always precomputed, so the computational cost involved by the static length algorithm is comparable to the 0^{th} order interpolation one. In view of the results obtained over *static length* and *dynamic length*, it is clearly preferable the first one, since the second one gets worse accuracy and its computational performance is lower.

5. Conclusions

In this work, we have developed a new mitigation technique based on a data-source model that does not add a significant increase of computational cost at decoder. This is possible thanks to the methodology described in this paper, in which we can use precomputed estimations based on the *N* nearest speech vectors received before and after a burst of lost packets, without an exponential increasing of memory requirements.



	0	1	2	3	4	5
0th interp.	99,04	99,05	98,94	97,45	94,83	89,07
1st est.	99,04	99,05	98,92	97,70	95,55	90,68
2nd est.	99,04	99,05	98,88	97,90	95,78	91,34
static 3rd est.	99,04	99,05	98,94	97,96	95,99	91,36
dynam. 3rd est.	99,04	99,04	98,92	97,90	95,84	90,96

Figure 4: Results of the proposed reconstruction techniques over the six conditions.

This new methodology opens new ways to reconstruct lost sequences through modeling the source information from a training database. It could also be applied in other fields where the data-source is correlated.

Computational costs are similar to a 0^{th} order interpolation, and memory requirements can be restricted according to the available resources.

6. References

[1] G.W. Cermak, "VoIP speech quality as function of Codec, Manufacturer, jitter and packet loss", *GTE laboratory technical report*, 1998.

[2] J. Van Sciver, J.Z. M, F. Vanpoucke and H. Van Hamme, "Investigation of speech recognition over IP channels", *IEEE International Conference on Acoustics, Speech and Signal Processing 2002*, vol. 4, pp. 3812-3815.

[3] C. Pelaez-Moreno, A. Gallardo-Antolin and F. Diaz-de-Maria, "Recognizing voice over IP: a robust front-end for speech recognition on the world wide web", *IEEE Transactions on Multimedia*, vol. 3, Jun. 2001.

[4] "ETSI ES 201 108 v1.1.2 Speech Processing, Transmission and Quality aspects; Distributed speech recognition; Front-end feature extraction algorithm; Compression algorithms", ETSI Standard, 2000.

[5] C. Perkins, O. Hodson, and V. Hardman, "A survey of packet-loss recovery techniques for streaming audio", *IEEE Network*, vol. 12, Aug. 1998, pp. 40-48.

[6] A.M. Peinado, V. Sánchez, J.L. Pérez-Córdoba and A. de la Torre, "HMM-Based Channel Error Mitigation and its Aplication to Distributed Speech Recognition", accepted in Speech Communication.

[7] H.G. Hirsch and D. Pearce, "The AURORA experimental framework for the performance evaluation of speech recognition systems under noisy conditions", *ISCA ITRW ASR2000*, Sept. 2000.

[8] "Aurora project database 2&3", http://www.icp.inpg.fr/ELRA/aurora2.html.

[9] J.C. Bolot, "End-to-end frame delay and loss behavior in the internet", *Proc. ACM SIGCOMM*, Sept. 1993, pp. 289-298.

[10] V. Sánchez, A. M. Peinado and J.L. Pérez-Córdoba, "Low Complexity Channel Error Mitigation for Distributed Speech Recognition over Wireless Channels", *IEEE ICC-2003 Proceedings*.

MITIGATION OF CHANNEL ERRORS IN EFR-BASED SPEECH RECOGNITION

Ángel M. Gómez, Antonio M. Peinado, Victoria Sánchez, José L. Pérez-Córdoba, Antonio J. Rubio

Dept. of Electronics and Computer Technology
Universidad de Granada, Spain
amgg@ugr.es

ABSTRACT

Network-based speech recognition (NSR) using the conventional speech channel with the Enhanced Full Rate (EFR) or the Adaptive Multi-Rate (AMR) codec is a very attractive approach since no change to existing mobile phones is needed. However, NSR reveals a degrading performance due to both transmission channel errors and the speech encoding process in comparison with Distributed Speech Recognition (DSR), where speech features are efficiently coded and transmitted on a data channel.

In this paper we focus on the degradation of the speech features caused by channel errors in an NSR system and propose methods to improve the quality of these features. Applying these methods, it turns out that the performance of an NSR system based on EFR coding is comparable to that based on DSR.

1. INTRODUCTION

The increasing development of cellular networks has thrown down a new challenge: the speech recognition in mobile devices that enables the access to voice activated services. These services can be implemented in a variety of conceptual solutions. A first approach could be to perform the speech recognition in the mobile device itself. Although this embedded solution can be feasible, its functionality is quite limited by hardware constraints and power consumption. It is therefore considered to be more efficient and practical to perform the recognition on a remote server.

In this scenario, there are two approaches. The first one, widely employed today, is known as *Network-based Speech Recognition* (NSR) [1]. NSR uses a full-duplex speech channel, with speech coding (for bit rate reduction) and channel coding (for error protection), to send the speech data to the remote server. The second one is referred to as *Distributed Speech Recognition* (DSR) [2]. In DSR, the speech recognition task is distributed between the local mobile device, which extracts and encodes the speech features, and the remote server, which performs the recognition itself. In this way, DSR avoids the speech coding step and the transmission is performed over a data channel, unlike in NSR.

However, there are still some problems in the final deployment of DSR. The current handsets are not capable of carrying out feature extraction and it would be necessary to include new hardware in the device. In addition, for some types of applications, it would be desirable to have the transmitted speech signal available just in case a further verification is required. Finally, although a standard has been established to ensure compatibility between the terminal and the remote recognizer [2], it does not cover the areas of data transmission or any higher level application protocols needed for the final implementation.

The NSR approach avoids these problems by performing the recognition from the decoded speech. The Enhanced Full Rate (EFR) codec, the most widely used codec in GSM, can achieve very similar results to DSR with clean transmission [1,3]. However DSR outperforms it in the presence of channel errors. In this work, we analyze these errors and their effects on speech recognition (section 2) and propose some solutions for them (section 3). Finally, the conclusions of this work are summarized in section 4.

2. EXPERIMENTAL FRAMEWORK

In order to evaluate and compare the techniques proposed in this paper, the ETSI STQ-AURORA Project experimental framework was adopted [4]. The speech data has been extracted from clean sentences of the Aurora-2 database (connected digits spoken by American English speakers). Training is performed from a set of 8440 clean utterances and test is carried out over the clean sentences of set A, with 4004 utterances.

The front-end used in this work is the one proposed in the ETSI standard [2]. It provides a 14-dimension feature vector containing 13 MFCCs (including the 0th order one) plus log-energy. The recognizer is the one provided by Aurora and uses eleven 16-state continuous HMM word models, (plus silence and pause, that have 3 and 1 states, respectively) with 3 Gaussians per state (except silence, with 6 Gaussians per state). The recognition performance is measured in terms of word accuracy.

Under the EFR scheme, the speech samples are transmitted using a full-duplex channel. These samples are coded and decoded according to GSM 6.60 standard [5]. Channel coding, decoding, error detection and correction and bad frame mitigation tasks are accomplished according to GSM 5.03 and GSM 6.61 [6,7]. On the other hand, under the DSR scheme, the speech features obtained from the front-end are quantized using a Split Vector Quantizer (SVQ) that groups them into pairs (MFCCs 1 and 2, MFCCs 3 and 4, ..., MFCC 0 and log-Energy). Each pair has its own codebook that is generated utilizing a weighted distance measure. The resulting bitstream is transmitted according to ETSI DSR standard through a data channel. After decoding, the error mitigation algorithm proposed in the DSR ETSI standard is applied.

Work supported by the Spanish CICYT Project TIC-2001-3323

The channel is simulated using the GSM error patterns (EPx, *x=1,2,3*) to corrupt the bit stream. These error patters are in AEG format and represent three channel conditions: EP1 (10dB C/I, good quality), EP2 (7dB C/I, medium quality) and EP3 (4dB C/I, lower quality).

3. ANALYSIS OF GSM-EFR CHANNEL ERRORS

When a speech frame reaches the receiver, it is decoded applying error correction and checking the various protection mechanisms included in the frame. As a result of this process, a Bad Frame Indicator (BFI) will be enabled if a transmission error is detected in that frame. Due to the discriminative treatment of the frames by the decoder, we should distinguish between different types of noises derived from the channel noise.

3.1. Bad frame noise and background noise

If the BFI of a frame is enabled (BFI=1), that frame has been seriously damaged and its synthesis would be very unpleasant for a listener. For this reason, in order to improve the subjective perception of the signal, these frames are replaced by a repetition or extrapolation of the last received frame or frames. The GSM standard 6.61 does not impose any specific substitution and muting algorithm, however, it proposes an example which is usually the implemented one [7]. This substitution is performed in such a way that the output level gradually becomes comfort noise. On the other hand, it can not be asserted that those frames whose BFI are not marked (BFI=0) are entirely correct, since the error protection is perceptually applied and all the speech parameters are not equally protected. In this sense, the signal transmitted with errors can present anomalies with regards to the clean transmitted signal, which would influence the recognition accuracy.

In our study, we have designed two experiments intended for evaluating the effects of the aforementioned errors. We have tested the system performance in these situations:

- *Background Noise*. In this situation, we only take into account the noise generated by unmarked frames (BFI=0). The frames marked as no valid (BFI=1) are replaced with the corresponding valid frame. This valid frame is built up from the parameters obtained in a clean transmission.
- *Bad Frame Noise*. In this case, we only take into account the noise from frames with marked BFI. In order to do this, all frames, except those whose BFI is enabled, are replaced with those obtained in a clean transmission. This avoids the presence of anomalies in unmarked BFI frames.

Table 1 (cols 4 and 5) shows the results obtained from these two experiments together with those of DSR and EFR (cols 2 and 3), under different channel conditions. As it can be seen, the *Bad frame noise* is mainly the responsible for the performance degradation in noisy channel conditions, while the *Background noise* has a negligible importance. Due to the bursty nature of the wireless channel (in spite of the interleaving introduced in the encoder) a high proportion of bad frames appear consecutively, that is, constituting bursts. The recognition accuracy in accordance with the length (*l*) of the bursts on EP3 condition is shown in figure 1. When *l<1*, there are no bad frames (a burst involves at least 1 bad frame), and we obtain the same results as in clean conditions. In the other extreme, when the length is less than ∞, all bad frames are present and we obtain the same results as with the EP3 condition. As it can be observed, the incremental accuracy reduction is negligible for burst lengths bigger than five, due to the small frequency of appearance of these bursts.

Channel	DSR	EFR	Backgr. Noise	Bad Fr. Noise	Codec Memory Noise	Bad Fr. Isolated Noise
Clean	99.04	98.70	-	-	-	-
EP1	99.04	98.44	98.50	98.61	98.44	98.68
EP2	98.95	96.91	98.31	97.53	97.73	98.28
EP3	93.41	84.48	98.22	85.80	93.54	90.47

Table 1. *Word accuracy in recognition with each kind of noise.*

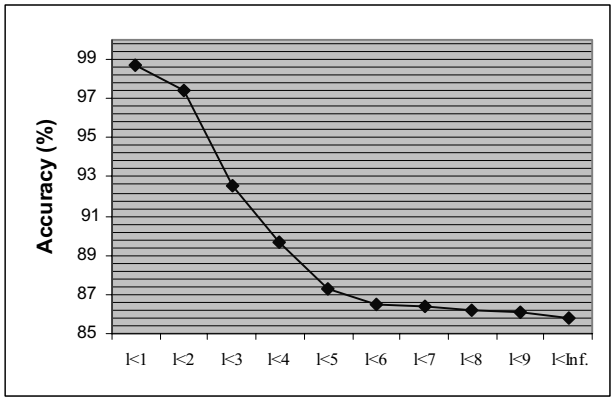


Figure 1. *Word accuracy versus burst length (l) on EP3.*

3.2. Codec memory noise

Analyzing the speech samples obtained in a noisy transmission, we can observe a degradation of the signal corresponding to correct frames after a burst (bad frame noise). This is due to the memory of the CELP type codec. In this sense, although several frames after the burst had been received without errors, the resulting signal would be degraded due to the previous erroneous frames. This degradation constitutes what we call *Memory noise*.

In the same way as in the previous experiments, we can isolate the effects of the bad frame noise from the memory noise. In this case, we previously eliminate the background noise and operate at a different level substituting speech samples instead of speech parameters. We consider two new experiments:

- *Bad Frame Isolated Noise*. In this case, we isolated the alterations caused by the bursts, not by the associated memory effect. To this end, speech samples belonging to good frames (BFI=0) are replaced with the corresponding correct speech samples.
- *Memory Noise*. In this experiment, the noise generated after a burst by the memory of the codec is isolated. For this purpose, speech samples belonging to bad frames (BFI=1) are replaced with the corresponding correct speech samples, leaving only the corruption corresponding to memory noise.

Table 1 (cols 6 and 7) shows the results of both experiments. As it can be seen, the memory noise has an important responsibility for the reduction of the recognition accuracy. This confirms the influence of a burst beyond its limit in terms of bad frames.

4. IMPROVING ACCURACY OVER EFR

Since our goal is the recognition of the speech degraded by channel errors, we will try to compensate the speech features used for recognition rather than the speech signal. However, there are differences of size and shift between the windows of the EFR codec and the Aurora feature extractor. This difficulty can be avoided with a mapping function which relates each bad EFR frame with a bad feature vector. In our work, this function is defined as:

$$F_{map}(n)=\begin{cases}1 & \left(BFI\left(\left\lfloor\frac{n}{2}\right\rfloor\right)=1\right) or \left(BFI\left(\left\lfloor\frac{n+1}{2}\right\rfloor\right)=1\right) \\ 0 & ;otherwise\end{cases} \quad (1)$$

where n is the time index of a given feature vector and $BFI(m)$ is the bad frame indicator of frame m (both starting from 0). $F_{map}(n)$ is 0 when feature vector n is received and equal to 1 when feature vector n is bad (see figure 2).

4.1. Burst reconstruction

Whenever an error burst appears, frames with BFI enabled are considered as lost frames in GSM 6.61 [5]. In this situation, there is no information about the original signal. Then, the corresponding bad feature vectors (according to the mapping function) are lost and must be reconstructed. This reconstruction can be accomplished from the last and the first vectors received before and after the burst, respectively, by means of a simple linear interpolation:

$$\hat{\boldsymbol{x}}(t)=\boldsymbol{x}(t_s)+\frac{\boldsymbol{x}(t_e)-\boldsymbol{x}(t_s)}{t_e-t_s}\left(t-t_s\right) \quad \left(t_s<t<t_e\right) \quad (2)$$

where $\hat{\boldsymbol{x}}(t)$ is the estimated feature vector at time t, $\boldsymbol{x}(t_e)$ is the first vector after the burst and $\boldsymbol{x}(t_s)$ is the last vector before the burst. Although this is a very simple technique, an important improvement over EFR can be obtained as shown in table 2 (col 4, *EFR interpolation*).

4.2. Memory noise compensation

By contrast to burst errors, where there are lost frames, the memory noise only involves signal degradation. In a first approach, this noise can be considered similar to acoustic noise, whereby, under this assumption, it is feasible to apply an acoustic noise compensation algorithm as FCDCN (Fixed Codeword-Dependent Cepstral Normalization) [8] to try to compensate the codec memory noise. FCDCN applies a correction based on simultaneously recorded noisy and clean speech data (stereo data). This correction depends on the instantaneous SNR (Signal-to-Noise Ratio) of the input. Furthermore, for each codeword q, we should consider a different correction. It usually represents a quantization index for the input, relating the speech vectors of the input and the correction factor to apply. In this way, the degraded speech vectors are compensated as follows:

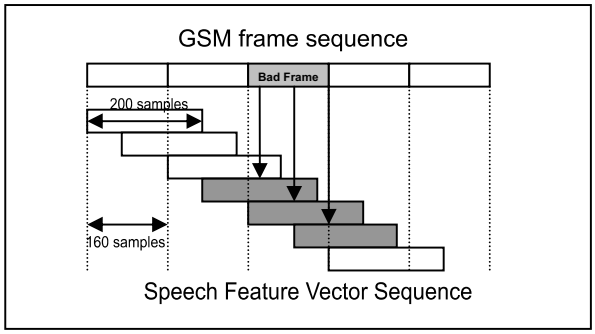


Figure 2. *Mapping function between GSM frames and speech feature vectors.*

$$\hat{\boldsymbol{x}}=\boldsymbol{x'}+\boldsymbol{r}\left[SNR,q\right] \quad (3)$$

where $\hat{\boldsymbol{x}}$ is the estimated vector, $\boldsymbol{x'}$ is the noisy vector and $\boldsymbol{r}$ is the correction vector for a given SNR and a given codeword q.

The memory noise depends on the previous burst error: the longer the burst length, the higher the noise level. Furthermore, it decreases as good frames are received. Therefore, the instantaneous *SNR* of the noisy feature vector depends on the previous burst length and the distance to it. Due to this dependence, a different FCDCN correction should be applied for each burst length and time interval after it, modifying equation (3) to:

$$\hat{\boldsymbol{x}}=\boldsymbol{x'}+\boldsymbol{r}\left[l,d,q\right] \quad (4)$$

where l is the length of the previous burst and d is the distance from the current noisy vector $\boldsymbol{x'}$ to the first vector after the burst (measured in number of vectors).

In order to reduce the computational burden and memory requirements, the maximum burst length can be limited to $l=5$, since the reduction on accuracy is negligible for longer burst lengths (figure 1). Furthermore, the maximum distance to the burst can be set on $d=20$ feature vectors after the burst. As far as we know, this is a safe limit since the correction factors at that distance are close to zero (no correction).

On the other hand, the codewords are defined through the SVQ quantizer used by the Aurora front-end described on section 2. This involves working with feature pairs instead of feature vectors. In this way, each noisy feature vector is quantized giving seven indices or codewords. Each one selects the compensation factor for its corresponding feature pair. This gives the last modification on equation (3):

$$\hat{\boldsymbol{p}}=\boldsymbol{p'}+\boldsymbol{r}(l,d,q) \quad \left(q=SVQ(\boldsymbol{p'})\right) \quad (5)$$

where $\hat{\boldsymbol{p}}$ is the compensated feature pair while $\boldsymbol{p'}$ is the noisy feature pair and q is the SVQ quantization index for $\boldsymbol{p'}$.

Every feature vector received after a burst (affected by memory noise) is compensated until its distance to the previous burst is longer to 20 or a new burst appears. On the other hand, if the previous burst length is longer than 5, we reuse the correction factors applied on burst lengths equal to 5.

Finally, this algorithm requires stereo speech data in order to compute the compensation factors. The training database can be used to accomplish this objective. Simulating burst errors of same length during the transmission, we can build up as many stereo data as it is needed. Given M bursts with the same length l, each one ending at time t_n ($n=1,..,M$), the correction factors at distance d are calculated as follows:

$$\begin{aligned} \boldsymbol{r}(l,d,q) &= \frac{1}{M}\sum_{n=1}^{M}\left(\boldsymbol{p}_{t_n+d} - \boldsymbol{p'}_{t_n+d}\right) & 1 \le l \le 5 \\ q &= SVQ\left(\boldsymbol{p'}_{t_n+d}\right) & 0 < d \le 20 \end{aligned} \qquad (6)$$

where $\boldsymbol{p}_t$ and $\boldsymbol{p'}_t$ are the clean and noisy feature pairs at time t ($t=t_n+d$), respectively.

The proposed adapted FCDCN for memory noise compensation can be used together with linear interpolation for burst reconstruction. Due to the fact that the burst reconstruction depends on the previous and next received vector, it must be applied after memory noise compensation. Table 2 shows the results obtained applying both algorithms (col 5, *EFR Interp. & Adapted FCDCN*). It can be seen that it outperforms EFR, approaching the DSR performance.

4.3. Extension to codec noise

The aforementioned correction factors were computed comparing noisy transmitted speech with clean transmitted encoded speech. However, we can also compensate the distortion introduced by the coding process by computing the correction pairs from non-encoded speech. In this way, memory and codec noise are compensated at the same time after every burst. Moreover, an additional set of correction pairs, $\boldsymbol{r}(q)$, can be computed comparing encoded and non-encoded speech. This set is applied over the feature vectors in the beginning, when there is no previous burst, and after the 20 vectors after a burst, compensating only the distortion introduced by the codec.

Table 2 shows the results of this extension combined with linear interpolation for burst reconstruction (col 6, *EFR Interp. & Adapted FCDCN (clean speech)*). Extending in this way the algorithm, we can improve the recognition accuracy, obtaining a performance quite close to DSR one.

5. CONCLUSIONS

In this work, we have focused our study in the effect that trasmission channel errors have on a NSR system using the EFR speech codec. We have analyzed the impact of three different types of errors over the recognition system caused by an erroneous transmission: background noise, bad-frame isolated noise and codec memory noise.

From this analysis, we have observed that the system degradation is mainly due to the two last error types. By means of a differentiated treatment of these errors, we have shown that the NSR approach can achieve results similar to DSR. Furthermore, the proposed algorithms do not have especial computational requirements and can be easily applied on real time. Besides, they are extensible to other CELP type codecs.

Channel	DSR	EFR	EFR Interpolation	EFR Interp. & Adapted FCDCN	EFR Interp. & Adapted FCDCN (clean speech)
Clean	99,04	98,70	98,70	98,70	98,81
EP1	99,04	98,44	98,43	98,46	98,64
EP2	98,95	96,91	97,55	97,82	98,19
EP3	93,41	84,48	90,76	94,04	94,04

Table 2. *Word accuracy (%) in recognition with the proposed enhancement algorithms.*

The main disadvantage of these algorithms is the necessity of the BFI flags. The mapping function requires these flags to discriminate between received and lost speech frames. This requires either direct access to the GSM bitstream or an algorithm capable of directly detecting bad frames from the speech samples.

Finally, in the case of the AMR codec, the speech and channel encoding is modified in order to face the channel conditions. Although less affected by transmission channel errors, we will still have similar errors as the ones described in this paper. In AMR we would additionally have to consider the degradation introduced by the speech encoding process on the NSR system. Further work will address this problem.

6. REFERENCES

[1] T. Fingscheidt, S. Aalburg, S. Stan and C. Beaugeant, "Network-based vs. distributed speech recognition in adaptive multi-rate wireless systems", ICSLP 2002, Denver, September 2002.

[2] "ETSI ES 201108 Speech Processing, Transmission and Quality aspects; Distributed speech recognition; Front-end feature extraction algorithm; Compression algorithms", ETSI Standard, 2000.

[3] H.G. Hirsh, "The influence of speech coding on recognition performance in telecommunication networks", ICSLP 2002, Denver, September 2002.

[4] H.G. Hirsch and D. Pearce, "The AURORA experimental framework for the performance evaluation of speech recognition systems under noisy conditions", *ISCA ITRW ASR2000*, Sept. 2000.

[5] "ETSI EN 300 726. Enhanced Full Rate (EFR) speech transcoding", ETSI Standard, 1999.

[6] "ETSI EN 300 909. Channel Coding", ETSI Standard, 1999.

[7] "ETSI EN 300 727. Substitution and muting of lost frames for Enhanced Full Rate speech traffic channels", ETSI Standard, 1999.

[8] A. Acero, "Acoustical and Environmental Robustness in Automatic Speech Recognition", Ph.D. Thesis, Dept. of Electrical and Computer Engineering, Carnegie Mellon Univ., 1990.

A Comparison of Packet Loss Compensation Methods and Interleaving for Speech Recognition in Burst-Like Packet Loss

Alastair James[1], Ángel Gómez[2] and Ben Milner[1]

[1]School of Computing Sciences, University of East Anglia, UK
[2]Department of Electronic and Computer Technology, University of Granada, Spain
{a.james, b.milner}@uea.ac.uk, amgg@ugr.es

Abstract

This work compares the performance of three compensation methods for speech recognition in the presence of packet loss. Two methods, cubic interpolation and a novel maximum a posteriori (MAP) estimation, aim to restore the feature vector stream in the event of packet loss, while the third technique applies compensation in the decoding stage of recognition through missing feature theory. To improve performance in burst-like packet loss, interleaving is introduced to disperse bursts of loss. Experiments on the ETSI Aurora connected digit task show best performance to be given by a combination of missing feature theory and cubic interpolation. This raises performance from 50.3% to 69.8% at a packet loss rate of 50% and average burst length of 20 packets. Including interleaving further increases performance to over 76%.

1 Introduction

The move towards mobile and handheld devices for speech communication has lead to distributed speech recognition (DSR) systems being developed. The Aurora DSR standard proposed by the European Telecommunication Standards Institute (ETSI) gives improved speech recognition accuracy by replacing the low bit-rate speech codec on the terminal device with the static MFCC feature extraction component of the speech recogniser [1]. Including noise compensation on the terminal device gives good performance on noise contaminated speech. Figure 1 shows a typical DSR architecture along with the proposals outlined in this work.

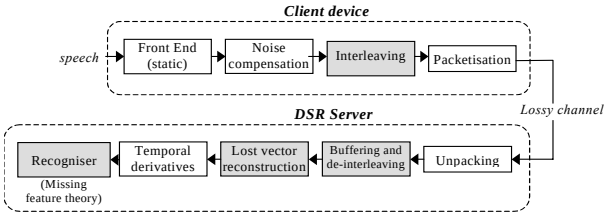


Figure 1: DSR architecture with packet loss compensation and interleaving.

The networks across which DSR systems transmit packetised speech data often do not guarantee reliable delivery. When packet loss occurs, or too many bits are corrupted so that bit level forward error correction cannot correct the frame, then portions of the feature vector stream become lost. Techniques to improve recognition performance on these unreliable channels essentially fall into three groups. The first set of techniques aim to protect the feature vectors through forward error correction. Such methods include cyclic redundancy checks (CRC) and Reed-Solomon coding [1][2]. Other techniques restore the feature vector stream in the event of lost packets through estimation of missing vectors. Simple methods use repetition of previously received vectors or linear and non-linear interpolation [3][4]. The final set of techniques apply packet loss compensation either partly or fully at the decoding stage through modification of the observation probabilities within hidden Markov models (HMM). Missing feature theory has been shown effective for this and also weighted Viterbi recognition which adjusts the contribution of estimated feature vectors according to how accurate they are likely to be [5][6]. These techniques are all effective for short duration bursts of packet loss but degrade as burst lengths increase. To reduce burst lengths, interleaving the feature vectors prior to transmission has been shown effective at giving substantial gains in recognition accuracy, although at the expense of an increase in delay [4].

The aim of this work is to compare a range of packet loss compensation techniques both with and without interleaving. Packet loss compensation methods are discussed in section 2 and a novel method of estimating lost packets is also introduced based on maximum a posteriori (MAP) estimation. Section 3 describes the interleaving process. Experimental results which compare the compensation techniques are presented in section 4 as well as a study of the optimal interleaving depth. Finally a conclusion is made in section 5.

2 Compensation against Lost Vectors

This section describes three techniques to compensate for lost feature vectors in the event of packet loss. Two methods, interpolation and MAP estimation, attempt to restore the feature vector stream by estimating lost vectors, while the third method, missing feature theory, compensates for lost vectors at the decoding stage of recognition.

2.1 Interpolation

Several interpolation schemes have been considered for estimating vectors lost due to packet loss [3][4]. Of these non-linear interpolation using cubic Hermite polynomials has been found to give best performance [4], where the n^{th} lost vector in a burst of length β, starting at vector $b+1$ is given as

$$\hat{\mathbf{x}}_{b+n} = \mathbf{a}_0 + t\mathbf{a}_1 + t^2\mathbf{a}_2 + t^3\mathbf{a}_3 \qquad 1 \le n \le \beta \tag{1}$$

where $t=n/(\beta+1)$ and the multivariate coefficients, $\{\mathbf{a}_0,.., \mathbf{a}_3\}$, can be computed from the two vectors preceding and following the burst of loss, $\mathbf{x}_{\text{before}}$ and $\mathbf{x}_{\text{after}}$, and their first derivates, $\mathbf{x}'_{\text{before}}$ and $\mathbf{x}'_{\text{after}}$. Expressing the interpolation function in terms of Hermite basis functions gives the estimate of the n^{th} feature vector within the burst as,

$$\begin{aligned}\hat{\mathbf{x}}_n = {} & \mathbf{x}_{before}\left(1-3t^2+2t^3\right)+\mathbf{x}_{after}\left(3t^2-2t^3\right) \\ & +\mathbf{x}'_{before}\left(t-2t^2+t^3\right)+\mathbf{x}'_{after}\left(t^3-t^2\right) \qquad 1 \le n \le \beta\end{aligned} \tag{2}$$

where the derivates are approximated by $\mathbf{x}'_{before} = \beta(\mathbf{x}_{before} - \mathbf{x}_{before-1})$ and $\mathbf{x}'_{after} = \beta(\mathbf{x}_{after+1} - \mathbf{x}_{after})$. This requires two vectors either side of the burst from which to compute the derivatives. If two vectors are not available the derivative is set to zero. In practice it was found that rapid fluctuations of the feature vector stream resulted in overestimation of derivative components causing the interpolation to overshoot. Improved performance was achieved by reducing large derivative estimates by applying logarithmic compression to the vector differences,

$$f(x) = \mathrm{sgn}(x)\log(|x|+1) \tag{3}$$

which gives $\mathbf{x}'_{before}$ and $\mathbf{x}'_{after}$ as,

$$\mathbf{x}'_{before} = \beta \times f(\mathbf{x}_{before} - \mathbf{x}_{before-1}) \tag{4}$$

$$\mathbf{x}'_{after} = \beta \times f(\mathbf{x}_{after+1} - \mathbf{x}_{after}) \tag{5}$$

2.2 MAP estimation

A better approach for estimating the value of lost vectors is through statistical methods which use prior information about the nature of the signal [7]. In maximum a-posteriori (MAP) estimation a sequence of lost vectors, $\boldsymbol{X}_m$, can be calculated in order to maximize their likelihood conditioned on the values of the correctly received vectors, $\boldsymbol{X}_o$, and the overall distribution of the feature vector stream $P(X;\mu, \Sigma)$. Assuming this distribution is Gaussian, MAP estimation can be simplified to a linear regression [8] given by,

$$\hat{\mathbf{X}}_m = \mu_m + \Sigma_{mo}\Sigma_{oo}^{-1}(\mathbf{X}_o - \mu_o) \tag{6}$$

where μ_m and μ_o are the mean vectors of $\boldsymbol{X}_m$ and $\boldsymbol{X}_o$ respectively, Σ_{oo} is the auto-covariance matrix of $\boldsymbol{X}_o$ and Σ_{mo} is the cross-covariance matrix between $\boldsymbol{X}_m$ and $\boldsymbol{X}_o$. The assumption of wide-sense stationarity leads to the assumption that the mean and covariance of the MFCC features are independent of their current position in time. Given the k_1^{th} coefficient of the MFCC vector at time instant t_1, $S(t_1,k_1)$, and the k_2^{th} coefficient of the MFCC vector at time t_2, $S(t_2,k_2)$, this assumption allows the means and covariance to be given as,

$$\begin{aligned} \mu(t_1,k_1) &= E[S(t,k_1)] = \mu(k_1) \\ \mu(t_2,k_2) &= E[S(t,k_2)] = \mu(k_2) \\ E[(S(t_1,k_1)-\mu(k_1))(S(t_2,k_2)-\mu(k_2))] &= c(t_1-t_2,k_1,k_2) = c(\tau,k_1,k_2) \end{aligned} \tag{7}$$

where $E[\ ]$ is the expectation operator, $\mu(k)$ is the expected value of each coefficient in the MFCC vector and $c(\tau, k_1,k_2)$ defines the covariance between any coefficient and any other coefficient τ time instants later in the MFCC vector sequence.

Lost vectors are reconstructed by arranging the received MFCCs into $\boldsymbol{X}_o$, whilst the lost features are arranged into $\boldsymbol{X}_m$. Since both the mean of all the coefficients in the MFCC vector sequence and the covariance between any two components (equation **7**) are known, an estimate of the lost vectors, $\hat{\boldsymbol{X}}_m$, can be made. In practice not all the received features in the utterance are used in $\boldsymbol{X}_o$ as this imposes too much of a computational overhead. Instead $\boldsymbol{X}_o$ is limited to a few vectors in the region of the loss.

Isolated lost vectors can be efficiently reconstructed through MAP estimation, however bursts of lost vectors require an iterative strategy where each vector is individually reconstructed and their estimations are reused in the reconstruction of the next vector. Otherwise a large amount of $c(\tau, k_1,k_2)$ values would be required or the most inner lost vectors of a burst could not be reconstructed since they are too far in time from the observed (received) features. Different strategies have been considered for the order of vector estimation within a burst with the most effective being iterative reconstruction from the outer to the inner vectors of a burst.

MAP estimation is often computationally expensive due to the inversion of large covariance matrices. However, not all observed coefficients involved in the estimation are equally relevant. Therefore, the size of the auto-covariance matrix can be reduced by reconstructing each coefficient of the feature vector separately from the most relevant observed coefficients. This selection can be made by considering their relative covariances in relation to the coefficient under estimation,

$$r(\tau,k_1,k_2) = \frac{c(\tau,k_1,k_2)}{\sqrt{c(0,k_1,k_1)c(0,k_2,k_2)}} \tag{8}$$

Only those features having a high relative covariance, with respect to the coefficient under estimation, are used in the MAP estimation. In particular the DCT operation applied during feature extraction means that coefficients are considerably more correlated along time than quefrency and this fact is supported by their observed relative covariances. Therefore, each coefficient of the feature vector, $S(t,k)$, can be reconstructed from received features using the same MFCC coefficient, i.e., $S(t-\tau,k),\ldots,S(t-1,k),S(t+1,k),\ldots,S(t+\tau,k)$. This gives a considerable reduction in the size of the observation covariance matrix and leads to a substantial reduction in computation[8].

2.3 Missing feature theory

The methods outlined above attempt to reconstruct the feature vector stream based on those feature vectors correctly received. An alternative approach is to compensate for lost vectors at the decoding stage through the technique of missing feature theory [5]. The observation probability, $b_j(\boldsymbol{x}_i)$, associated with the i^{th} feature vector, $\boldsymbol{x}_i$, in state j of the HMM is modified according to whether particular coefficients of the vector were received or not,

$$b_j(\mathbf{x}_i) = \prod_{k=1}^{K}\left(\frac{1}{\sqrt{2\pi\sigma_k^2}}\exp\left(-\frac{(x_{k,i}-\mu_k)^2}{2\sigma_k^2}\right)\right)^{\rho_{k,i}} \tag{9}$$

K is the dimensionality of the feature vector, μ_k and σ_k are the mean and variance of the k^{th} coefficient of the feature vector in state j. The variable $\rho_{k,i}$ is set to 1 if the k^{th} coefficient of the i^{th} feature vector is suitable for inclusion in the observation calculation, or set to 0 if it is unsuitable. The feature vector in this work comprises 12 static MFCC coefficients and a log energy term which are augmented by their velocity and acceleration derivatives to give a $K=39$ dimensional feature vector. Temporal derivative components are calculated using a 'sliding-window' technique [1], where the window extends $w_V=3$ frames either side of the current element for the velocity and $w_A=2$ frames either side for the acceleration. As shown in [9], the loss of a burst of β static vectors has the effect of corrupting $2w_V+\beta$ velocity and $2(w_V+w_A)+\beta$ acceleration features. Therefore, for every static vector removed from the calculation, several velocity and acceleration features will need to be removed by setting the corresponding $\rho_{k,i}$ values to 0, as shown in figure 2a. Note that for a particular i, when $\rho_{k,i}$ is zero for all k, no information can be obtained from the

observation and the decoded state lattice depends wholly on the HMM state transition probability matrix for this frame.

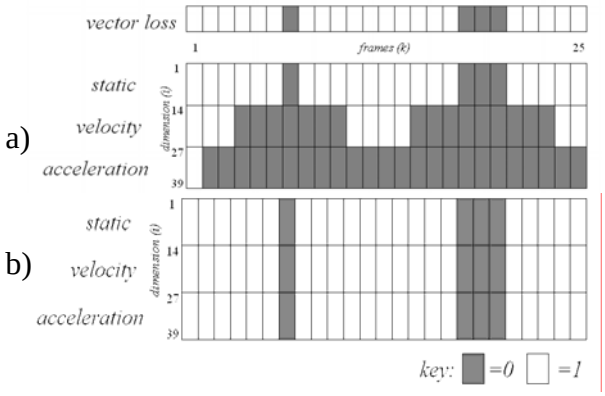


Figure 2: Methods of considering temporal derivatives

In severe packet loss, where bursts of loss are positioned close together, the regions of removed temporal derivative components can overlap, causing the higher-order temporal derivatives to be removed from the calculation for an increased number of frames. In order to prevent this, an alternative method is proposed, whereby the missing static feature vectors are reconstructed (using a method such as cubic interpolation) purely for the purpose of calculating the temporal derivatives. The $\rho_{k,i}$ values are only set to 0 for those feature vectors that are not received, as shown in figure 2b.

3 Interleaving

The packet loss compensation methods described in the previous section are effective for short duration bursts of loss but deteriorate at longer burst lengths. An effective method to reduce burst lengths in the received feature vector stream is to employ an interleaver on the terminal device. For a given sequence of feature vectors, $\boldsymbol{X} = \{\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{N-1}\}$, the interleaving operation can be expressed as a permutation producing a re-ordered sequence, $\boldsymbol{X}'$, given as,

$$\mathbf{X}' = \{\mathbf{x}_{\pi(0)}, \mathbf{x}_{\pi(1)}, \mathbf{x}_{\pi(2)}, \ldots, \mathbf{x}_{\pi(N-1)}\} \quad (10)$$

The interleaving function, $\pi(i)$, gives the index of the vector to be output at the i^{th} time instance. Feature vectors are returned to their original order on the receiver side through de-interleaving which is given by the inverse function of π, i.e.,

$$\pi^{-1}(i) \quad where \quad \pi\left(\pi^{-1}(i)\right) = i \quad (11)$$

In the presence of burst-like packet loss the interleaving process is able to distribute long bursts of loss into a series of shorter duration losses.

The block interleaver of degree d operates by re-arranging the transmission order of a $d \times d$ block of input vectors. Two block interleavers, π_{block1} and π_{block2}, [4] are considered optimal in terms of maximising their spread for given degree, and are given,

$$\pi_{block}(id + j) = (d - 1 - j)d + i \quad where\ 0 \le i,j \le d-1 \quad (12)$$

$$\pi_{block2}(id + j) = jd + (d - 1 - i) \quad where\ 0 \le i,j \le d-1 \quad (13)$$

It is interesting to observe that π_1 and π_2 form an invertible pair as $\pi_1 = \pi_2^{-1}$ *and* $\pi_2 = \pi_1^{-1}$. The interleaving operation is equivalent to a rotation of the block of feature vectors either 90° clockwise or 90° anti-clockwise as shown in figure 3.

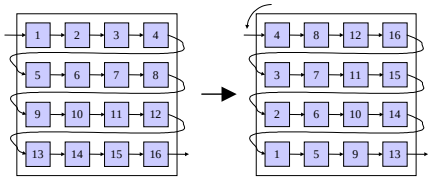


Figure 3: Rotation of block 90° anti-clockwise.

The degree of the interleaver determines both the spread and delay of the interleaver. For the block interleaver the delay, δ_{block}, and spread, s_{block}, are given as,

$$\delta_{block} = d^2 - d \quad and \quad s_{block} = d \quad (14)$$

This shows that increasing the degree of the interleaver increases its ability to disperse bursts of loss, but at the expense of increasing delay.

4 Experimental Results

The experiments in this section first compare the effectiveness of the three lost vector compensation methods. The effect of interleaving is then considered together with an examination into the effect of increasing the interleaving depth.

The recognition task for all experiments is the Aurora connected digit database [1]. Digits are modelled using 16-state, 3-mode HMMs, trained from the set of clean digits. The test set comprises 4004 noise-free digits strings (13,159 digits in total) which gives baseline accuracy of 99% with 95% confidence error bands of +/- 0.38% at 95% accuracy. As per the ETSI standard, two vectors are carried by each packet.

4.1 Lost vector compensation

The effectiveness of the lost vector compensation methods are evaluated on four different channels which were simulated by a 3-state Markov chain [4]. Table 1 shows the conditions of the four channels which vary in terms of the packet loss rate, α, and average burst length, β.

	Packet loss rate, α	*Av. burst length, β*
Channel A	10%	4 packets
Channel B	10%	20 packets
Channel C	50%	4 packets
Channel D	50%	20 packets

Table 1: Simulated channel conditions

Table 2 shows recognition performance of the compensation methods for the channel conditions A to D. For MAP estimation and missing feature theory, the two variants of each technique, as discussed in sections 2.2 and 2.3, are evaluated. For the MAP estimation methods, $\tau = 5$ time instants before and after the lost were used.

Considerable improvements are attained by applying the methods of compensation considered in this work. MAP methods give higher accuracy than cubic interpolation. It can be seen that missing feature theory methods generally out-perform the reconstruction methods, particularly when the average burst length is large (channels *B* and *D)*. Also, when the average burst length is short (channels *A* and *C*) there is a substantial difference between the performances of the two missing feature methods. This is because for these channels, the bursts of lost vectors are positioned closer together, causing the triangular regions removed by the first missing feature method to overlap. Restraining MAP estimation to the

bandlimited case gives slightly higher accuracy than original MAP estimation but at a considerable reduction in complexity.

	A	*B*	*C*	*D*
No compensation	92.2	89.5	50.6	50.3
Cubic interpolation	96.9	91.3	86.1	59.3
MAP	96.9	92.0	86.5	61.5
MAP – bandlimited	97.0	92.0	86.6	61.9
Missing – triangle	96.8	93.1	85.8	67.3
Missing – interpolate	97.5	93.4	90.0	69.8

Table 2: Recognition accuracy with no interleaving

4.2 Interleaving

The experiments in this section now apply interleaving to the feature vector stream prior to transmission. The experimental configuration is identical to that in the previous section and table 3 shows recognition accuracy attained by the various forms of the three compensation methods and a block interleaver (as described in section 3) of $d = 4$.

	A	*B*	*C*	*D*
No compensation	94.1	90.0	54.0	52.4
Cubic interpolation	98.4	93.3	93.0	67.9
MAP	98.3	93.7	93.1	70.1
MAP – bandlimited	98.3	93.8	93.1	70.7
Missing – triangle	97.5	93.9	88.5	68.8
Missing – interpolate	98.4	94.6	94.5	76.2

Table 3: Recognition accuracy with block interleaving

Comparing the interleaved results with the no interleaving results of table 2 shows an increase in recognition accuracy for all compensation methods, the magnitude of which is greater if the average burst length of the channel was large before interleaving. Note that, for the first missing feature method, for channels *A* and *C* (where the average burst length was short before interleaving), a less pronounced increase in performance in observed. This is because the decrease in burst length results in a higher occurrence of overlap.

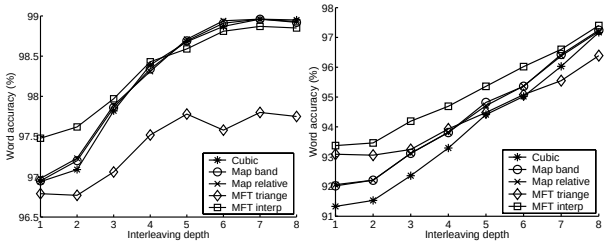


Figure 4: The effect of varying interleaving depth.

An interleaver's ability to disperse packet loss is related to its degree [4]. Figures 4a and 4b show the effect of increasing interleaving depth in the range $d=1$ (corresponding to no interleaving) to $d=8$ on channels *A* and *B* described above. A similar pattern of results is observed as from table 3, in that increasing the interleaving depth results in an increase in word accuracy for all methods, however, this increase is less pronounced for the first missing feature method for the same reasons as out-lined above. In figure 4a a leveling off of accuracy is observed where the depth of the interleaver becomes sufficiently large to fully distribute the bursts of packet loss of channel *A* ($d>6$). However, this is not repeated in figure 4b as an interleaving degree of nearly $d=40$ would be required in order to distribute the longer burst lengths. As indicated by equation 14 this would introduce prohibitively long delays to the system.

5 Conclusions

This work has shown that packet loss can have a severe effect on the accuracy of DSR systems. The methods outlined here give substantial improvements in these conditions. Results suggest that it is more beneficial to compensate for lost vectors in the decoding stage of the recogniser, rather than attempting to reconstruct the feature vector stream beforehand. This is especially true in the presence of large bursts of losses, as the accuracy of such methods falls off as burst length increases.

Of the reconstruction methods, it has been shown that MAP estimation performs significantly better than cubic interpolation. In addition to this, performing MAP estimation using only those values in the same quefrency band results in a slight increase in recognition accuracy whilst greatly decreasing the complexity of the estimation.

With missing feature methods, it is important to consider how to treat corrupted temporal derivates within the feature vector stream. One approach is to remove all temporal derivates effected by the loss of static elements from the feature vector stream. However, substantially improved performance was achieved by calculating the temporal derivatives using a reconstructed version of the static feature vector stream prior to recognition.

Interleaving has given substantial increases in recognition accuracy by dispersing bursts of packet loss. However the degree of the interleaver, and hence its delay, must be carefully considered in the design of such a system.

6 Acknowledgements

The authors gratefully acknowledge the support of the UK Engineering and Physical Sciences Research Council (EPSRC) (grant GR/R88243/01) and the Spanish CICYT Project TIC-2001-3323 in funding this work.

7 References

[1] ESTI document - ES 202 050 – STQ: DSR – Extended advanced front-end feature extraction algorithm, 2003

[2] C.B. Boulis, M. Ostendorf, E.A. Riskin and S. Otterson, "Graceful degradation of speech recognition performance over packet-erasure networks", IEEE Trans. On Speech and Audio Processing, vol. 10, no. 8, pp. 580-590, 2002.

[3] L. Docio-Ferandez and C. Garcia-Mateo, "Distributed speech recognition over IP networks on the Aurora 3 database", Proc. ICSLP, 2002.

[4] A.B. James and B.P. Milner, "An analysis of interleavers for robust speech recognition in burst-like packet loss". Proc. ICASSP, 2004.

[5] T. Endo, S. Kuroiwa and S. Nakamura, "Missing feature theory applied to robust speech recognition on IP networks", Proc. Eurospeech, 2003.

[6] A. Bernard and A. Alwan, "Channel noise robustness for low-bitrate speech recognition", Proc. ICSLP, 2002.

[7] A.M. Gómez et al, "A Source Model Mitigation Technique for Distributed Speech Recognition over Lossy Packet Channels". Proc. Eurospeech, 2003.

[8] B. R. Ramakrishana, "Reconstruction of Incomplete Spectrograms for Robust Speech Recognition". PhD thesis, Carnegie Mellon University, 2000.

[9] B.P. Milner and A.B. James, "Analysis and compensation of packet loss in Distributed Speech Recognition using interleaving". Proc. Eurospeech, 2003

Statistical-based Reconstruction Methods for Speech Recognition in IP Networks

Angel M. Gómez, Antonio M. Peinado, Victoria Sánchez,
Ben P. Milner[*]*, Antonio J. Rubio*

Dpt. Electrónica y Tecnología de Computadores, University of Granada, Spain
[*]School of Computing Sciences, University of East Anglia, UK
ammg@ugr.es

Abstract

This work shows the performance of statistical-based reconstruction techniques when a burst-like packet loss network is used to transmit speech feature vectors on a DSR architecture. Two different approaches to exploit prior information about the speech are outlined. The first models the sequence of quantized vectors through transition probabilities to make estimations based on data-source information, while the second uses prior knowledge of the means and covariances of the feature vector stream to make a maximum a-posteriori (MAP) estimate of lost vectors. These methods provide better results than those obtained by the AURORA nearest repetition, especially in the presence of bursts of losses. However, they require either a notable amount of memory or a high time complexity. Therefore, a novel solution based on the previous methods is proposed and evaluated.

1. Introduction

Since its beginning the Internet has not only been growing in size, incorporating many new networks, but also in functionality, adding new services. As many others, voice services like speech transmission and speech recognition are being integrated into Internet services. Transmission services as Voice over IP (VoIP) offer an alternative to traditional speech transmission systems and a lot of research effort is being devoted to offer reliable speech recognition over IP.

The Distributed Speech Recognition (DSR) [1] approach is very attractive since it is based on a client-server architecture as many other services on the Internet. On the client side, a front-end analyzes, quantizes and packetizes speech data and sends it over a communication channel. On the server side, a remote server, called back-end, receives the speech data and performs speech recognition. Figure 1 illustrates this architecture. In that way, only those parameters which are relevant to the recognition process are transmitted, reducing significantly the data rate. In order to do so, a common front-end must be established. The DSR standard was originally proposed in mobile environments, so a protocol suitable for IP is being developed. IETF Audio Video Transport workgroup, along with Motorola, are working on a standard [2] which will define the real time protocol (RTP) [3] payload for speech recognition. This standard can be found as a draft and uses an equivalent format to DSR for mobile environments.

However, DSR must also deal with the packet losses typical of IP networks. Packet losses are caused by the inability of IP networks to offer a reliable and quality packet delivering service, since these were designed to offer a best effort service. In fact, on congested IP networks, routers will discard packets if their input flow exceeds their output flow for a given data route. In this scenario, a packet loss usually occurs in bursts, where multiple consecutive packets are lost. Besides, new portable devices like PDAs accessing to the Internet through a wireless link are appearing. This link is subject to errors which are reflected at the back-end as lost packets.

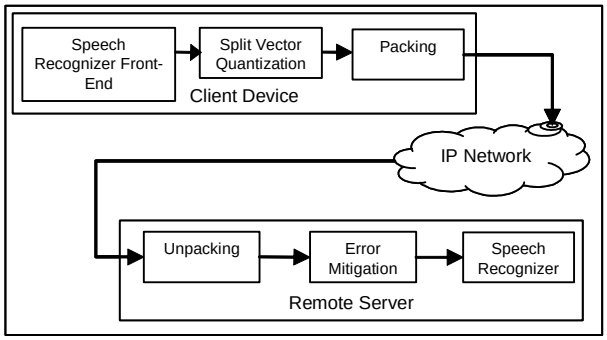


Figure 1: *Block diagram of the system architecture.*

As a consequence of lost packets, the recognition accuracy is diminished and error mitigation techniques are needed. In this work two statistical-based reconstruction methods are evaluated. These methods use prior information about the speech providing a better performance than the usual repetition technique proposed by the standard. The experimental framework is established in section 2. Along with the above mentioned ones, a novel method which joins the advantages of both methods is described in section 3. Finally, simulation results are shown in section 4 and conclusions in section 5.

2. Experimental Framework

2.1. Database, Front-end and Recognizer

To evaluate and compare the mitigation techniques proposed in this paper, the ETSI STQ-AURORA Project Database 2.0 experimental framework was adopted [4]. The speech data has been extracted from clean sentences of the Aurora-2 database (connected digits spoken by American English speakers). Training is performed from a set of 8440 utterances and test is carried out over the 4004 clean sentences of set A.

The front-end used in this work is the one proposed in the ETSI standard [1]. This front-end provides a 14-dimension feature vector containing 13 MFCCs (including the 0th order one) plus log-energy. These features are grouped into pairs and

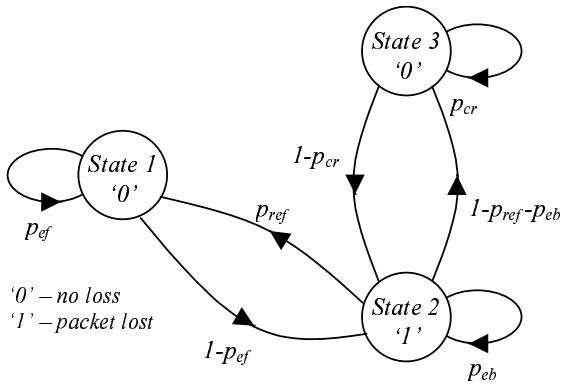


Figure 2: *Three state packet loss model.*

quantized by means of seven Split Vector Quantizers (SVQ). All codebooks have a 64-center size (6 bits), except the one for MFCC-0 and log-Energy, which has 256 centers (8 bits). The recognizer is the one provided by Aurora and uses eleven 16-state continuous HMM word models, (plus silence and pause, that have 3 and 1 states, respectively) with 3 Gaussians per state (except silence, with 6 Gaussians per state).

2.2. Transmission and Channel Model

After the SVQ quantization, each feature pair at time t is represented by a vector $\mathbf{c}_t$ ($\mathbf{c}_t \in \{\mathbf{c}^{(i)}; (i = 0, \ldots, 2^M - 1)\}$) (M=6, 8 in this work). The corresponding SVQ indexes i_t (at time t) are encoded and sent over an IP network.

IP networks send data through datagrams or packets. Packet transmission is performed according to the DSR draft for IP [2]. In order to reduce the transmission overhead, each packet contains at least the speech features from two short-time analysis frames. Thus, when a packet is lost we lose two feature vectors. Although more than a pair of feature vectors can be sent per packet, it is not recommended to do so since the loss of those packets will cause the appearance of longer bursts.

DSR payload frames are embedded on RTP frames which include a sequence number. Thanks to this number, lost packets can be identified and reconstructed. Bolot [5] studied the distribution of packet loss in Internet and concluded that it occurs in bursts and could be approximated by a Markovian loss model. In this work, packet loss is modeled using a three state Markov chain with no explicit duration distribution to model burst lengths [6]. Figure 2 illustrates the model topology. This model allows to independently set the packet loss rate and the average burst length by means of transition probabilities, being specially suitable to evaluate reconstruction algorithms in a wide range of conditions. These conditions will comprise from a packet loss rate from 10% to 50% (with 10% increments) with an average burst lenght of 1, 4, 8, 12 and 20 packets.

3. Statistical-based Reconstruction Techniques

3.1. Data-Source Techniques

Data-source reconstruction techniques are based on the modeling of the data source by means of transition probabilities from a given symbol i (or sequence of them) to another symbol j (or sequence of them). A prediction for a lost sequence of symbols can be accomplished through this model and the received symbols [7].

From the IP network, considered as our data source, we receive quantized speech features, that is, SVQ indexes ($i = 0, \ldots, 2^M - 1$) or symbols which represent a pair of features. From now on, we will focus on the mitigation of a given feature pair (the rest of feature pairs are processed in the same way). When a packet loss of $2B$ length occurs at time $t = 1$, we will use the last received SVQ index before the burst (i_0) and the first one after it (i_{2B+1}), to build two estimations. These estimations try to minimize the minimum mean square error in accordance with the transition probabilities of the model [8]. Assuming the transition probabilities are time independent, the estimation only depends on the received index and the distance in time to it (l), but not on the current time instant t. Therefore, a precalculated sequence of estimations can be stored for each possible symbol i:

$$\begin{aligned} E_F(i) &= (\hat{\mathbf{c}}_1(i), \hat{\mathbf{c}}_2(i), \ldots, \hat{\mathbf{c}}_L(i)) \\ E_B(i) &= (\hat{\mathbf{c}}_{-L}(i), \hat{\mathbf{c}}_{-L+1}(i), \ldots, \hat{\mathbf{c}}_{-1}(i)) \\ \hat{\mathbf{c}}_l(i) &= \sum_{j=0}^{2^M-1} \mathbf{c}^{(j)} P(i_{t+l} = j | i_t = i) \end{aligned} \quad (1)$$

where $E_F(i)$ is the *forward estimation* and $E_B(i)$ is the *backward estimation* for a symbol i, L is the maximum length for the estimation sequence and $\mathbf{c}^{(j)}$ is the feature pair corresponding to the symbol j.

When a burst occurs, the first B speech feature pairs are reconstructed through the forward estimation of i_0 ($E_F(i_0)$) and the last B speech feature pairs are reconstructed through the backward estimation of i_{2B+1} ($E_B(i_{2B+1})$). If B is greater than L, then the estimations from time $t = L$ and $t = 2B - L + 1$ are repeated toward the middle of the burst.

Further results can be obtained if we extend the above mitigation scheme to a N-order data-source model where the estimations depend on several symbols previous to the burst, $E_F(i_0, i_{-1}, \ldots, i_{-N})$, and next to it, $E_B(i_{2B+1}, i_{2B+2}, \ldots, i_{2B+N})$. Although the previous formulas can be easily extended in order to do so, the main problem is to store the sequences of estimations, due to the huge amount of possible combinations of indexes ($(2^M - 1)^N$ with $M = 6, 8$).

However, not all the combinations appear with an equal frequency in the database. In fact, some combinations do not even appear. Therefore, we can solve this problem by storing only those combinations whose appearance frequency in the training database is bigger than a certain threshold μ. This considerably reduces the required amount of precalculated data and also has a beneficial effect because those combinations with a low frequency of appearance often have bad forward/backward estimations (there are less than μ examples to train them).

When a burst appears, we build a *forward reference combination* of length N taking the N indexes previous to the burst. If the first lost vector is at time $t = 1$, then we will take the SVQ indexes $i_0, i_{-1}, \ldots, i_{-N}$. At this step, it is possible that we can not build the reference combination because there are not previous speech vectors at certain time $t = -r$, due to a previous packet loss or an utterance beginning. Since the 0^{th} order interpolation works well as reconstruction technique, we can suppose that the unknown indexes ($i_{-r}, \ldots, i_{-N}$) are copies of the first known index (i_{-r+1}).

Once the forward reference combination has been built, we search it in the table of registers, getting its corresponding forward estimation and replacing the first B speech feature pairs with it. An example diagram of a forward reconstruction with

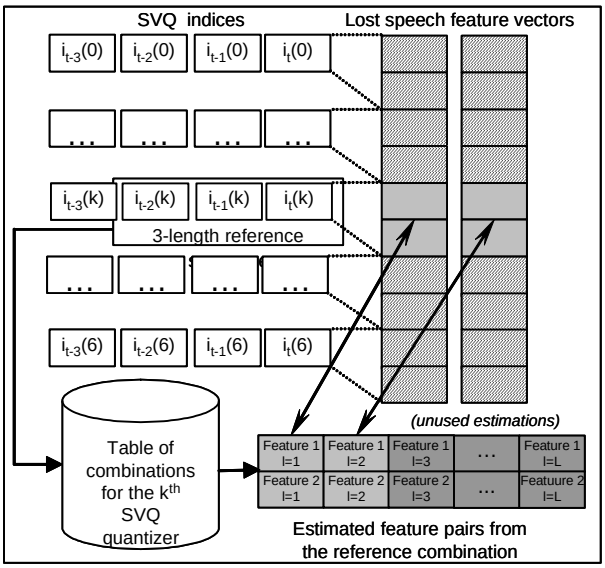


Figure 3: *Forward reconstruction of two feature pairs based on a 3^{th} order source model.*

an $N = 3$ order model can be found in figure 3.

In order to perform the backward reconstruction, we build a *backward reference combination* for each feature pair in an analogous way as we did in the forward case. Then, we search the reference in the table and replace the last B feature pairs with the corresponding backward estimation.

3.2. MAP Estimation

Another approach for estimating the value of lost vectors through statistical information is the maximum a-posteriori (MAP) estimation [9]. In MAP estimation, a sequence of lost vectors, X_m, are calculated in order to maximize their likelihood conditioned on the values of the correctly received vectors, X_o, and the overall distribution of the feature vector stream $P(X; \mu, \sigma)$. Assuming this distribution is Gaussian, the MAP estimation can be simplified to a linear regression [10] given by,

$$\hat{\mathbf{X}}_m = \mu_m + \Sigma_{mo}\Sigma_{oo}^{-1}(\mathbf{X}_o - \mu_o) \tag{2}$$

where μ_m and μ_o are the mean vectors of $\mathbf{X}_m$ and $\mathbf{X}_o$ respectively, Σ_{oo} is the auto-covariance matrix of $\mathbf{X}_o$ and Σ_{mo} is the cross-covariance matrix between $\mathbf{X}_m$ and $\mathbf{X}_o$.

In order to obtain μ_m and Σ_{mo}, wide-sense stationarity is assumed, thus the mean and covariance of the speech features are assumed to be independent of their position in time. Given the k_1^{th} coefficient of the feature vector at time instant t_1, $S(t_1, k_1)$, and the k_2^{th} coefficient of the feature vector at time t_2, $S(t_2, k_2)$, their means and covariance are given as:

$$\begin{aligned} \mu(t_1, k_1) &= E[S(t, k_1)] = \mu(k_1) \\ \mu(t_2, k_2) &= E[S(t, k_2)] = \mu(k_2) \\ E[(S(t_1, k_1) - \mu(k_1))(S(t_2, k_2) - \mu(k_2)] &= \\ c(t_1 - t_2, k_1, k_2) &= c(\tau, k_1, k_2) \end{aligned} \tag{3}$$

where $E[]$ is the expectation operator, $\mu(k)$ is the expected value of each coefficient in a feature vector and $c(\tau, k_1, k_2)$ defines the covariance between any coefficient and any other coefficient τ time instants later in a feature vector sequence. Since both the mean of all the components and the covariance between any two components are known in the feature vector sequence, μ_m and Σ_{mo} can be constructed (and also μ_o and Σ_{oo}^{-1}) and equation 2 can be solved.

Due to the inversion of large covariance matrices (Σ_{oo}^{-1}), in practice not all the received features in the utterance are used in $\mathbf{X}_o$ as this imposes too much of a computational overhead. Instead, $\mathbf{X}_o$ is limited to a few vectors around the region of the loss. However, the more observations are taken into account in the MAP estimation, the better estimations are obtained.

For this reason, the size of the auto-covariance matrix must be reduced by reconstructing each coefficient of the feature vector separately from the most relevant observed coefficients. In particular the DCT operation applied during feature extraction means that coefficients are considerably more correlated along time than quefrency. Therefore, when a burst appears from $t = 1$ to $t = 2B$, each coefficient of the lost vector, $S_l(t, k)$, is reconstructed from received features using the same vector coefficient, i.e., $S_o(-N, k), \ldots, S_o(0, k), S_o(2B + 1, k), \ldots, S_o(2B + N, k)$.

3.3. Hybrid solution

The data-source technique is very fast since the estimations are precalculated and the only operation implied is the search of the reference combinations. The time complexity of the sequential search is $\Theta(n)$, where n is the size of the register table. However, arranging the combinations in the table, we can use a binary search whose complexity is $\Theta(\log(n))$. On the contrary, it needs a large amount of memory. This memory requirement can be partially solved by quantizing the estimations.

On the other hand, the memory requirements of the MAP estimation are reasonably low. However, matrix inversions are required whose time complexity is $\Theta(n^3)$, where n is the number of observations employed in the MAP estimation.

The data-source and MAP techniques can be combined, resulting in a novel technique which not only offers better results, but also requires less memory than the data-source technique and less computation than the MAP estimation.

In the source-model technique, when a certain reference combination did not appear in the table of estimations, 0^{th} order interpolation was applied. This assured us that the technique obtained, in the worst case, at least the same results as the AURORA reconstruction. By using the MAP estimation instead of the 0^{th} order interpolation, it is possible not only to assure better results in the worst case, but also to be more restrictive with the frequency of appearance of the combinations by setting a higher threshold μ. A higher threshold implies a reduction on the amount of memory required but also, since the rejected combinations do not frequently appear in the training database, MAP estimation is only used in a few cases during the reconstruction.

4. Results

Previous techniques have been evaluated under the channel model and conditions described in section 2, that is, at packet loss rates from 10% to 50% and average burst lengths of 1, 4, 8, 12 and 20 packets, with 2 vectors per packet. Table 1 shows the performance on recognition accuracy of the standard reconstruction technique (nearest repetition) whilst table 2 shows the performance of a 3^{th}order source model reconstruction. The length of the sequence of estimations is limited to $L = 40$, while the acceptance threshold have been set to $\mu = 10$, that is, a combination of symbols must appear 10 times in the training

Packet loss rate	Av. burst length				
	1	*4*	*8*	*12*	*20*
10%	99.05	96.72	93.35	92.35	90.91
20%	98.99	93.45	87.95	84.47	82.01
30%	98.98	91.34	82.05	76.64	74.09
40%	98.91	87.69	76.19	69.48	66.47
50%	98.92	84.12	69.48	63.96	57.97

Table 1: *Results on accuracy recognition with standard reconstruction*

Packet loss rate	Av. burst length				
	1	*4*	*8*	*12*	*20*
10%	99.05	97.22	94.64	93.69	92.29
20%	99.02	94.69	90.45	87.27	84.58
30%	99.00	93.22	85.92	80.72	76.98
40%	98.93	90.30	80.77	74.73	70.73
50%	98.90	87.51	75.63	69.56	63.07

Table 2: *Results on accuracy recognition with the Data-Source estimation.*

database in order to store its forward and backward estimation. This reduces the number of stored combinations from 18 million to less than 132K.

Table 3 shows the performance of the MAP estimation. The MAP estimation is computed through the 10 coefficient values in the same quefrency previous to the estimating vector and the 10 values next to it.

Finally, in table 4, the performance of the hybrid technique can be found. The MAP and the data-source parameters are the same as before, except for the threshold μ which has been set to 100 instead of 10. This offers a reduction on the number of stored combinations from 132K to less than 17K.

5. Conclusions

This work addresses the problem of remote speech recognition on packet switched networks as Internet. Due to fact that IP networks were designed to offer a best effort service, they are unable to offer a reliable and quality packet delivering service, causing packet loss.

The DSR approach is very attractive for IP networks since it is based on a client-server architecture as many other services. This work is focused on reconstruction techniques suitable for a DSR architecture which transmits speech feature vectors over a burst-like packet loss network.

Techniques based on the modeling of the data source through transition probabilities and MAP estimation are described in this paper. Both techniques offer a superior performance when compared with the reconstruction based on the nearest repetition. However, the data-source one requires a large amount of memory and the MAP has a high time complexity. On the other hand, source-based reconstruction is a very fast technique whilst MAP estimation requires a reasonably low amount of stored data.

For this reasons, a new hybrid technique is proposed. This techniques offers a trade off between the previous techniques. While it requires less memory and computational resources than the data-source and the MAP reconstruction, respectively, better results are obtained.

Packet loss rate	Av. burst length				
	1	*4*	*8*	*12*	*20*
10%	99.02	97.24	94.76	93.73	92.34
20%	99.05	94.83	90.66	87.50	84.78
30%	99.01	92.98	85.10	80.76	77.45
40%	98.99	90.30	80.58	74.70	71.00
50%	98.92	87.35	75.46	69.21	63.28

Table 3: *Results on accuracy recognition with the MAP estimation.*

Packet loss rate	Av. burst length				
	1	*4*	*8*	*12*	*20*
10%	99.06	97.21	94.89	93.93	92.45
20%	99.00	94.78	90.64	87.67	84.89
30%	98.98	93.30	86.31	81.19	77.59
40%	98.94	90.29	80.80	75.14	71.36
50%	98.91	87.59	75.96	70.47	64.07

Table 4: *Results on accuracy recognition with the Hybrid reconstruction.*

6. Acknowledgements

The authors gratefully acknowledge the support of the Spanish CICYT Project TIC-2001-3323 in funding this work.

7. References

[1] "ETSI ES 201 108 v1.1.2 Speech Processing, Transmission and Quality aspects; Distributed speech recognition; Front-end feature extraction algorithm; Compression algorithms", ETSI Standard, 2000.

[2] "RTP Payload Format for DSR ES 201 108", IETF Audio Video Transport WG, Internet draft, 2002.

[3] "RTP: A Transport Protocol for Real-Time Applications",I ETF Audio Video Transport WG, RFC3550, 2003.

[4] H.G. Hirsch and D. Pearce: "The AURORA experimental framework for the performance evaluation of speech recognition systems under noisy conditions", ISCA ITRW ASR2000, Sept. 2000.

[5] J.C. Bolot: "End-to-end frame delay and loss behavior in the Internet", *Proc. ACM SIGCOMM*, Sept. 1993.

[6] A.B. James and B.P. Milner: "An analysis of interleavers for robust speech recognition in burst-like packet loss", *Proc. ICASSP*, 2004.

[7] A.M. Gómez et al: "A Source Model Mitigation Technique for Distributed Speech Recognition over Lossy Packet Channels", *Proc. Eurospeech*, 2003.

[8] A.M. Peinado, et al: "HMM-Based Channel Error Mitigation and its Aplication to Distributed Speech Recognition", *Speech Communication*, 2003.

[9] A. James, A. Gómez and B. Milner: "A Comparison of Packet Loss Compensation Methods and Interleaving for Speech Recognition in Burst-Like Packet Loss", Accepted in ICSLP 2004.

[10] B. R. Ramakrishana: "Reconstruction of Incomplete Spectrograms for Robust Speech Recognition". PhD thesis, Carnegie Mellon University, 2000.

PACKET LOSS CONCEALMENT BASED ON VQ REPLICAS AND MMSE ESTIMATION APPLIED TO DISTRIBUTED SPEECH RECOGNITION

Antonio M. Peinado, Ángel M. Gómez, Victoria Sánchez, José L. Pérez-Córdoba, Antonio J. Rubio

Dpt. de Teoría de la Señal, Telemática y Comunicaciones
Universidad de Granada, Spain
{amp,amgg,victoria,jlpc,rubio}@ugr.es

ABSTRACT

This paper proposes a new packet loss concealment technique based on the inclusion in each packet of a few FEC bits, representing data replicas, combined with a minimum mean square error estimation (MMSE). This technique is developed for an Aurora-2 distributed speech recognition system working over an IP network. In addition to the data representing the transmitted speech frames, each packet includes some FEC bits representing a strongly VQ-quantized version (replicas) of previous and subsequent frames. When a loss burst occurs, the lost frames can be reconstructed from the VQ replicas. In order to mitigate the degradation introduced by the coarse VQ quantization of the replicas, a model-based MMSE estimation is applied. The experimental results show that, under a strongly degraded channel, it is possible to obtain up to 83.31 % of word accuracy with only 4 FEC bits or 88.47 % with 8 FEC bits per packet, when the Aurora mitigation algorithm only obtains 76.98 %.

1. INTRODUCTION

When transmitting speech data over a packet network one of the most common problems found is packet loss. Packet losses introduce audio distortions that cause perceived voice quality degradation in the case of IP telephony and a reduction of performance in other speech-based services such as Distributed Speech Recognition [1]. Many packet loss recovery techniques have been proposed which can be broadly classified into two classes: sender based techniques and receiver based techniques [2]. Among the first ones, we have forward error correction (FEC), where repair information is transmitted so that a lost packet can be recovered from that repair data, and interleaving. Among the second class, we have frame repetition, interpolation or more sophisticated regeneration techniques based on signal models.

In this paper we focus on the FEC approach and propose a technique that with very few overhead bits combined with Hidden Markov Model (HMM) based forward-backward minimum mean squared error estimation (FBMMSE) [4] obtains a significant improvement in the performance of a distributed speech recognition system based on the ETSI standard [1] for adverse IP channel conditions. The FBMMSE technique was originally proposed by us in a wireless channel context, where it effectively mitigated wireless channel errors. In the context of the FEC technique for IP transmission, the FBMMSE estimation will be reformulated and used to mitigate the distortion introduced by the coarse quantization of the redundant information included in the proposed FEC scheme.

Work supported by MEC/FEDER project TEC2004-03829/TCM.

The paper is organized as follows. First, the experimental framework is described. Then, we present a review of MMSE estimation. In section 4 we explain the proposed technique in detail and present experimental results. We conclude with section 5, where we discuss the payload format for the ETSI DSR standard and indicate several solutions to introduce the proposed FEC overhead bits.

2. EXPERIMENTAL FRAMEWORK

The front-end utilized in this work is the one proposed in the ETSI standard [1] and developed by the Aurora working group. This front-end segments the speech signal into overlapped frames of 25 ms and provides a 14-dimension feature vector (per frame) containing 13 Mel Frequency Cepstrum Coefficients (MFCC) ($C(k)$ $(k = 0, \ldots, 12)$) plus log-Energy ($\log E$). These features are grouped into pairs and quantized by means of seven Split Vector Quantizers (SVQ). All codebooks have a 64-center size (6 bits), except the one for MFCC-0 and log-Energy, which has 256 centers (8 bits). The bitstream is generated by grouping frames into pairs (88 bits) that are protected by a 4-bit CRC. The Aurora packet loss mitigation algorithm can be summarized as follows: once a loss burst, containing 2B frames, is detected, the first B frames are substituted by the last received frame before the burst and the last B ones by the first received frame after the burst.

The recognizer is the one provided by Aurora-2 and uses eleven 16-state continuous HMM word models (plus silence and pause, that have 3 and 1 states, respectively) with 6 gaussians per state. The training and testing data are extracted from the Aurora-2 speech database. Training is performed with 8440 clean sentences and test is carried out over set *A* (4004 clean sentences distributed into 4 subsets). The performance of the recognition system will be measured in terms of the Word Accuracy (WAcc). The Wacc values obtained with this system are 99.02 % (without quantization) and 99.04 % (after SVQ quantization).

The transmission channel has been modeled by a Gilbert model [3], which is used to simulate six different channel conditions that are summarized in table 1, where *clp*, *ulp* and d_{av} are the packet loss probability when the previous packet has been already lost, the *a priori* probability of a packet loss and the mean loss burst duration (in number of packets), respectively. We will consider that each packet contains two frames (one frame pair).

# Condition	clp	ulp	d_{av}
1	0.147	0.006	1.172
2	0.330	0.090	1.492
3	0.500	0.286	2.000
4	0.600	0.385	2.500
5	0.700	0.500	3.333
6	0.800	0.550	5.000

Table 1. *Description of the simulated channel conditions.*

3. REVIEW OF MMSE ESTIMATION

In our previous work [4] we showed that the MMSE estimation is a powerful technique to mitigate the errors introduced by a wireless channel. In this section we present a brief review of this technique under a formulation suitable for the considered application. The estimation is performed on a feature pair basis, since this is the encoding unit used in the Aurora standard. After the SVQ quantization [1], each feature pair is represented by a vector $\mathbf{c}$ ($\mathbf{c} \in \{\mathbf{c}^{(i)}; i = 0, \ldots, 2^M - 1\}$) ($M$=6,8 in this work). We consider that, at the back-end, the received vector $\hat{\mathbf{c}}$ can be affected by some type of distortion. We also consider that this distortion has a bursty characteristic affecting $T-1$ frames, corresponding $t = 0$ and $t = T$ to the last and first correctly received vectors before and after an error burst, respectively. The MMSE estimation of the received parameter vector at time t, which considers the previous and subsequent received vectors, is obtained as,

$$\tilde{\mathbf{c}}_t = E[\mathbf{c}_t|\hat{\mathbf{c}}_0, \hat{\mathbf{c}}_1, \ldots, \hat{\mathbf{c}}_T] = \sum_{i=0}^{2^M-1} \mathbf{c}^{(i)} \gamma_t(i) \quad (0 < t < T) \tag{1}$$

with

$$\gamma_t(i) = P(\mathbf{c}_t^{(i)}|\hat{\mathbf{c}}_0, \hat{\mathbf{c}}_1, \ldots, \hat{\mathbf{c}}_T) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=0}^{2^M-1} \alpha_t(j)\beta_t(j)}$$

$$\alpha_t(i) = P(\mathbf{c}_t^{(i)}|\hat{\mathbf{c}}_0, \hat{\mathbf{c}}_1, \ldots, \hat{\mathbf{c}}_t)$$

$$\beta_t(i) = P(\hat{\mathbf{c}}_{t+1}, \ldots, \hat{\mathbf{c}}_T|\mathbf{c}_t^{(i)})$$

where $\alpha_t(i)$ and $\beta_t(i)$ are the forward and backward conditional probabilities, respectively. We have also expressed $\mathbf{c}_t = \mathbf{c}^{(j)}$ as $\mathbf{c}_t^{(j)}$ for notation simplicity. The generation of each quantized feature pair is modeled by an HMM model with transition probabilities $a_{ij} = P(\mathbf{c}_t^{(j)}|\mathbf{c}_{t-1}^{(i)})$ and observation probabilities $b_i(\hat{\mathbf{c}}_t) = P(\hat{\mathbf{c}}_t|\mathbf{c}^{(i)})$. The conditional probabilities can be computed from the following forward and backward recursions,

$$\alpha_t(i) = \left[\sum_{j=0}^{2^M-1} \alpha_{t-1}(j) a_{ji}\right] b_i(\hat{\mathbf{c}}_t)/K_t \quad (t > 0) \tag{2}$$

$$\beta_t(i) = \sum_{j=0}^{2^M-1} a_{ij} b_j(\hat{\mathbf{c}}_{t+1}) \beta_{t+1}(j) \quad (t < T) \tag{3}$$

where K_t is a normalization factor at time t. The following initial conditions are applied to $(t = 0)$ and $(t = T)$,

$$\alpha_0(i) = P_i b_i(\hat{\mathbf{c}}_0)/K_0 \qquad \beta_T(i) = 1 \tag{4}$$

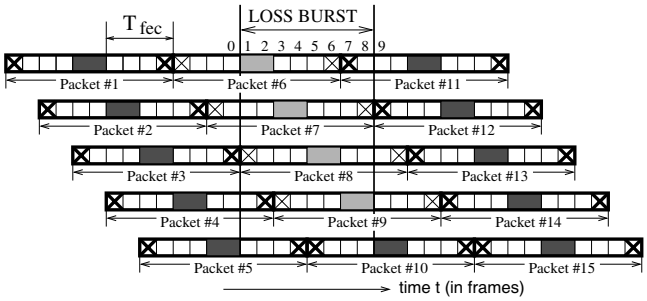


Fig. 1. *Each frame pair (filled with gray) is sent along with a FEC code containing information about the frames (marked with ×).*

where P_i is the *a priori* probability of $\mathbf{c}^{(i)}$.

In our previous work we named this technique as forward-backward MMSE (FBMMSE) estimation in order to remark the use of past and future frames by introducing the forward and backward probabilities and a decoding delay.

4. USE OF VQ REPLICAS AND MMSE ESTIMATION

The problem of applying the FBMMSE estimation described above to the case of a lossy packet channel is that no information is received from the channel during a packet loss period (that is, we do not have vectors $\hat{\mathbf{c}}_t$), what would make the FBMMSE technique useless. A way of solving this problem would be the introduction of information (as FEC bits) about previous and subsequent frames in each packet. Then, there would be some information available about the lost frames during a loss burst. Besides, we would be breaking the burst into shorter bursts and, therefore, increasing the performance. This fact can be easily shown by means of the following experiment. Let us suppose that, along with the feature vectors corresponding to the current frame pair, we also include in the packet exact replicas of the feature vectors corresponding to the frame located T_{fec} frames before the current frame pair and to the frame located T_{fec} frames after the current frame pair. This is depicted in figure 1 for $T_{fec} = 4$. The replicas are marked with the symbol ×. Frames not marked (in white) are not included in the packet. Each packet would then be composed of four frames. The numbering assigned to packets indicates the order in which the packets are sent, while time t is measured in frames. The frame pairs associated to packets lost during a loss burst are indicated in light gray (packets 6, 7, 8 and 9). It can be seen that by using this scheme not all the frames corresponding to lost packets are lost during a burst (frames 2, 4, 5 and 7, marked with bold ×, are recovered). For those frames that are definitively lost (frames 1,3,6 and 8; marked with weak ×), the following mitigation algorithm is applied: for each time $t \leq B$ of a loss burst of length $2B$, the last feature vector received (original or replica) is repeated forwards until a new feature vector is received. For the second half of the burst a similar operation is performed backwards.

The results of this experiment (AURORA+) for $T_{fec} = \{6, 10\}$ are shown and compared with the basic Aurora mitigation algorithm in figure 2. They illustrate the utility of breaking the bursts into shorter ones. This "breaking" idea has been previously and successfully applied for DSR in a different way by performing frame interleaving [5].

However, sending all this additional redundancy increases the bandwidth requirements and, therefore, the loss rate. In order to

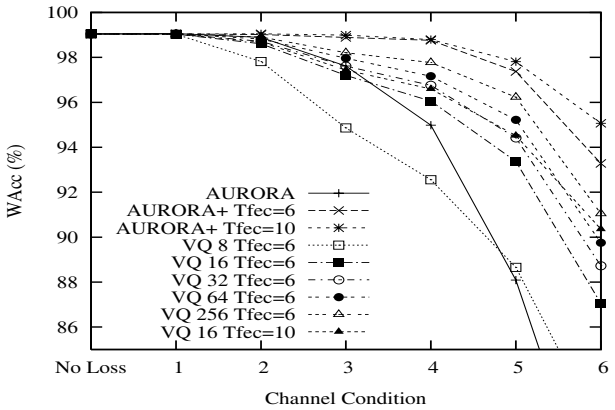


Fig. 2. *Comparison of different mitigation procedures: Aurora, Aurora+ (Tfec=6,10).*

maintain the final bit-rate within a reasonable limit, the repeated feature vectors can be VQ-quantized using a codebook that includes all features (13 MFCCs + logE) with 2^N centroids (N bits). We will use the following distance measure for the codebook design (k-means is used) and in the quantization process,

$$d_W(\mathbf{x}_t, \mathbf{x}_r) = \frac{\sum_{k=1}^{12} (C_t(k) - C_r(k))^2}{\sigma_C^2} + \frac{(C_t(0) - C_r(0))^2}{\sigma_{C\,0}^2} + \frac{(\log E_t - \log E_r)^2}{\sigma_{\log E}^2} \quad (5)$$

where $\mathbf{x}$ represents a 14-dimension feature vector, σ_C^2 is the sum of the MFCC (1-12) variances, and σ_{C0}^2 and $\sigma_{\log E}^2$ are the variances of $C(0)$ and $\log E$, respectively. Each packet should include the following information:

1. 88 bits corresponding to the SVQ-quantized features of the current frame pair.
2. $2 \times N$ bits corresponding to the VQ-replicas.

At the back-end, the VQ replicas can be directly used to improve the recognition. In the case of a feature vector definitively lost, we apply the same mitigation algorithm used in the experiment AURORA+ (that is, repetition of original SVQs or VQ replicas). The results of this strategy are also depicted in figure 2 (experiments VQ) for different VQ codebook sizes (8, 16, 32, 64 and 256 centroids) and $T_{fec} = 6$. It is observed that this technique can provide results similar or better than Aurora, for all channel conditions, using a VQ codebook size of 32 centers (5 bits) or higher, for $T_{fec} = 6$. By increasing the delay T_{fec}, it is possible to improve the performance. As an example, figure 2 also shows that the performance of Aurora is achieved with a VQ codebook of 16 centers and $T_{fec} = 10$ for channel conditions 1, 2 and 3, and is considerably improved for conditions 4 and 5.

Although we have shown that the VQ replicas are useful by themselves, they can be further exploited by performing an FB-MMSE estimation, since we have now information about some of the lost frames. In order to do this, we will divide the received VQ replicas $\mathbf{x}$ into feature pairs $\hat{\mathbf{c}}$. The transition probabilities $a_{ij} = P(\mathbf{c}_t^{(j)} | \mathbf{c}_{t-1}^{(i)})$ of the HMM model are determined from the training data as in [4]. The main difference from the wireless case

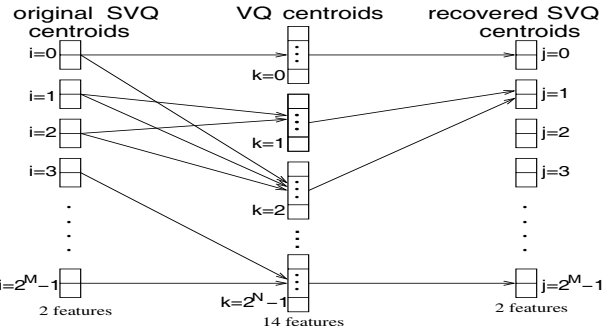


Fig. 3. *Example of the sequence of quantizations applied to the replicas corresponding to one of the SVQ feature pairs.*

treated in [4] resides in the calculation of the observation probabilities $b_i(\hat{\mathbf{c}}_t) = P(\hat{\mathbf{c}}_t | \mathbf{c}^{(i)})$, as the distortion of the received $\hat{\mathbf{c}}_t$ is now due to a strong VQ process and not due to the wireless channel transmission errors. The determination of the observation probabilities $b_i(\hat{\mathbf{c}}_t)$ at each time t ($0 \leq t \leq T$) will be determined, depending on the case, in the following way:

1) In the case that vectors at times $t = 0$ or $t = T$ have been correctly received, the corresponding observation probabilities must be set as,

$$b_i(\hat{\mathbf{c}}_0), b_i(\hat{\mathbf{c}}_T) = \begin{cases} 0 & \mathbf{c}^{(i)} \neq \hat{\mathbf{c}}_0, \mathbf{c}^{(i)} \neq \hat{\mathbf{c}}_T \\ 1 & \mathbf{c}^{(i)} = \hat{\mathbf{c}}_0, \mathbf{c}^{(i)} = \hat{\mathbf{c}}_T \end{cases} \quad (6)$$

2) In the case there is only available a VQ replica at time t ($0 \leq t \leq T$), we will divide the received vector $\mathbf{x}$ into feature pairs that are SVQ quantized again obtaining $\hat{\mathbf{c}}_t$ (as mentioned previously, the SVQ quantization does not involve any reduction of the recognition performance). This process is illustrated in figure 3. Then, we have to determine $b_i(\hat{\mathbf{c}}_t) = P(\hat{\mathbf{c}}_t | \mathbf{c}^{(i)})$. In order to do that, we can see that an original SVQ centroid can correspond to several VQ centroids (depending on the other features different from the considered feature pair). Also, each VQ centroid corresponds to one recovered SVQ centroid, although the contrary can be false (specially when we use a large VQ codebook). This scheme involves the use of a discrete HMM in the FBMMSE estimation, where the observation probabilities $b_i(\hat{\mathbf{c}}_t) = P(\hat{\mathbf{c}}_t | \mathbf{c}^{(i)})$ are obtained from the training database as frequencies of appearance as,

$$b_i(\hat{\mathbf{c}}_t = \mathbf{c}^{(j)}) = \frac{\text{No. recovered symbol } j \text{ given original } i}{\text{No. original symbol } i} \quad (7)$$

It would be also possible to model these observation probabilities by probability density functions (the HMM model would be continuous and the second SVQ process would be unnecessary) where we should select a suitable parametric form for the corresponding *pdf*'s. The discrete version has been selected for simplicity.

3) The third case occurs when there is not any information available at time t ($0 \leq t \leq T$) from the channel. In this case, the forward-backward algorithm progresses without using the observation probabilities, or, equivalently, by considering that the VQ codebook has 1 center (0 bits transmitted). In this case, $b_i(\hat{\mathbf{c}}_t) = 1$ for all i.

Tables 2 and 3 show the word accuracies obtained with the proposed mitigation techniques for several codebook sizes and T_{fec} = $\{6, 10\}$. The best results correspond, as expected, to the FB-MMSE (FB) technique, that obtains excellent results even with

Chan	Mit	Codebook Size				AUR
		4	16	64	256	
1	VQ	98.94	99.03	99.04	99.02	99.06
	FB	99.05	99.05	99.04	99.04	
2	VQ	97.23	98.60	98.81	98.90	98.88
	FB	99.03	99.04	99.04	99.05	
3	VQ	92.67	97.19	97.96	98.20	97.61
	FB	98.14	98.58	98.78	98.76	
4	VQ	89.18	96.05	97.15	97.77	94.98
	FB	96.58	97.63	98.11	98.32	
5	VQ	83.36	93.36	95.22	96.22	88.08
	FB	92.29	95.32	96.00	96.44	
6	VQ	73.94	87.05	89.74	91.04	76.98
	FB	83.31	88.47	90.00	90.98	

Table 2. *Word Accuracy obtained by VQ and FBMMSE (for $T_{fec} = 6$) in comparison with Aurora (AUR) for different channel conditions (Chan).*

Chan	Mit	Codebook Size				AUR
		4	16	64	256	
1	VQ	98.94	99.03	99.04	99.02	99.06
	FB	99.05	99.05	99.04	99.04	
2	VQ	97.12	98.67	98.83	98.91	98.88
	FB	98.99	99.07	99.05	99.07	
3	VQ	92.92	97.43	98.04	98.32	97.61
	FB	98.22	98.71	98.86	98.80	
4	VQ	89.98	96.59	97.43	97.92	94.98
	FB	96.84	98.02	98.35	98.61	
5	VQ	84.67	94.51	95.91	96.72	88.08
	FB	93.01	96.37	96.99	97.60	
6	VQ	77.52	90.35	92.32	93.56	76.98
	FB	85.37	91.73	93.39	94.43	

Table 3. *Word Accuracy obtained by VQ and FBMMSE (for $T_{fec} = 10$) in comparison with Aurora (AUR) for different channel conditions (Chan).*

codebook sizes as low as 4 or 16. The differences between VQ and FBMMSE tend to diminish as the codebook size is increased. This fact is more noticeable for $T_{fec} = 6$ and is the logical consequence of having long gaps in the middle of the loss bursts, in which case, the mitigations tends to the Aurora algorithm in the case of VQ, and to an estimation with uniform distributions for the observation probabilities in the case of FBMMSE (the obtained estimate would only depend on the source model through the transition probabilities a_{ij}).

5. PAYLOAD FORMAT AND IMPLEMENTATION

In this paper, we have proposed a FEC technique that uses data replicas and MMSE estimation to mitigate the effect of packet losses in a DSR system, obtaining excellent results even with very few FEC bits. In this section we will follow the recommendation of reference [6] regarding the payload format for the DSR standard and propose several solutions to introduce the FEC bits that allow the implementation of the proposed technique. Taking into account this recommendation and the constraint of one frame pair per packet, the payload format is the one depicted in figure 4. Packets are aligned into words of 32 bits. As a result, there are 4 free bits that are filled with zeros in [6].

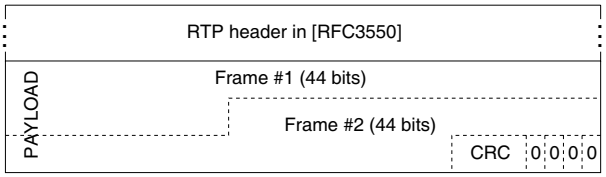


Fig. 4. *Payload format for the DSR standard with one frame pair per packet.*

This payload format suggests us several ways of including the FEC information required for the application of the proposed mitigation methods:
1) We can use the free four bits to introduce two VQ replicas quantized with a 4-center codebook (2 bits/replica). As we can see in tables 2 and 3, in this case it is necessary to apply FBMMSE in order to improve the Aurora results.
2) The system performance can be meaningfully improved if we could reuse the 4 bits devoted to the CRC code to introduce more FEC bits. In this case the replicas are quantized with a 16-center codebook (4 bits/replica). We should use again FBMMSE to improve the Aurora results. However, in this case the VQ technique only provides worse results than Aurora for channel conditions 2 and 3. Since condition 3 corresponds to an average burst duration of 4 frames, a possible solution for mitigation would be a combination of Aurora and VQ. Then, the 2 first and 2 last frames of the burst would be mitigated according to Aurora. The inner $2B - 4$ frames would be mitigated with the VQ technique.
3) Any other increase of FEC would require to include a new 32-bits word in the packet. This case opens a number of new ways for mitigation such as the introduction of 16-bits replicas, the use of 8 bits for the MFCCs and 8 bits for the Energies ($MFCC(0)$ and $\log E$), or even the introduction of four 8-bit replicas. Obviously, these approaches would produce as good or better results than the (best) case of 2 VQ replicas of 8 bits (256 centers) since more information is available.

6. REFERENCES

[1] ETSI ES 201 108 v1.1.2, 2000. Distributed Speech Recognition; Front-end Feature Extraction Algorithm; Compression Algorithms. April 2000.

[2] Perkins et al: "A survey of packet loss recovery techniques for streaming audio". *IEEE Network*, vol. 18, pp. 40-48, Sept. 1998.

[3] A.M. Gómez, A.M. Peinado, V. Sánchez, A. Rubio: "A Source Model Mitigation Technique for Distributed Speech Recognition over Lossy Packet Channels". *Proc. of Eurospeech-2003*, pp. 2733-36, Geneve, Sept. 2003.

[4] A.M. Peinado, V. Sanchez, J.L. Perez-Cordoba, A. de la Torre: "HMM-Based Channel Error Mitigation and its Application to Distributed Speech Recognition". *Speech Communication*, vol. 41/4, pp. 549-561, Nov. 2003.

[5] B.P. Milner, A.B. James: "Analysis and Compensation of Packet Loss in Distributed Speech Recognition Using Interleaving". *Proc. of Eurospeech-2003*, pp. 2693-37, Geneve, Sept. 2003.

[6] Q. Xie: RFC 3557 - RTP Payload Format for European Telecommunications Standard ES 201 108 Distributed Speech Recognition Encoding. July 2003.

Interleaving and MMSE Estimation with VQ Replicas for Distributed Speech Recognition over Lossy Packet Networks

Angel M. Gómez, Antonio M. Peinado, Victoria Sánchez,
José L. Carmona, Antonio J. Rubio

Department of Signal Theory, Networking and Communications
University of Granada, Spain
amgg@ugr.es

Abstract

In this work we evaluate the performance of MMSE estimation with a media-specific FEC based on VQ replicas in comparison with MAP estimation and interleaving, both operating in a DSR system over a loss-prone packet switched network. Both schemes combine a sender-driven with a receiver-based technique and, as we show, clearly outperform the standard Aurora mitigation. However, as independent techniques, interleaving and FEC codes could be jointly applied. Although this would provide better results, a direct combination of FECs and interleaving involves a sum of the delays of both operations. In this work, we introduce a double stream-based strategy that avoids this sum of delays.

Index Terms: distributed speech recognition, loss-prone channels, forward error correction, interleaving, error concealment, maximum a posteriori estimation.

1. Introduction

Packet losses characterize most packet switched networks and can introduce significant limitations to performing Distributed speech recognition (DSR) [1]. Moreover, packet losses tend to appear in bursts and, in DSR, this burst-like nature causes the most negative impact. Thus, DSR has shown to be tolerant to high loss ratios ($\sim$50%) as long as the average burst length is reasonably short (one or two frames) [2].

Media-specific FECs techniques can be especially useful to increase robustness against such losses. These techniques replicate each feature vector in another packet. Indeed, *replicas* can be used not only to recover some lost frames, but also to break bursts of losses into shorter bursts [3]. Since short bursts are better reconstructed, the recognition performance can be improved. However, in order to keep the redundant data into a reasonable size, replicas must be strongly quantized. Exact replacements for lost packets are not obtained and, therefore, an important part of the success of this scheme will depend on the error concealment (EC) technique which manages these degraded replicas.

Alternatively, robustness against bursts can be also increased by applying an *interlaver* prior to transmission. By means of a reordering of the feature vectors, interleaving reduces the perceived burst length at the receiver, improving the recognition performance. As FEC codes, interleaving causes a delay in the transmission but, as advantage, it does not increase the required bandwidth.

In this work, we evaluate a previously proposed FEC-based technique, the MMSE estimation with vector quantized (VQ) replicas [3], in comparison with an interleaver successfully applied to DSR, the optimal delay block interleaver [2, 4]. As we will show, the results individually obtained by these techniques can be improved if both are jointly applied. However, a direct composition of these techniques results in a sum of their delays. In this work, we propose an scheme whereby this increase of delay is avoided.

Work supported by MEC/FEDER project TEC2004-03829/TCM.

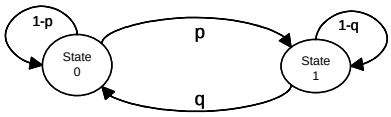


Figure 1: *2-state Markov model. State 0 is error free and state 1 causes frame erasure.*

2. Experimental framework

The experimental setup is based on the framework proposed by the ETSI STQ-Aurora working group [5]. On the client side, the Aurora DSR front-end segments the speech signal into overlapped frames of 25 ms every 10 ms. Each speech frame is represented by a 14-dimensional feature vector containing 13 MFCCs (including the 0th order one) plus log-Energy. These features are grouped into pairs and quantized by means of seven Split Vector Quantizers (SVQ). All codebooks have a 64-center size (6 bits), except the one for MFCC-0 and log-Energy, which has 256 centers (8 bits). IP packets are generated according to the recommendations of the RTP payload format for DSR [6], where at least two frames (one frame pair) per packet are transmitted in order to avoid too high a network overhead due to headers. Following the RFC recommendations, one frame pair per packet is sent.

The recognizer is the one provided by Aurora [5] and uses eleven 16-state continuous HMM word models, (plus silence and pause, which have 3 and 1 states, respectively), with 3 gaussians per state (except silence, with 6 gaussians per state). The training and testing data are extracted from the Aurora-2 database (connected digits). Training is performed with 8440 clean sentences and testing is carried out over set *A* (4004 clean sentences distributed into 4 subsets).

The channel burstiness exhibited by IP communications is modeled by a 2-state Markov model [7], also known as a Gilbert-Elliot model. Figure 1 depicts this model, where p is the probability of the next packet being lost, provided the previous one has arrived; and q is the probability of the next packet not being lost, given that the previous one was lost. These parameters can be set

in accordance with an average burst length (L_{loss}) and a loss ratio (R_{loss}). The frame numbering included in the RTP header will be used to rearrange the received packets and to detect the frame losses.

3. MMSE estimation with VQ replicas

In a previous paper [3], we introduced a simple media-specific FEC technique that, with very few overhead bits, obtained very good results when combined with a powerful EC algorithm, the forward-backward MMSE estimation (FB-MMSE) [8]. In the proposed FEC scheme, each packet is composed of four frames. Along with the current frame pair, VQ-quantized versions of the feature vectors corresponding to the frames located T_{fec} frames before and after it are included in the packet. These VQ replicas are chosen from a codebook of N bits, which is obtained by a k-means algorithm using the following weighted distance measure:

$$d_W(\mathbf{x}_r, \mathbf{x}_s) = \frac{\sum_{k=1}^{12} (c_r(k) - c_s(k))^2}{\bar{\sigma}_c^2} \quad (1)$$

$$+\frac{(c_r(0) - c_s(0))^2}{\sigma_{c_0}^2} + \frac{(\log E_r - \log E_s)^2}{\sigma_{\log E}^2} \quad (2)$$

where $\mathbf{x} = (c(0), \ldots, c(12), \log E)$ represents the 14-dimension feature vector, $\bar{\sigma}_c^2$ is the average of MFCCs(1-12) variances, and $\sigma_{c_0}^2$ and $\sigma_{\log E}^2$ are the variances of $c(0)$ and log$E$, respectively.

These replicas could be directly used but, as we mentioned before, they can be further exploited by applying an FB-MMSE estimation. In order to do so, we will work on a feature pair basis (the encoding unit of the standard). After the SVQ quantization [5], each feature pair is represented by a vector $\mathbf{c}$ ($\mathbf{c} \in \{\mathbf{c}^{(i)}; i = 0, \ldots, 2^M - 1\}$) ($M$=6, 8 in this work). We consider that, at the back-end, the received vector $\hat{\mathbf{c}}$ can be affected by some type of distortion. We also consider that this distortion has a bursty characteristic affecting $T-1$ frames, corresponding $t = 0$ and $t = T$ to the last and first correctly received vectors before and after an error burst, respectively.

FB-MMSE estimation is based on an HMM model of speech (further details can be found in [8]). In order to apply it, the transition and observation probabilities of the model, a_{ij} and $b_i(\hat{\mathbf{c}}_t)$ respectively, must be obtained. The transition probabilities a_{ij} can be determined from the training database as in [8]. Regarding the observation probabilities $b_i(\hat{\mathbf{c}}_t) = P(\hat{\mathbf{c}}_t|\mathbf{c}^{(i)})$, we will consider that all feature pairs during a burst $(\hat{\mathbf{c}}_1, \ldots, \hat{\mathbf{c}}_{T-1})$ have been received. These will be determined depending on the type of feature vector considered (received, replica or definitively lost):

- The observation probabilities corresponding to vectors at time $t = 0$ and $t = T$ (assuming they have been received) must be set as,

$$b_i(\hat{\mathbf{c}}_0), b_i(\hat{\mathbf{c}}_T) = \begin{cases} 0 & \mathbf{c}^{(i)} \neq \hat{\mathbf{c}}_0, \mathbf{c}^{(i)} \neq \hat{\mathbf{c}}_T \\ 1 & \mathbf{c}^{(i)} = \hat{\mathbf{c}}_0, \mathbf{c}^{(i)} = \hat{\mathbf{c}}_T \end{cases} \quad (3)$$

- In the case where a VQ replica is available at time t ($0 \leq t \leq T$), it is divided into feature pairs that are again SVQ quantized, obtaining $\mathbf{c}_t^{(j)}$, as Figure 2 illustrates. It is observed that a recovered SVQ centroid can correspond to several VQ centroids, which can also correspond to several original SVQ centroids. Therefore, given an original SVQ centroid $\mathbf{c}^{(i)}$ we can observe several recovered SVQ centroids $\mathbf{c}^{(j)}$ after the double quantization process. It is then possible to determine the observation probabilities from the training database as frequencies of appearance, as follows,

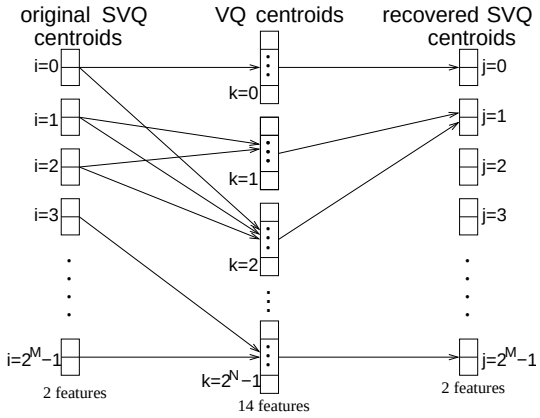


Figure 2: *Example of the sequence of quantizations applied to the replicas corresponding to one of the SVQ feature pairs.*

$$b_i(\hat{\mathbf{c}}_t = \mathbf{c}^{(j)}) = P(\mathbf{c}^{(j)}|\mathbf{c}^{(i)}) = \frac{\text{no. } c^{(j)} \text{ given original } c^{(i)}}{\text{no. original symbol } c^{(i)}} \quad (4)$$

This scheme implicitly involves the use of a discrete HMM in the FB-MMSE estimation. It would also be possible to model these observation probabilities by probability density functions and the second SVQ process would be then unnecessary. However, since SVQ quantization does not involve any reduction in recognition performance [8, 1], the discrete version has been chosen for simplicity.

- Finally, when neither an SVQ vector nor a VQ replica is available at time t ($0 \leq t \leq T$), a degenerated VQ quantization with 0 bits is assumed. Thus, all the original SVQ centroids correspond to only one VQ centroid, the overall mean feature vector, and the observation probabilities are assigned as $b_i(\hat{\mathbf{c}}_t = \mathbf{c}^{(j)}) = 1, \forall i, j$. In this case, the forward-backward algorithm mainly progresses guided by the transition probabilities as if the observation probabilities were not used.

This combined technique can significantly improve the robustness against packet losses even with only a few overhead bits. Table 1 shows the results obtained by this scheme using only 4 bits per replica (16 VQ centroids) in comparison with the Aurora standard mitigation (based on the repetition of the nearest received vector). Different delays are considered, corresponding to different values of T_{fec} ($T_{fec} = 6, 12, 20, 30$). It should be noted that by reusing the 8 bits devoted to CRC (only included for compatibility purposes, since IP protocols include their own error protection schemes) and the zero padding bits [6], it is possible to introduce these replicas without any actual bandwidth increase.

4. Interleaving and MAP estimation

An alternative way to break bursts into shorter ones is to permute the order in which complete frames are transmitted. As a consequence, when frames are restored into their original order at the re-

ceiver, consecutive frame erasures are perceived as shorter bursts. To this end, an interleaver can be applied. For an input sequence $\ldots, a_{-2}, a_{-1}, a_0, a_1, a_2, \ldots$ an interleaver can be expressed as a permutation $\pi : \mathcal{Z} \rightarrow \mathcal{Z}$ producing a reordered output sequence $\ldots, b_{-2}, b_{-1}, b_0, b_1, b_2, \ldots$ such that $a_i = b_{\pi(i)}$. Every interleaver has a corresponding deinterleaver that acts on the output of the original interleaver and puts the symbols back into their original order (with a possible time delay δ), that is,

$$\pi^{-1}(\pi(i)) = i + \delta \qquad \forall i. \tag{5}$$

The main advantage of interleaving is that it does not increase bandwidth requirements, but it does have the disadvantage of increasing the delay. There exist different interleavers with different delays, complexities and memory requirements. An interleaver that has been successfully applied to DSR is the optimal delay block interleaver [2, 4, 9]. The block interleaver of degree d operates by re-arranging the transmission order of a $d \times d$ block of input vectors. There are two block interleavers considered optimal in terms of maximizing the spread of bursts for a given degree. They are given by,

$$\pi_1(id + j) = (d - 1 - j)d + i \qquad 0 \leq i, j \leq d - 1, \tag{6}$$

$$\pi_2(id + j) = jd + (d - 1 - i) \qquad 0 \leq i, j \leq d - 1. \tag{7}$$

These two interleavers form an invertible pair, that is, $\pi_1 = \pi_2^{-1}$ and $\pi_2 = \pi_1^{-1}$ and are equivalent to a rotation of the block of feature vectors either 90º clockwise or 90º anti-clockwise (as shown in figure 3). The delay introduced by these interleavers is related to their degree and is equal to $\delta = d(d - 1)$ frames.

Table 2 shows the results obtained by applying optimal delay block interleavers of different degree ($d = 3, 4, 5, 6$). At the receiver, the Aurora standard mitigation is used as EC technique. As can be observed, better results are obtained when the degree of the interleaver, that is, the delay, increases. However, in comparison with table 1, the MMSE estimation based on VQ replicas achieves better results at the only cost of a few overhead bits (that, as we mentioned in section 3, could be introduced without increasing the final bandwidth).

At this point, it can be argued that Aurora standard mitigation is a rather poor EC technique. More advanced techniques have been proposed which exploit statistical information relating to the feature vector stream and provide better results [10]. Maximum a-posteriori (MAP) estimation, for example, replaces the sequence of lost vectors, $\mathbf{X}_m$, by an estimate that maximizes its likelihood conditioned on the received vectors, $\mathbf{X}_o$, and the distribution of the feature vector stream, $P(\mathbf{X}; \mu, \sigma)$. Although it is a coarse approximation, MAP estimation assumes the feature vector stream is Gaussian, so that the MAP estimate reduces to a linear regression entirely described by its mean and variance as [11],

$$\hat{\mathbf{X}}_m = \mu_m + \Sigma_{mo}\Sigma_{oo}^{-1}(\mathbf{X}_o - \mu_o) \tag{8}$$

where μ_m and μ_o are the mean vectors of $\mathbf{X}_m$ and $\mathbf{X}_o$ respectively, Σ_{oo} is the auto-covariance matrix of $\mathbf{X}_o$ and Σ_{mo} is the cross-covariance matrix between $\mathbf{X}_m$ and $\mathbf{X}_o$. Due to the inversion of large covariance matrices (Σ_{oo}^{-1}), which imposes too much of a computational overhead, different variations have been proposed to optimize this estimation [10, 2, 4]. In this work we have chosen the fastest one, consisting on a sliding window applied independently over each feature sequence (further details can be found in [10]).

Table 3 shows the word accuracy obtained with MAP estimation using optimal delay block interleavers. As can be observed,

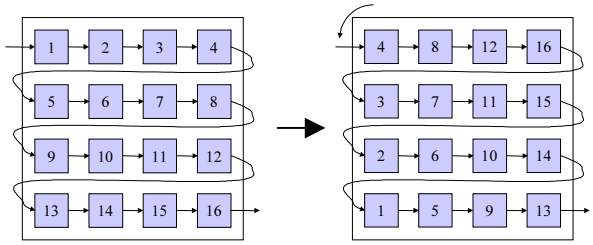


Figure 3: *Illustration of a 4×4 block interleaver. Rotation of 90º anti-clockwise.*

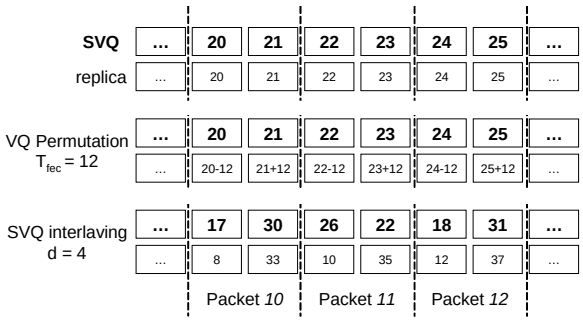


Figure 4: *Example of application of FEC ($T_{fec} = \pm 12$) and interleaving ($d = 4$) in a double stream scheme.*

this combined application of techniques achieves further improvements in comparison with table 2. These results are even somewhat better than those obtained by MMSE estimation with VQ replicas, but with the additional advantage that they do not involve any bandwidth increase.

5. Double-Stream scheme

As independent techniques, there is no reason why FEC codes and interleaving cannot be jointly applied. However, a direct composition of both operations results in a sum of their delays. Thus, if packets are interleaved after FEC codes have been obtained, the delay of the interleaving is added to the delay of the FEC. The same happens when interleaving is applied prior to obtaining the FEC codes.

In this work, we propose an alternative scheme in where this sum of delays is avoided. To this end, feature vectors are grouped in two independent streams. The first stream consists of feature speech vectors coded with SVQ quantization as the standard does, whilst the second stream contains VQ replicas of the first stream. As in section 3, a packet is composed of four vectors: two SVQ vectors from the first stream and two replicas from the second one. Initially, vectors of both streams are ordered by their corresponding time instant. Then, VQ vectors of the second stream are permuted following the proposed FEC scheme, that is, the frames of the current pair of replicas are exchanged with the frame located T_{fec} frames before it and the frame located T_{fec} frames after it. At this point, the resulting packets would be equal to those described in section 3. However, the SVQ vectors of the first stream are now interleaved. In order to do so, the aforementioned optimal delay block interleavers (equations (6) and (7)) can be applied. Figure 4 illustrates the sequence of operations.

At the receiver, the SVQ vectors are restored into their origi-

Condition	*Delay (ms)*				Aurora
R_{loss}, L_{loss}	*60*	*120*	*200*	*300*	
10%, 1	99.00	98.99	99.06	99.03	98.98
20%, 2	98.80	98.79	98.87	98.85	98.08
30%, 4	95.76	97.24	97.72	97.82	90.92
40%, 8	84.65	90.35	93.21	94.58	76.61
50%, 12	72.89	79.76	84.92	87.50	63.27

Table 1: *Word accuracy obtained with MMSE estimation with VQ replicas ($T_{fec} = 6, 12, 20, 30$) in comparison with Aurora.*

Condition	*Delay (ms)*			
R_{loss}, L_{loss}	*60*	*120*	*200*	*300*
10%, 1	99.05	98.98	99.04	99.00
20%, 2	98.79	98.85	98.98	98.95
30%, 4	95.03	96.57	97.75	98.25
40%, 8	83.15	87.07	90.64	93.45
50%, 12	71.06	75.66	80.32	84.46

Table 2: *Word accuracy obtained with block interleaving ($d = 3, 4, 5, 6$) and Aurora standard mitigation.*

Condition	*Delay (ms)*				MAP
R_{loss}, L_{loss}	*60*	*120*	*200*	*300*	estimation
10%, 1	99.02	99.03	99.07	99.01	99.04
20%, 2	98.81	98.86	99.03	98.91	98.31
30%, 4	95.97	97.19	98.03	98.43	93.20
40%, 8	87.08	90.24	92.55	95.04	81.89
50%, 12	76.07	80.65	83.78	88.08	70.19

Table 3: *Word accuracy obtained with MAP estimation with and without block interleaving ($d = 3, 4, 5, 6$).*

Condition	*Delay (ms)*			
R_{loss}, L_{loss}	*60*	*120*	*200*	*300*
10%, 1	99.03	99.03	99.03	99.01
20%, 2	98.92	98.99	99.01	99.00
30%, 4	96.48	98.21	98.72	98.80
40%, 8	86.39	92.19	95.76	97.46
50%, 12	75.77	82.47	88.65	92.21

Table 4: *Word accuracy obtained using a double-stream strategy with block interleaving and MMSE estimation with VQ replicas.*

nal order by means of the corresponding deinterleaver while FEC codes are extracted from packets and used as replicas of lost frames. As in section 3 the MMSE estimation is used to exploit these replicas. Since FEC and interleaving operations are applied over independent streams, this scheme has the advantage of a resulting delay equal to the maximum delay of both operations.

Table 4 shows the results obtained by this scheme for several delays with 4-bit replicas. This double stream-based strategy offers similar or better results than those obtained by interleaving and MAP estimation. Only when the delay is equal to 60 ms, interleaving combined with MAP offers a marginal improvement in the last two conditions (40%, 8 and 50%, 12).

6. Conclusions

In this work we have evaluated the HMM-based MMSE estimation with VQ replicas in comparison with interleaving. As it has been shown, MMSE estimation with VQ replicas performs better than interleaving with a simple mitigation technique, but when interleaving is combined with a statistical-based reconstruction method, the MAP estimation, similar results are obtained, with the advantage of no bandwidth increase.

However, interleaving and VQ replicas could be jointly applied, providing better results than the isolated application of only one of these techniques. Since a direct combination of both operations involves a sum of their delays, we introduce in this work a double stream-based strategy where FEC codes are considered a second virtual stream. Thus, two streams are multiplexed in packets: one with SVQ vectors and the other one with VQ replicas. While VQ replicas are organized following the usual scheme (taking frames at T_{fec} time instants before and after the current frame pair), SVQ vectors are interleaved by means of a block interleaver.

As a result, the proposed strategy achieved the best performance with the advantage of involving a delay equal to the maximum delay of both operations.

7. References

[1] D. Pearce: "Enabling New Speech Driven Services for Mobile Devices: An overview of the ETSI standard activities for Distributed Speech Recognition Front-ends". *AVIOS 2000: The Speech Applications Conference*, San Jose (USA), May 2000.

[2] B. Milner and A. James, "Robust Speech Recognition over Mobile and IP Networks in Burst-Like Packet Loss", *IEEE Trans. Speech and Audio Processing*, January 2006.

[3] A.M. Peinado, A.M. Gómez, V. Sánchez, J.L. Pérez-Córdoba, A.J. Rubio: "Packet Loss Concealment based on VQ Replicas and MMSE Estimation Applied to Distributed Speech Recognition", in *Proc. ICASSP*, Philadelphia, 2005.

[4] B.P. Milner and A.B. James: "Analysis and Compensation of Packet Loss in Distributed Speech Recognition using Interleaving". in *Proc. Eurospeech*, 2003.

[5] D. Pearce, H. Hirsch: "The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions". in *Proc. ICSLP*, vol. 4, pp. 29-32, Beijing, China, October 2000.

[6] "RTP Payload Format for DSR ES 201 108", IETF Audio Video Transport WG, RFC3557, July 2003.

[7] W. Jiang, H. Schulzrinne: "Modeling of Packet Loss and Delay and Their Effect on Real-Time Multimedia Service Quality", in *Proc. NOSSDAV*, June 2000.

[8] A.M. Peinado, V. Sánchez, J.L. Pérez-Córdoba, A. de la Torre: "HMM-Based Channel Error Mitigation and its Application to Distributed Speech Recognition". *Speech Communication*, Vol 41/4, 2003.

[9] A. James, A. Gómez and B. Milner: "A Comparison of Packet Loss Compensation Methods and Interleaving for Speech Recognition in Burst-Like Packet Loss", in *Proc. ICSLP*, 2004.

[10] A. M. Gómez, A. M. Peinado, V. Sánchez, B. P. Milner, A. J. Rubio: "Statistical-based Reconstruction Methods for Speech Recognition in IP Networks", in *Procs. Cost 278 and ITRW workshop*, 2004.

[11] B. R. Ramakrishana: "Reconstruction of Incomplete Spectrograms for Robust Speech Recognition". PhD thesis, Carnegie Mellon University, 2000.

RECOGNITION OF CODED SPEECH TRANSMITTED OVER WIRELESS CHANNELS

Angel M. Gómez*, Antonio M. Peinado, *Senior, IEEE,* Victoria Sánchez, *Member, IEEE,* and Antonio J. Rubio, *Senior, IEEE*

Dept. Teoría de la Señal, Telemática y Comunicaciones
University of Granada, Facultad de Ciencias, Campus de Fuentenueva S/N
Granada, Spain 18071. Phone: +34 958243271. Fax: +34 958243230.
{amgg,amp,victoria,rubio}@ugr.es

Abstract

Network-based speech recognition (NSR) and distributed speech recognition (DSR) have been proposed as solutions to translate speech recognition technologies to mobile environments. NSR is the most straightforward solution since it does not require any modification in the mobile phone, however DSR offers higher robustness against codec compression and transmission channel degradation. This paper explores an alternative approach for remote speech recognition which combines the advantages of NSR and DSR. In this scheme, a standard speech codec is used for speech transmission but the recognition is performed from the received codec parameters. In particular, we focus on the effect of transmission channel errors, which can cause a more severe performance reduction on speech recognition than codec distortion. First, we show that an NSR solution can approach DSR through a reconstruction technique along with an adapted noise reduction technique originally proposed for acoustic noise. Then, these results are improved by working with recognition features directly extracted from the codec bitstream by means of parameter transcoding. Required modifications on current networks in order to access the bitstream are described. The network upgrading with the tandem free operation (TFO) protocol is an attractive solution. This upgrade not only offers an overall improvement on the end-to-end speech quality, but would also allow a recognition performance similar, and even higher in poor channel conditions, to

This paper has been supported by the Spanish MEC, project TEC 2004-03829/TCM and FEDER funds.

that obtained by DSR when parameter transcoding along with the proposed mitigation techniques are applied.

Index Terms

Speech recognition, remote speech recognition, cellular radio, speech codecs, transmission errors, decoding, decoded speech signal, error compensation, transcoding.

RECOGNITION OF CODED SPEECH TRANSMITTED OVER WIRELESS CHANNELS

I. INTRODUCTION

During the last years, there has been an arising interest in translating speech recognition technologies to mobile environments in order to enable access to remote voice-activated services. Such speech recognition systems can adopt several architectures. In the embedded approach, the recognition is carried out in the mobile device itself [1]. This approach has the advantage that very little information (extracted from the recognized message) is transmitted. However, its functionality can be quite restricted due to hardware constraints and power consumption.

A more powerful approach is that where the recognition is performed by a remote server. In addition to avoiding the constraints of the aforementioned embedded architecture, this approach introduces new advantages. The mobile device is now a thin client with the only function of carrying out some type of encoding of the input speech. Therefore it is discharged of any maintenance or upgrading regarding speech recognition or any query generation to the service offered by the remote server. This feature also facilitates the operation in different languages.

There are several possible architectures for a remote speech recognition system. The most straightforward is based on the transmission of the speech signal after source encoding (usually CELP-based) and channel encoding. At the server side, the recognition is performed on the decoded speech. This approach has been referred to as Network-based Speech Recognition (NSR) [2]. Another possibility is to have a client with a local front-end which processes the speech signal in order to directly obtain the parameters (usually mel frequency cepstrum coefficients, MFCC) used by the remote server (back-end) to perform recognition. This approach is known as Distributed Speech Recognition (DSR) and has been promoted by the STQ ETSI Aurora working group, which has issued four ETSI standards [3]–[6]. DSR avoids the speech coding step and the transmission is performed over a data channel. One of the arguments employed in [7] to support the DSR approach against NSR is its robustness against transmission channel errors and codec compression degradation.

There is a third approach for remote speech recognition that avoids the need for new hardware in the

mobile phone (as NSR) and can provide robustness against codec compression and transmission channel degradation (as DSR). In this scheme, the speech codec implemented in the mobile phone is used, and the recognition is directly performed from the received codec parameters. Previous work dealing with this approach has been mainly concerned with the codec degradation [8]–[12]. In this paper we will focus on robustness against transmission channel errors as they can cause a more severe performance reduction on speech recognition [7]. First we will show that, when suitable compensation algorithms are applied to mitigate the effect of channel errors, the NSR solution approaches the DSR results. These results will be later improved through the aforementioned third approach, that is, when working with features extracted from the codec parameters. This approach, referred to in the following as transcoding, can achieve a performance similar or even higher to that obtained by the ETSI standard for DSR [3].

As starting point we will use the results presented in [7]. This paper showed that the Aurora DSR standard [3] provided a much better recognition performance than that obtained with NSR using an Enhanced Full Rate (EFR) coder under a GSM transmission scheme and three different channel conditions. This starting point and the experimental framework used throughout the paper will be described in section 2. In section 3, we will identify and analyze the causes of performance reduction when recognizing from EFR decoded speech over degraded channels and develop solutions to mitigate their effects in section 4. In section 5 we will propose the parameter transcoding from EFR to DSR as a method to provide further improvements, and will justify the use of this transcoder in the current networks. Finally, in section 6 we will summarize our conclusions and future work.

II. Experimental Framework

The experimental setup is based on the framework proposed by the mentioned ETSI STQ-Aurora working group [13]. The Aurora DSR front-end [3] provides a 14-dimension feature vector containing 13 MFCCs (including the 0th order one) plus log-Energy. The recognizer is the one provided by Aurora and uses eleven 16-state continuous HMM word models, (plus silence and pause, that have 3 and 1 states, respectively), with 3 gaussians per state (except silence, with 6 gaussians per state). The training and testing data are extracted from the Aurora-2 database (connected digits). Training is performed with 8440 clean sentences and test is carried out over set *A* (4004 clean sentences distributed into 4 subsets).

Under the EFR scheme, the speech signal is encoded and decoded according to GSM 6.60 standard [14]. Channel coding, decoding, and bad frame mitigation (substitution and muting) are accomplished according to GSM 5.03 [15] and GSM 6.61 [16]. The corresponding operations for the DSR standard can be found in [3]. The Aurora error mitigation algorithm can be summarized as follows: once a burst,

containing $2B$ frames, is detected, the first B frames are replaced by the last correct frame received before the burst and the last B ones by the first one received after the burst.

The channel will be simulated using the GSM error patterns EPx (x=1,2,3), which are applied to the bitstream containing the encoded EFR parameters [13]. These error patterns are in AEG format and represent three channel conditions: EP1 (10 dB C/I, good quality), EP2 (7 dB C/I, medium quality) and EP3 (4 dB C/I, lower quality). For their application to DSR, these patterns are decimated to the appropriate bit rate (4.8 kbps) as in [7]. The performance obtained by EFR and DSR is shown in table I (EFR and DSR columns). As expected, DSR provides a better performance (measured as word accuracy) than that obtained by the recognition of EFR-coded speech.

III. Effect of channel degradation over EFR based speech recognition

A. Bad frame errors and background errors

The EFR decoder applies error detection and correction using the protection bits included within each frame. However, the parameters of the encoded speech and their individual bits are not equally protected, being divided into protected and unprotected bits according to their subjective importance to speech quality. Information about the condition of each frame is given by a bad frame indicator (BFI) which is enabled (BFI=1) only when a frame seriously damaged is received. Since the synthesis of bad frames would be very annoying to the listener, the decoder discards such frames and applies a substitution and muting algorithm [16]. The purpose of frame substitution is to conceal the effect of lost frames, whilst the output muting in the case of several lost frames indicates the breakdown of the channel to the user and avoids generating possible annoying sounds as a result from the frame substitution procedure.

Since the BFI is determined through the protected bits only, it cannot be asserted that those frames not marked (BFI=0) are exactly the sent ones. As a result, we can consider that the whole degradation is caused by two types of errors: *background errors*, in the case of frames with BFI=0 but with bits altered, and *bad frame errors* where frame substitution and muting is applied (BFI=1).

In order to decouple the effect of both errors, we have carried out two experiments:

- *Background errors*: In this situation we only take into account the noise generated by unmarked frames (BFI=0). The frames marked with BFI=1 are replaced by the corresponding correct frame (with the codec parameters corresponding to a clean transmission). Thus, we are exclusively evaluating the effect of the background errors.
- *Bad frame errors*: In this case, we only take into account the noise due to frames with marked BFI. The unmarked frames (BFI=0) are replaced by the corresponding correct frame, assuring us there

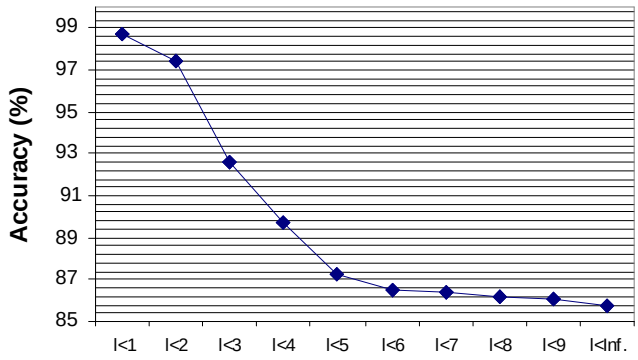


Fig. 1. *Word accuracy versus burst length (l) on EP3 channel condition when recognizing EFR coded speech.*

are not any bits modified. Thus, we are only evaluating the effect of the bad frame errors.

Table I shows the results of these experiments under different channel conditions. It can be observed that the bad frame errors are the main source of performance reduction, while the effect of the background errors is negligible. This suggests that the subjective quality criteria used for bit protection would also be adequate for speech recognition.

Bad frames usually appear grouped, making up bursts. In figure 1 we analyze the effect of the burst length l over the recognition performance (bursts have been extracted under the EP3 channel condition). Each label $l < n$ indicates that only those bursts with length less than n are considered (longer bursts are replaced by the corresponding correct frames). Thus, $l < 1$ coincides with a clean transmission and $l < \infty$ with the EP3 condition. Although longer bursts are more damaging, the effect of lengths longer than five frames is almost negligible since they are very scarce even in the worst condition (EP3).

B. Codec memory noise

Analyzing the speech signal obtained in a noisy transmission, we can observe a degradation of the signal corresponding to correct frames after a burst (bad frame errors). Figure 2 shows the decoded signal obtained under a clean transmission and the same signal during and after an error burst. It is observed that, after a bad frame burst, the decoded speech signal requires some frames to recover and approximate the original signal. This degradation is inherent to the predictive nature of the encoding process and we will refer to it as *Memory noise*.

Our objective now is to separate the effects of the 'substitution and muting' noise during the burst from the memory noise after the burst and study the impact of each type of noise on the reduction of the recognition accuracy. Unlike in the previous subsection, where erroneously received parameters were

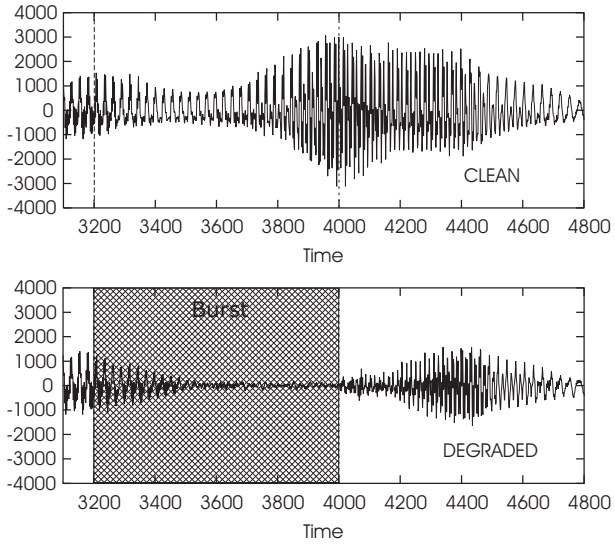


Fig. 2. *Segment of a clean signal and the same signal degraded by memory noise after a burst (from sample 3201 to 4000).*

replaced by clean EFR codec parameters, the new experiments are now carried out by substituting noisy speech samples by those corresponding to a clean transmission. These experiments are:

- *'Substitution and Muting' noise*: speech samples belonging to correctly received frames (BFI=0) are replaced by the corresponding correct samples. Thus, the only noisy samples that remain are those where the substitution and muting algorithm is applied.
- *Memory noise*: speech samples belonging to bad frames (BFI=1) are replaced by their corresponding correct samples, so that the memory noise is the only remaining degradation (background errors have been previously eliminated).

The results of both experiments are also shown in table I. It is observed that the memory noise which appears in correct frames received after each burst is also an important source of degradation. In the following section, we will restrict our study to the 'substitution and muting' and memory noises, neglecting the effect of the background noise.

IV. Improving the recognition from decoded speech

As a first approach, we will show in this section how to improve the performance of a conventional NSR system, where recognition is carried out on the decoded speech [17]. This is the most immediate way to recognize coded speech, obtaining the corresponding feature vector from the decoded speech. Assuming we know the BFI flag of the codec bitstream (this will be commented later in section 5), the first step will be to classify each feature vector used for recognition as correct or erroneous. In order to

TABLE I

Word accuracy under different channel conditions for DSR and EFR with background/bad frame errors and memory/'substitution and muting' noise.

Chan.	DSR	EFR	Backg. Errors	Bad Frame Errors	Memory Noise	Subst. & Mut. Noise
Clean	99.04	98.70	–	–	–	–
EP1	99.04	98.44	98.50	98.61	98.44	98.68
EP2	98.95	96.91	98.31	97.53	97.73	98.28
EP3	93.41	84.48	98.22	85.80	93.54	90.47

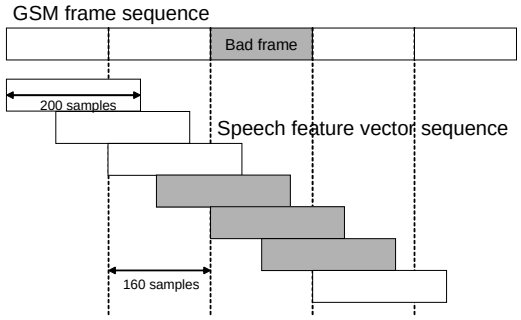


Fig. 3. *Mapping function between EFR frames and recognizer frames.*

achieve this, it must be taken into account the possible differences of size and shift between the frames of the speech codec and the analysis windows employed by the recognizer, as in fact occurs between the windows of the EFR codec and the Aurora feature extractor. In this case, we will apply the following mapping function which marks each feature vector as correct (0) or incorrect (1),

$$F_{map}(n) = \begin{cases} 1 & (BFI(\lfloor \frac{n}{2} \rfloor) = 1) \text{or} (BFI(\lfloor \frac{n+1}{2} \rfloor) = 1) \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where n is the time index of the recognizer feature vector and $BFI(m)$ is the bad frame indicator of the codec frame m. This mapping is depicted in figure 3.

A. Burst reconstruction

As mentioned in the previous section, when bad frames are received at the decoder, these are considered as lost and they are replaced, that is, as if no information about these frames was available. The synthesized

speech signal is the result of a delay-constrained algorithm (substitution and muting) which only tries to improve the subjective speech quality. However, delay is not so critical in speech recognition as in speech transmission. In fact, the very mitigation algorithm of the DSR standard waits until the reception of the first vector after the burst to recover it. Thus, the feature vectors corresponding to the bad (lost) frames (according to the mapping function) can be reconstructed by applying a simple linear interpolation between the last and first correct vectors before and after the error burst:

$$\hat{\mathbf{x}}(t) = \mathbf{x}(t_s) + \frac{\mathbf{x}(t_e) - \mathbf{x}(t_s)}{t_e - t_s}(t - t_s) \quad (t_s < t < t_e) \tag{2}$$

where $\hat{\mathbf{x}}(t)$ is the estimated feature vector at time t, and $\mathbf{x}(t_s)$ and $\mathbf{x}(t_e)$ the last and first correct feature vectors before and after the burst, respectively. Despite of the simplicity of this technique, the recognition performance can be meaningfully improved, as shown in table II (experiment *EFR interpolation*).

B. Memory noise compensation

Even if all distortion due to 'substitution and muting' noise was removed, there would still be left the degradation due to the codec memory which has a very different nature. We are not dealing now with lost information, but with a signal degradation that we have called memory noise. An appropriate way of diminishing its effect over recognition is to apply some type of acoustic noise compensation algorithm suitable to this specific degradation. A good and simple choice in order to compensate this degradation would be the FCDCN (Fixed Codeword-Dependent Cepstral Normalization) technique [18], which can be implemented when stereo training data (the same signal is available in clean and noisy conditions) are available. The basic idea of FCDCN is to apply a correction vector $\mathbf{r}$ to the noisy feature vector $\mathbf{y}$ that depends on the instantaneous SNR of the input signal and on vector $\mathbf{y}$ itself. The dependence of $\mathbf{r}$ on $\mathbf{y}$ is simplified by quantizing $\mathbf{y}$. Thus, the resulting estimation $\hat{\mathbf{x}}$ of the clean feature vector $\mathbf{x}$ is obtained as,

$$\hat{\mathbf{x}} = \mathbf{y} + \mathbf{r}(SNR, q) \tag{3}$$

where q is the quantization index corresponding to $\mathbf{y}$.

The main problem for the application of FCDCN to memory noise compensation is how to obtain the SNR. As shown in figure 2, the degradation level is higher for the first frames after the burst and diminishes as time goes on. For this reason, we will consider that the instantaneous SNR depends on the length l of the previous burst and the distance t from the end of the burst, both measured in number of vectors. Thus, a set of correction vectors $\mathbf{r} = \mathbf{r}(l, t, q)$ will be estimated from the stereo training speech

database. This estimation can be carried out for each burst length l by simulating (on the training data) as many error bursts as needed in order to obtain a good estimation. Thus, if we have obtained M bursts of length l, the correction vector is obtained by averaging as,

$$\mathbf{r}(l,t,q) = \frac{1}{M}\sum_{m=1}^{M}\left(\mathbf{x}_t^{(m)} - \mathbf{y}_t^{(m)}\right) \tag{4}$$

where $\mathbf{x}_t^{(m)}$ and $\mathbf{y}_t^{(m)}$ correspond to the clean and noisy feature vectors at time t after the end of burst m. As can be observed, the estimation of the correction vectors has a high amount of calculations. However, these need to be carried out only once (in the training stage). In order to apply the compensation of equation (3), it is only required the VQ quantization of vector $\mathbf{y}$. Thus, the computational burden involved is rather low.

In order to limit the memory requirements, the maximum burst length is fixed to $l = 5$ since the effect of longer bursts is negligible as shown in figure 1. If $l > 5$, then the correction vectors for $l = 5$ are used. Moreover, it is considered that the effect of a burst does not surpass $t = 20$ (for $t > 20$ the correction tends to zero).

In our implementation, each feature vector is split in seven pairs of features (MFCCs 1 and 2, 3 and 4, ... , 11 and 12, and MFCC 0 and the log-Energy) in the same way as the encoding of feature vectors in the Aurora standard is carried out. In fact, we have used the split vector quantizers (SVQ) provided with the ETSI standard to obtain the quantization index q.

The compensation of memory noise by the proposed Adapted FCDCN (A-FCDCN) technique will be used together with burst reconstruction based on linear interpolation. This last technique must be applied after A-FCDCN in order to use a corrected version of the first feature vector after the burst. The results obtained by this combination are shown in table II (EFR interpolation & A-FCDCN). The main conclusion is that this combination approaches the performance of DSR for EP1 and EP2 and outperforms it for EP3.

C. Compensation of codec noise

The above correction vectors were estimated by comparing decoded speech transmitted under clean and noisy channel conditions (with bursts of bad frames). However, we can obtain further improvements if we estimate the correction vectors by using clean original speech that has not gone through the coding/decoding process. In this case, we are alleviating the distortion introduced by the codec in addition to the memory noise. The compensation of the codec noise can be extended to the frames not affected by memory noise (beginning of the sentence until the first burst and frames with $t > 20$ from the end of

TABLE II

Word accuracies obtained from DSR, EFR, EFR interpolation, EFR interpolation & A-FCDCN, and EFR interpolation & extended A-FCDCN.

Chan.	DSR	EFR	EFR Interp.	EFR Interp. & A-FCDCN	EFR Interp. & extended A-FCDCN
Clean	99.04	98.70	98.70	98.70	98.81
EP1	99.04	98.44	98.43	98.46	98.64
EP2	98.95	96.91	97.55	97.82	98.19
EP3	93.41	84.48	90.76	94.04	94.04

a burst) by applying to those frames a new set of corrections $\mathbf{r}(q)$ obtained from the comparison of clean original and coded/decoded speech. Table II shows that this new combination of techniques (interpolation & extended A-FCDCN) provides further improvements for the less noisy conditions (clean, EP1 and EP2).

V. EFR/DSR TRANSCODING

A. Transcoder justification

The techniques proposed in the previous section require the knowledge of, at least, the BFI flag along with the decoded signal. This information could be provided in a cooperative network where PCM samples belonging to a bad frame were marked in some way. This would imply changes in the PCM transcoder located in the rate adaptor unit (TRAU), but since the TRAU is a functional entity of the base transceiver station (BTS) or, in more advanced networks, the base station controller (BSC) [19], this approach only involves changes in centralized hardware (figure 4a).

However, these changes can be avoided through the tandem free operation protocol (TFO) established in the ETSI TS 128 062 standard [20]. This standard is intended to avoid the traditional double speech encoding/decoding in a mobile to mobile configuration call, improving the speech quality and possibly reducing the transmission delay. In order to avoid the tandem operation, the TRAU units exchange TFO frames carrying compressed speech (codec bitstream) and in-band signaling (BFI flags among others) along the network. Since the procedures used to establish TFO are considered system independent, it would be possible to attach to the network a TFO compatible device which not only accesses the BFI flag, but also directly transcodes the EFR codec parameters into feature vectors usable by the recognition server, obtaining further improvements (figure 4b).

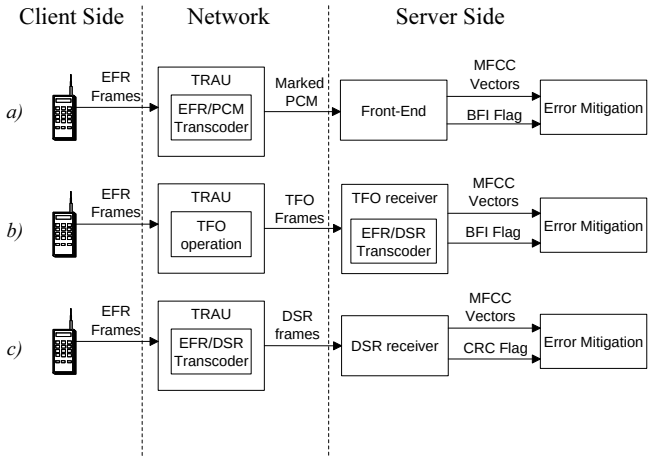


Fig. 4. *Alternative solutions to conventional DSR: a) PCM marking, b) TFO protocol, c) DSR transcoding in the TRAU. In each one modifications on the mobile phone are avoided.*

This transcoder can be an independent device at the back-end through the TFO protocol, but also an integral part of the TRAU in networks without support for TFO. Without changes in the user hardware (the mobile device), encoded speech could be directly transcoded to speech features and sent to a DSR compatible recognizer through the same mechanisms as in a DSR architecture (figure 4c).

Speech recognition from speech codec parameters and transcoding has been previously researched by a number of authors [8]–[12]. However, to our knowledge extent, it has never been developed an EFR to DSR transcoder. In this section, we will propose an EFR/DSR transcoder which converts the EFR parameters into the Aurora recognition parameters (MFCCs and log-Energy), being fully compatible with the DSR-based back-ends. That is, the existing speech recognizer has not to be retrained. Moreover it will include compensation mechanisms for the wireless channel distortion.

B. Transcoder description

The EFR codec operates on speech frames of 160 samples which are divided into four subframes of 40 samples. Each EFR frame contains two sets of LSPs obtained from a 10th order linear prediction analysis carried out twice every frame using two different asymmetric windows. These windows are centered in the second and fourth subframe (40 samples) respectively. On the other hand, the DSR feature extraction algorithm is performed over 200 samples every 80 samples or, on a subframe-term basis, over 5 subframes every 2 subframes. Since LSP sets and DSR frames are produced at the same rate (one every 80 samples, that is, 2 subframes), a naive assignment could be one LSP set for each DSR

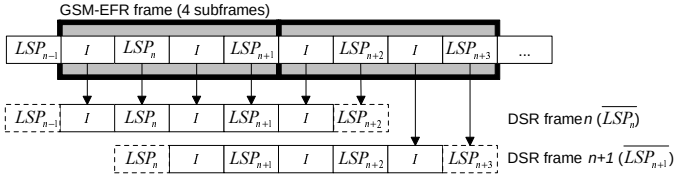


Fig. 5. *Assignment of the LPS sets to each DSR analysis window (LSP sets are interpolated from the adjacent sets in subframes marked as 'I').*

frame. However, this assignment would only accurately represent the second subframe of the analysis window. A more accurate representation, based on an average of the LSP sets involved in the synthesis of each subframe, can be given for the DSR analysis window as:

$$\overline{\mathbf{LSP_n}} = \frac{\mathbf{LSP}_{n-1} + 4\mathbf{LSP}_n + 4\mathbf{LSP}_{n+1} + \mathbf{LSP}_{n+2}}{10} \tag{5}$$

where $\overline{\mathbf{LSP_n}}$ is the LSP set assigned to the n^{th} DSR frame and $\mathbf{LSP_n}$ is the n^{th} LSP set received. Figure 5 depicts this averaging. These LSPs are then converted into a set of LPC coefficients ($a_l; l = 1, \ldots, 10$).

The MFCC coefficients $MFCC(k)$ (k=0,...,12) can be obtained by using the procedure described in the DSR standard but substituting the FFT spectrum by the LPC spectrum,

$$|H(\omega_i)| = \sigma |H'(\omega_i)| \tag{6}$$

where σ is the LPC gain, $\omega_i = 2\pi i/N$ ($i = 0, \ldots, N = 255$), and $|H'(\omega_i)|$ is the gain-normalized LPC spectrum,

$$|H'(\omega_i)| = \frac{1}{\left|1 + \sum_{l=1}^{10} a_l e^{-j\omega_i l}\right|} \qquad (\omega_i = 2\pi i/N) \tag{7}$$

If we apply a triangular filterbank with $M = 23$ filters to $|H(\omega_i)|$, and a DCT transformation to the log-outputs of the filterbank, it is easy to derive that,

$$MFCC(k) = \begin{cases} M \log \sigma + MFCC'(k) & k = 0 \\ MFCC'(k) & k = 1, \ldots, 12 \end{cases} \tag{8}$$

where $MFCC'(k)$ represents the cepstral coefficients corresponding to the normalized LPC spectrum $|H'(\omega_i)|$.

In order to obtain $MFCC(0)$ and the log-Energy, we need to compute the average energy σ^2 of the LPC filter excitation. For each EFR subframe, the excitation signal is obtained as,

$$u(n) = g_p v(n) + g_c c(n) \tag{9}$$

where g_p is the adaptive codebook gain, $v(n)$ is the adaptive codebook excitation signal, g_c is the innovative codebook gain, and $c(n)$ the innovative codebook signal [14]. Therefore, the excitation energy of subframe m is,

$$E(m) = g_p^2 E_d(m) + g_c^2 E_c(m) \tag{10}$$

where $E_d(m)$ is the adaptive signal energy, and $E_c(m)$ is the innovative signal energy. Gains g_p and g_c are directly obtained from the received bitstream, and the innovative signal energy is computed from the innovative signal $c(n)$ ($n = 0, \ldots, 39$) as,

$$E_c(m) = \frac{1}{N_{sf}} \sum_{n=0}^{N_{sf}-1} c^2(n) \tag{11}$$

where $N_{sf} = 40$ is the subframe length.

However, $v(n)$ is not available in the received bitstream, since it is obtained from the excitation $u(n)$ through a long term prediction filter with fractional pitch. In the bitstream this fractional pitch is represented by the nearest integer pitch lag, T, plus a fractional value k. Considering only the integer pitch lag simplifies the computation as long term prediction filtering becomes a simple shift. In our case, since the excitation signal $u(n)$ is not available, but we do have its energy $E(m)$ in each subframe, then the energy $E_d(m)$ of $v(n)$ corresponding to subframe m will be approximated by means of a weighted average of the energies corresponding to the previous subframes, from where the adaptive signal vector would be extracted, as,

$$\begin{aligned} E_d(m) = \quad & E\left(m - \lfloor \frac{T}{N_{sf}-1} \rfloor\right) \frac{N_{prev}}{N_{sf}} + \\ & E\left(m - \lfloor \frac{T}{N_{sf}} \rfloor\right) \frac{(N_{sf} - N_{prev})}{N_{sf}} \end{aligned} \tag{12}$$

where $N_{prev} = T \bmod N_{sf}$. This formula is only valid for pitch delays $T \geq 40$. When $T < 40$ the EFR decoder uses the excitation signal of the current subframe to obtain the adaptive signal vector of this same subframe. Obviously, working on a subframe-term basis, this can not be done. Therefore, in this particular case, the adaptive signal energy corresponding to subframe m can only be approximated by using the excitation energy corresponding to the previous subframe $m-1$. This reveals two problems in our approach:

- The excitation energy inside a subframe can be tilted to the beginning or to the end of it. This fact can cause a bad estimation of $E_d(m)$.
- The minimum pitch delay considered in EFR is $T = 17$ samples. In this case, the adaptive signal energy would be roughly approximated by a whole subframe of 40 samples when only its last 17

samples are used in the calculation of the actual adaptive signal.

In order to relieve these problems, we have increased the time resolution in equations (11) and (12) by considering halves of subframes ($N_{sf} = 20$). Using half subframes, the aforementioned particular case only occurs when $T < 20$. Therefore, in the worst case, we will be approximating the average energy of 17 samples by that of 20 samples which is a much better estimation compared with the previous one.

Finally, the average excitation energy σ^2 of each DSR frame is obtained by adding the excitation energies of the 10 corresponding half subframes. The $MFCC(0)$ coefficient is computed as indicated in equation (8), and the log-Energy as,

$$logE = \log \frac{1}{2\pi} \int_{-\pi}^{\pi} \left| \frac{\sigma}{1 + \sum_{l=1}^{10} a_l e^{-j\omega l}} \right|^2 d\omega \tag{13}$$

The EFR coder quantizes the LSP coefficients using first order moving-average prediction. This means that LPC coefficients of the first EFR frame after a burst are incorrect and, therefore, the MFCC coefficients, from $MFCC(1)$ to $MFCC(12)$, of the first two DSR frames after that burst will also be damaged, as deduced from equation (8). However, the $MFCC(1-12)$ coefficients of the third and following DSR frames (after the burst) are correct. This result is the most important advantage of the transcoding approach in comparison with the recognition from decoded speech, since the effect of the memory noise has been mostly removed. Unfortunately, this is not the case of $MFCC(0)$ and $\log E$. In this case, although there are also some quantization artifacts, as in the LSPs encoding, that introduce errors, the main source of degradation is the recursive computation of the adaptive excitation of equation (12) (using the excitation obtained T samples before). This causes that the whole excitation signal and, therefore, the excitation energy σ^2, is erroneous during a longer period. As a consequence, $MFCC(0)$ and $\log E$ are unavoidably affected by memory noise.

C. Compensation of the remaining degradation and experimental results

The recognition performance obtained by the transcoder approach (experiment T-EFR), under different channel conditions, is shown and compared in table III with DSR, EFR and EFR with all the improvements introduced in the previous section (Improved EFR). The only error mitigation mechanism introduced in T-EFR is the linear interpolation of the recognition features during error bursts. We observe that the transcoder approach is slightly better than the improved EFR (except in clean conditions).

In order to compensate the remaining degradation (memory noise and codec noise), the A-FCDCN compensation technique developed in section 4 can be applied. The correction vectors are obtained in the same way as explained in the mentioned section. In particular, we introduce the following compensations:

TABLE III

Word accuracy obtained with DSR, EFR, Improved EFR, Transcoded-EFR and Improved Transcoded-EFR.

Chan.	DSR	EFR	Impr. EFR	T-EFR	Impr. T-EFR
Clean	99.04	98.70	98.81	98.73	98.88
EP1	99.04	98.44	98.64	98.68	98.82
EP2	98.95	96.91	98.19	98.26	98.57
EP3	93.41	84.48	94.04	94.08	95.57

1) An A-FCDCN compensation of the $MFCC(1-12)$ coefficients of the two first DSR frames after an error burst.
2) An A-FCDCN compensation of $MFCC(0)$ and $logE$ of the 20 first DSR frames after an error burst.
3) An A-FCDCN compensation of the codec noise is applied to the rest of the DSR feature vectors.

The recognition results can be also found in table III (experiment Improved T-EFR) and show that the transcoder approach can outperform the Improved EFR and approximate the performance obtained by DSR under clean, EP1 and EP2 conditions. In the case of the EP3 channel condition, the Improved T-EFR technique outperforms DSR in more than 2% of word accuracy.

VI. Conclusions and future work

In this paper we have described and evaluated alternative solutions to the problem of remote speech recognition in a mobile environment which combine the advantages of the NSR and DSR solutions. These alternative approaches provide a high robustness against transmission channel errors and codec compression (as in DSR) without changes in the user mobile phone (as in NSR).

Working in a GSM environment, we have first analyzed the effect of channel errors on EFR-based speech recognition. Channel errors were identified as the main source of degradation when the recognition is performed from EFR encoded and transmitted speech, being the degradation due to EFR coding almost negligible [2], [7].

Analyzing the EFR decoded speech, we have identified three sources of degradation when an error prone channel is used: background noise, 'substitution and muting' noise and codec memory noise, being the first one negligible and the two last ones the main responsible for the performance degradation in the speech recognizer. Although these types of noise can be treated directly on the decoded waveform, as it is shown in the NSR solution proposed in section 4, better results can be obtained by applying parameter

transcoding from EFR to DSR, where the memory noise can be confined to only two speech features (MFCC0 and logE). The FCDCN technique, that was originally proposed in an acoustic noise context, has resulted, with the appropriate changes, very effective in combating this memory noise (and easily extensible to other CELP type transcoders). As a result, the proposed EFR to DSR transcoder, jointly with the set of proposed mitigation techniques, provides a recognition performance similar or even higher to that obtained by the ETSI standard for DSR.

In order to apply the methods proposed in this paper it is necessary to access the bitstream. For this reason, we have proposed different architectures where this is possible. If the underlying network supports the TFO protocol, no modifications have to be done neither to the user device nor to the network, since the bitstream can be reached through a TFO compatible device. If the network is not TFO compatible, its upgrading can be very interesting to the mobile operators since they can not only provide remote recognition performance comparable to DSR but also an overall improvement on the end-to-end speech quality. If that was not possible, the EFR to DSR transcoder could be implemented in the TRAU as another transcoding function. Both solutions exhibit two important advantages: centralized modifications and scalability. While the user terminal remains unmodified, changes are only locally performed on the network, that is, it is not necessary a global network upgrade but only an upgrading of those ending parts of the network where, due to their localization (public buildings, airports, ...), speech recognition could be a service to offer.

In this paper we have not used any information regarding the reliability of the bits received. Both the DSR and the transcoder-based approach proposed could be further improved if this type of information combined with MMSE estimation techniques were used [21], [22]. On the other hand, since the techniques proposed along this paper do not require reliability measures, they could be straightforwardly applied to IP networks (lossy packets channels), by considering bad frames as lost frames.

This work can be straightforwardly extended to the Adaptive Multi-Rate (AMR) codec [23], where the source and channel encoding is modified in order to face the channel conditions. This codec is composed of several source codecs with a similar structure to that of the EFR codec (the EFR codec is in fact one of the codecs of AMR), so a similar type of transcoder could be developed for AMR. Although less affected by transmission channel errors, we will still have similar errors as the ones described in this paper. In AMR we would additionally have to consider the degradation introduced by the speech encoding process on speech recognition performance [24] which is negligible in the case of EFR. Further work will address this problem.

Another issue is the treatment of the acoustic noise that has given rise to the development of the

Advanced Front-End standard (AFE) [4]. Apart from the fact that there is recent work [25] that shows that back-end solutions against acoustic noise can outperform the ETSI Advanced Front-End standard, an interesting research subject would be the translation of the acoustic noise reduction methods from the AFE to a transcoding approach.

We have proposed in this paper robust alternatives to the DSR solution for a remote speech recognition system. The presented solutions do not exclude the DSR solution but they could coexist in a mobile network, with the advantage of being able to choose the most suitable solution in each situation.

REFERENCES

[1] S. Deligne, S. Dharanipragada, R. Gopinath, B. Maison, P. Olsen, H. Printz: "A robust high accuracy speech recognition system for mobile applications". *IEEE Transactions on Speech and Audio Processing*, pp. 551-561, Nov. 2002.

[2] T. Fingscheidt, S. Aalburg, S. Stan, C. Beaugeant: "Network-Based vs. Distributed Speech Recognition in Adaptive Multi-rate Wireless Systems". *Proc. of ICSLP*, pp. 2209-12, Sept. 2002.

[3] ETSI ES 201 108 v1.1.3. Distributed Speech Recognition; Front-end Feature Extraction Algorithm; Compression Algorithms. September 2003.

[4] ETSI ES 202 050 v1.1.3. Distributed Speech Recognition; Advanced Front-end Feature Extraction Algorithm; Compression Algorithms. November 2003.

[5] ETSI ES 202 211 v1.1.1. Distributed Speech Recognition; Extended Front-end Feature Extraction Algorithm; Compression Algorithms. November 2003.

[6] ETSI ES 202 212 v1.1.1. Distributed Speech Recognition; Extended Advanced Front-end Feature Extraction Algorithm; Compression Algorithms. November 2003.

[7] D. Pearce: "Enabling New Speech Driven Services for Mobile Devices: An overview of the ETSI standards activities for Distributed Speech Recognition Front-ends". *AVIOS 2000: The Speech Applications Conference*, San Jose (USA), 2000.

[8] J. M. Huerta and R. M. Stern: "Speech recognition from GSM codec parameters". *Proc. of ICSLP*, vol. 4, pp. 1463-1466, 1998.

[9] S.H. Choi, H.K. Kim, H. S. Lee and R.M. Gray: "Speech recognition method using quantised LSP parameters in CELP-type coders". *Electronics Letters*, vol. 34, no. 2, pp. 156-157, Jan. 1998.

[10] A. Gallardo-Antolín, C. Peláez-Moreno and F. Díaz-de-María: "A robust front-end for ASR over IP and GSM networks: an integrated scenario". *Proc. of Eurospeech*, Sept. 2001.

[11] H.K. Kim, R.V. Cox: "A Bitstream-Based Front-End for Wireless Recognition on IS-136 Communications System". *IEEE Trans. on Speech and Audio Processing*, vol. 9, no. 5, pp. 558-568, July 2001.

[12] B. Raj, J. Migdal and R. Singh: "Distributed Speech Recognition with Codec Parameters". *Proc. of IEEE Workshop on Automatic Speech Recognition and Understanding*, pp. 127-130, Dec. 2001.

[13] D. Pearce, H. Hirsch: "The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions". *Proceedings of ICSLP*, vol. 4, pp. 29-32, Beijing, China, October 2000.

[14] ETSI EN 300 726. Digital Cellular Telecommunications System; Enhanced Full Rate (EFR) Speech Transcoding (GSM 06.60), 1999.

[15] ETSI EN 300 909. Digital Cellular Telecommunications System; Channel Coding (GSM 05.03), 1999.

[16] ETSI EN 300 727. Digital Cellular Telecommunications System; Substitution and Muting of Lost Frames for Enhanced Full Rate Speech Traffic Channels (GSM 06.61).

[17] A.M. Gómez, A.M. Peinado, V. Sánchez, J.L. Pérez-Córdoba, A.J. Rubio: "Mitigation of Channel Errors in EFR-Based Speech Recognition". *Proc. of ICASSP*, May 2004.

[18] A. Acero: "Acoustical and Environmental Robustness in Automatic Speech Recognition", Ph.D. Thesis, Dept. of Electrical and Computer Engineering, Carnegie Mellon Univ., 1990.

[19] ETSI TS 148 060 v5.1.0. Digital Cellular Telecommunications System; In-band control of remote transcoders and rate adaptors for full rate traffic channels. June 2002.

[20] ETSI TS 128 062 v5.4.0. Digital Cellular Telecommunications System; Inband Tandem Free Operation (TFO) of speech codecs; Service description. September 2003.

[21] A.M. Peinado, V. Sánchez, J.L. Pérez-Córdoba, A. de la Torre: "HMM-Based Channel Error Mitigation and its Application to Distributed Speech Recognition". *Speech Communication* Vol 41/4, 2003.

[22] A.M. Peinado, V. Sánchez, J.L. Pérez-Córdoba, A. Rubio: "Efficient MMSE-Based Channel Error Mitigation Techniques. Application to Distributed Specch Recognition Over Wirelsss Channels". *IEEE Trans. On Wireless Communications*, In Press.

[23] ETSI EN 301 704 v7.2.1. Digital Cellular Telecommunications System; Adaptive Multi-Rate (AMR) speech transcoding (GSM 06.90). 1998.

[24] H. Kelleher, D. Pearce, D. Ealey, L. Mauuary: "Speech Recognition Performance Comparison Between DSR and AMR Transcoded Speech". *Proc. of ICASSP*, June 2000.

[25] A. Bernard, Y. Gong, X. Cui: "Can Back-ends be more Robust than Front-ends? Investigation over the AURORA-2 database". *Proc. of ICASSP*, May 2004.

Combining media-specific FEC and error concealment for robust distributed speech recognition over loss-prone packet channels

Angel M. Gómez*, Antonio M. Peinado, *Senior Member, IEEE,*
Victoria Sánchez, *Member, IEEE,* and Antonio J. Rubio, *Senior Member, IEEE*
Dept. Teoría de la Señal, Telemática y Comunicaciones
University of Granada, Facultad de Ciencias, Campus de Fuentenueva S/N
Granada, Spain 18071. Phone: +34 958243271. Fax: +34 958243230.
{amgg,amp,victoria,rubio}@ugr.es

Abstract

This paper presents a mixed recovery scheme for robust distributed speech recognition (DSR) implemented over a packet channel which suffers packet losses. The scheme combines media-specific forward error correction (FEC) and error concealment (EC). Media-specific FEC is applied at the client side, where FEC bits representing strongly quantized versions of the speech vectors are introduced. At the server side, the information provided by those FEC bits is used by the EC algorithm to improve the recognition performance. We investigate the adaptation of two different EC techniques, namely minimum mean square error (MMSE) estimation, which operates at the decoding stage, and weighted Viterbi recognition (WVR), where EC is applied at the recognition stage, in order to be used along with FEC. The experimental results show that a significant increase in recognition accuracy can be obtained with very little bandwidth increase, which may be null in practice, and a limited increase in latency, which in any case is not so critical for an application such as DSR.

EDICS Category: 5-HIDE

This paper has been supported by the Spanish MEC, project TEC 2004-03829/TCM and FEDER funds. Portions of this work were presented at the IEEE International Conference on Acoustics, Speech and Signal Processing, Philadelphia, USA, May 18-23, 2005.

I. Introduction

The last few years have witnessed an increasing use of IP networks. This has led to considerable interest in the integration of voice services, such as speech transmission and speech recognition, into those networks. Nowadays, speech services such as Voice over IP (VoIP) offer an alternative to traditional speech transmission systems; moreover, an important effort is being devoted to offer reliable speech recognition over IP.

A very attractive approach to speech recognition over IP networks is the distributed speech recognition (DSR) solution. This is based on a client-server architecture, as are many other services. On the client side, a front-end analyzes, quantizes and packetizes speech data and sends it through the network. On the server side, a remote server, called the back-end, receives the speech data and performs speech recognition. Thus, only those parameters which are relevant to the recognition process are transmitted, which significantly reduces the data rate. In order to achieve this, a common front-end must be established. Thus, several DSR standards [1]–[4] have been developed by the ETSI STQ Aurora working group. Since they were initially conceived for circuit-switched networks, the IETF Audio Video Transport workgroup, along with Motorola, has developed several recommendations [5], [6] which define the real time protocol (RTP) [7] payload formats for DSR. These formats are based on the payload formats of the ETSI standards.

When transmitting real-time data over a packet switched network, one of the most common problems encountered is that of packet loss. Packet losses are caused by the inability of IP networks to offer a reliable, high-quality packet delivery service. Furthermore, packet losses usually occur in bursts, in which multiple consecutive packets are lost. Although lost packets are not critical in most applications, since the receiving end can request their retransmission, in real-time applications such as speech recognition this would introduce a considerable delay, degrading the naturalness of the human-machine interaction.

Packet losses can have a serious effect on recognition performance, and so recovery techniques are needed. Sender-driven and receiver-based repair are two complementary paradigms embodying such techniques. While receiver-based repair or error concealment (EC) tries to minimize the effect of lost packets without assistance from the sender, sender-driven techniques assume such participation. Novel receiver-based techniques have recently been proposed to mitigate the effect of packet losses on the performance of DSR systems. These techniques exploit the redundant information present in the speech signal to achieve a better recognition performance than the standard reconstruction method based on replacing the lost vectors by copies of the nearest received ones. Reconstruction based on data-source modelling [10] and the maximum a posteriori (MAP) estimation method [11] are examples of this. Other

techniques, such as Weighted Viterbi Recognition (WVR) [12], [13], treat the losses directly inside the recognizer instead of offering estimates of lost vectors. This latter technique seems to offer the best performance in the presence of long bursts [11].

However, receiver-based concealment is usually limited in the sense that it implicitly relies on the assumption that the signal segment to be recovered is in steady state. Nevertheless, sender-driven and receiver-based repair are complementary techniques, and applications should use both methods to achieve the best performance. An example of this can be found in references [11], [14], where interleaving and estimation are jointly applied. However, other sender-based techniques, such as forward error correction (FEC), have not been widely applied to speech recognition over IP networks.

In this paper, we explore the application of FEC codes in order to improve speech recognition robustness against packet losses. We specially focus on media-specific ones where a replica, a strongly quantized version of the current feature vector, is repeated in another packet. Part of the success of these replicas results from their treatment at the back end. For this reason, we also introduce a combined strategy whereby the EC algorithm is specifically designed to exploit the information contained in the codes, even when this information is minimal. As a result, two EC algorithms are adapted to the use of replicas: HMM-based MMSE estimation and Weighted Viterbi Recognition. These algorithms provide a significant improvement in the performance of a DSR system under adverse channel conditions with very few overhead bits and a very limited delay.

The rest of this paper is organized as follows: first, the experimental framework is described; then, the advantages of using media-specific FECs on IP networks are introduced. In section 4 we present the proposed media-specific FEC and comment on the results of directly using replicas with no additional post-processing. Sections 5 and 6 are devoted to the detailed description of the two enhancing techniques proposed and to presenting the experimental results. In section 7, we discuss the payload format for the ETSI DSR standard and indicate several possibilities for the introduction of the proposed FEC overhead bits. Finally, the conclusions of this work are summarised in section 8.

II. Experimental Framework

The experimental setup is based on the framework proposed by the ETSI STQ-Aurora working group [8]. The Aurora DSR front-end [1] segments the speech signal into overlapped frames of 25 ms every 10 ms and provides a 14-dimension feature vector containing 13 MFCCs (including the 0th order one) plus log-Energy. These features are grouped into pairs and quantized by means of seven Split Vector

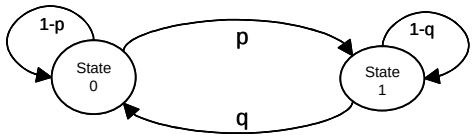


Fig. 1. Gilbert Model. State 0 is the error free state and state 1 causes packet loss.

TABLE I

CHANNEL LOSS TEST CONDITIONS.

Condition	p	q	l_{avr}	l_r
1	0.1111	1.0000	1	10%
2	0.1250	0.5000	2	20%
3	0.1071	0.2500	4	30%
4	0.0833	0.1250	8	40%
5	0.0625	0.0625	16	50%

Quantizers (SVQ). All codebooks have a 64-centre size (6 bits), except the one for MFCC-0 and log-Energy, which has 256 centres (8 bits).

The recognizer is the one provided by Aurora [8] and uses eleven 16-state continuous HMM word models, (plus silence and pause, which have 3 and 1 states, respectively), with 3 gaussians per state (except silence, with 6 gaussians per state). The training and testing data are extracted from the Aurora-2 database (connected digits). Training is performed with 8440 clean sentences and testing is carried out over set *A* (4004 clean sentences distributed into 4 subsets).

IP packets are generated according to the recommendations of the RTP payload format for DSR [5], where at least two frames (one frame pair) per packet are transmitted in order to avoid too high a network overhead due to headers. Although more frame pairs per packet could be transmitted, this is not recommended since longer consecutive frame losses occur when a packet is lost. For this reason, in this work only one frame pair per packet is sent.

The channel burstiness exhibited by IP communications is modelled by a 2-state Markov model [9], also known as a Gilbert-Elliot model. Figure 1 depicts this model, where *p* is the probability of the next packet being lost, provided the previous one has arrived; and *q* is the probability of the next packet not being lost, given that the previous one was lost. These parameters can be set in accordance with an

average burst length (l_{avr}) and a loss ratio (l_r). The reconstruction methods presented in this paper are tested under the channel conditions listed in Table I. As shown in the next section, long loss bursts (even at a small loss ratio) are the main cause of performance degradation. However, at the same loss rate, fewer and fewer bursts appear when l_{avr} increases, as the fact of longer bursts implies a greater number of losses. In order to prove the adequacy of the proposed techniques under adverse conditions with long bursts, some conditions may show unrealistically high amounts of packet loss. The purpose of this is to provide a significant number of long burst losses.

The frame numbering included in the RTP header allows us to rearrange the packets received and to detect the frame losses. Repetition of the nearest received feature vector [1] is used as the basic mitigation technique for a packet loss burst. This technique can be summarized as follows: once a burst, containing $2B$ frames, is detected, the first B frames are replaced by the last frame received before the burst and the last B ones by the first one received after the burst. The results obtained by this mitigation technique will be taken as our baseline.

III. Advantages of Media-specific FEC on bursty channels

FEC techniques rely on the addition of repair data from which the contents of lost packets can be recovered. This repair data must be kept to a reasonable size in order to avoid running into into the same problem as retransmission, that is, imposing too much overhead to the network and worsening the channel. Two classes of repair data may be added to a stream, namely media-specific and media-independent FECs.

In media-independent FEC schemes, the repair data added to the stream are independent of the contents of that stream. As a consequence, the repair is an exact replacement of a lost packet. Each code takes a codeword of k data packets and generates $n - k$ additional recovery packets, and thus a block of n packets are transmitted over the network. This imposes a minimum number of packets (belonging to the same block) which need to be received in order to reconstruct the information contained in a block. Well known media-independent FEC schemes include parity coding [17] and Reed-Solomon (R-S) codes [18]. Because of their excellent error correction properties, R-S codes have been applied to speech recognition over packet networks [16], [19], in particular due to their resilience against bursty losses.

Media-specific FEC schemes use knowledge of the stream to improve the repair process [20]. The idea underlying these techniques is the replication of each feature vector in multiple packets, so that if a packet is lost, another packet containing the same vector will be able to cover the loss. Obviously, the replicas are coded with a secondary encoding which requires fewer bits. Unlike media-independent FEC, these schemes do not impose a minimum number of packets (belonging to the same block) which

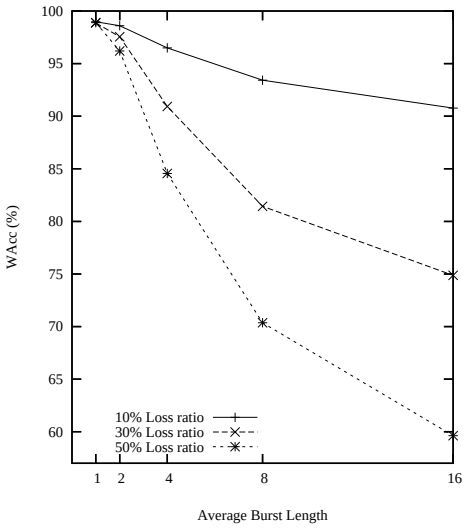


Fig. 2. Effect of the average burst length on speech recognition word accuracy at 10%, 30% and 50% loss ratio.

need to be received in order to reconstruct the information sent in a block. When a packet arrives, the additional information within it can be directly used to recover a lost frame.

Let us now analyze the effect of packet losses on speech recognition performance and how media-specific FEC can be useful to increase robustness against such losses. At the same loss rate, bursty losses are usually more damaging in data transmission than isolated losses. This is specially true in automatic speech recognition, in which the local stationarity of speech has an important effect on the performance of the different mitigation algorithms. Figure 2 depicts the word accuracy for several average burst lengths (1, 2, 4, 8 and 16 packets) at the same loss ratio (10%,30% and 50%). The repetition of the nearest received vector has been used as the mitigation technique (baseline). As can be seen, longer average burst lengths reduce recognition accuracy considerably more than do shorter ones for a given overall percentage of loss. Thus, when each packet contains replicas of other relatively distant packets, these can be used not only to recover some lost frames, but also to break bursts of losses into shorter bursts. Since short bursts are better reconstructed, the recognition performance can be improved. However, media-specific FEC does not obtain a repair which is an exact replacement for a lost packet. Therefore, an important part of the success of this scheme will depend on the EC algorithm which manages these degraded replicas at the decoder. Moreover, some frames would be irremediably lost (when a packet and its replicas are lost) and this too must be managed by the EC algorithm.

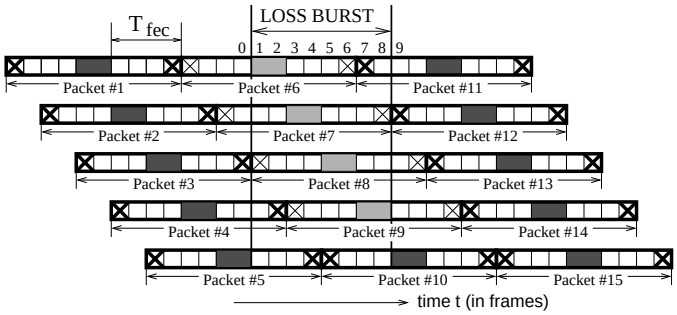


Fig. 3. Proposed FEC scheme. Each frame pair (marked in grey) is sent along with an FEC code containing information about the frames marked with symbol $\times$.

IV. Repairing loss packets through VQ replicas

Let us suppose that, along with the feature vectors corresponding to the current frame pair, we also include in the packet replicas of the feature vectors corresponding to the frames located T_{fec} time units before and after the current frame pair. Each packet would then be composed of four frames. This scheme, depicted in figure 3, is the same as the one proposed in our previous work [21]. In the figure, the replicas in each packet are marked with the symbol $\times$. Frames in white (not marked) are not included in the packet. Time t is measured in frames, while the numbering assigned to packets indicates the order in which the packets are sent. This figure also shows a burst of losses from $t = 1$ to $t = 2B(B = 4)$. The lost frames are indicated in light grey and their associated replicas marked with a weak $\times$.

Obviously, sending all this redundancy increases the bandwidth requirements and, therefore, the loss rate. The end goal of the FEC technique is to achieve recovery from losses while maintaining the final bit-rate within reasonable limits. In order to do so, media-specific FEC uses a secondary encoding which only requires a few bits. In our case, each replica, containing the 14 features (13 MFCCs plus log-Energy) is vector quantized (VQ) using a codebook with 2^N codewords (N bits). Under this scheme, each packet should include the following information:

1) 88 bits corresponding to the SVQ-quantized features of the current frame pair.
2) $2 \times N$ bits corresponding to the VQ-replicas.

The VQ codebook is obtained by applying the k-means algorithm over the 8440 utterances of the training database and using the following weighted distance measure [22]:

$$d_W(\mathbf{x}_r, \mathbf{x}_s) = \frac{\sum_{k=1}^{12} \left(c_r(k) - c_s(k)\right)^2}{\bar{\sigma}_c^2} + \frac{\left(c_r(0) - c_s(0)\right)^2}{\sigma_{c_0}^2} + \frac{\left(\log E_r - \log E_s\right)^2}{\sigma_{\log E}^2} \quad (1)$$

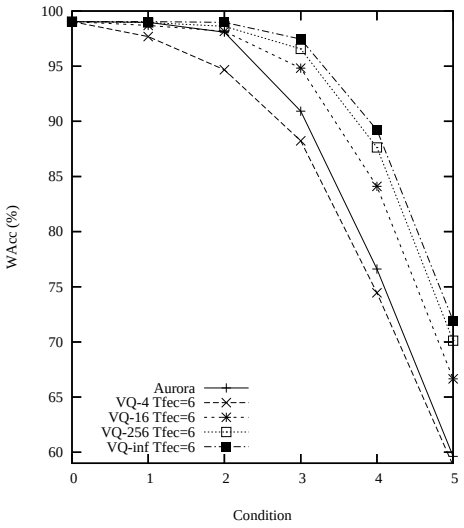


Fig. 4. Effect of direct reconstruction with VQ replicas with $T_{fec} = 6$ and different codebook sizes in comparison with AURORA.

where $\mathbf{x} = (c(0), \ldots, c(12), \log E)$ represents the 14-dimension feature vector, $\bar{\sigma}_c^2$ is the average of MFCCs(1-12) variances, and $\sigma_{c_0}^2$ and $\sigma_{\log E}^2$ are the variances of c_0 and $\log E$, respectively.

As can be seen in the example shown in Figure 3, while some frames (frames 2, 4, 5 and 7, marked with bold $\times$) are recovered, other frames (frames 1, 3, 6 and 8; marked with weak $\times$) are definitively lost when using this scheme. For these frames, we use the following basic mitigation algorithm: for each lost frame at time $t \leq B$ of a loss burst of length $2B$, the last feature vector received (original or replica) is repeated forwards. Analogously, for the second half of the burst ($t > B$) the first received vector is repeated backwards.

The results obtained by means of this strategy are depicted in Figure 4. Different VQ codebooks with 4, 16 and 256 centroids (experiments VQ-4, VQ-16 and VQ-256, respectively) with $T_{fec} = 6$ were tested. We observe that the proposed technique provides better results than the Aurora mitigation algorithm for $N > 4$ bits. However, when coarsely quantized replicas are used (VQ-4), this scheme achieves worse results than does AURORA. Even for VQ-16, a slight decrease in performance is found for channel condition 1. The results obtained with $T_{fec} = 10$ are shown in Figure 5. In this case, although VQ-4 replicas are not useful until conditions 4 and 5 (with long bursts), an overall improvement in performance is observed. Increasing T_{fec} to 10 frames results in an increase in the delay. Human-machine interaction becomes uncomfortable if the total delay is more than 500 ms [16]. However, with $T_{fec} = 10$, the delay is only 100 ms. Finally, only as a reference, the highest expected results, that is, when no quantization

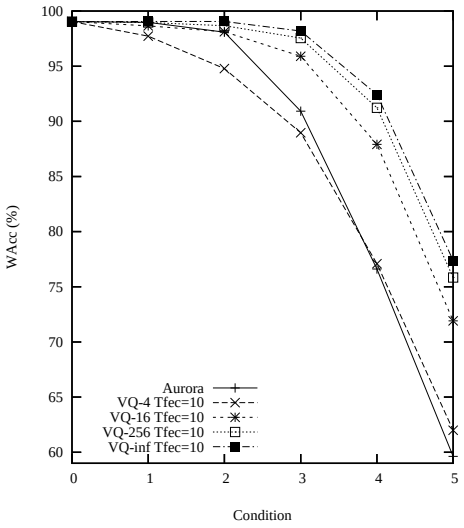


Fig. 5. Effect of direct reconstruction with VQ replicas with $T_{fec} = 10$ and different codebook sizes in comparison with AURORA.

is applied ($VQ - inf, T_{fec} = 6, 10$), are also shown in these figures.

Although the replicas can be directly used at the back end to improve recognition performance, when they are strongly quantized the contrary effect, that is, a reduction in performance is obtained, as shown above. Part of the success of media-specific FECs will depend on the mitigation algorithm which manages the degraded replicas at the decoder. This issue is discussed in the following sections.

V. MMSE ESTIMATION WITH VQ REPLICAS

In previous papers [23], [24], we showed that minimum mean square error (MMSE) estimation is a powerful technique to mitigate the errors introduced by a wireless channel. In [23], [24] it is assumed that all vectors are received at the back end, although affected by channel distortion. In wireless channels, transmission errors can be observed at the receiver as bit errors. On the contrary, packets are completely lost in IP networks. Therefore, the problem in applying MMSE techniques to the case of a lossy packet channel is that no information is received from the channel during a packet loss burst. However, this scenario changes with the introduction of VQ replicas, since they provide information about some lost frames. Now, the degradation of the feature vectors *received* during a bursty packet loss is of a completely different nature. In wireless channels it is due to bit errors, but when VQ replicas are used, distortion is caused by the strong quantization process applied. This approximation is also extensible to those feature vectors that are definitively lost. These can also be considered as received, but extremely distorted by a

degenerated quantization with 0 bits.

A. MMSE formulation for bursty distortions

MMSE estimation is performed on a feature pair basis, since this is the encoding unit used in the Aurora standard. After SVQ quantization, each feature pair is represented by a subvector $\mathbf{c}$ ($\mathbf{c} \in \{\mathbf{c}^{(i)}; i = 0, \ldots, 2^M - 1\}$) ($M$=6,8 in this paper). An MMSE estimation $\tilde{\mathbf{c}}$ of a transmitted feature pair $\mathbf{c}$, received as $\hat{\mathbf{c}}$, can be obtained as [25],

$$\tilde{\mathbf{c}} = E[\mathbf{c}|\hat{\mathbf{c}}] = \sum_{i=0}^{2^M-1} \mathbf{c}^{(i)} P(\mathbf{c}^{(i)}|\hat{\mathbf{c}}) \tag{2}$$

By means of the Bayes rule, the *a posteriori* probabilities $P(\mathbf{c}^{(i)}|\hat{\mathbf{c}})$ can be derived as,

$$P(\mathbf{c}^{(i)}|\hat{\mathbf{c}}) = \frac{b_i(\hat{\mathbf{c}})P_i}{\sum_{j=0}^{2^M-1} b_j(\hat{\mathbf{c}})P_j} \tag{3}$$

where P_i is the *a priori* probability of the codeword (or SVQ centroid) $\mathbf{c}^{(i)}$, and $b_i(\hat{\mathbf{c}}) = P(\hat{\mathbf{c}}|\mathbf{c}^{(i)})$ is the observation probability of the received but distorted vector $\hat{\mathbf{c}}$ given that $\mathbf{c}^{(i)}$ was transmitted.

The *a priori* knowledge about the speech source (P_i) can be enhanced by modelling each feature pair generation along time t by an ergodic continuous Hidden Markov Model (HMM), where each state s_i ($i = 0, \ldots, 2^M - 1$) represents an SVQ centroid $\mathbf{c}^{(i)}$. The HMM model is described by means of the observation probabilities $b_i(\hat{\mathbf{c}})$ and the transition probabilities

$$a_{ij} \equiv P(s_t = \mathbf{c}^{(j)}|s_{t-1} = \mathbf{c}^{(i)}) \tag{4}$$

For the sake of simplicity in the notation, we will express $s_t = \mathbf{c}^{(j)}$ as $\mathbf{c}_t^{(j)}$.

Let us now consider a distortion with a bursty characteristic affecting $T-1$ frames, where $(\hat{\mathbf{c}}_0, \hat{\mathbf{c}}_1, \ldots, \hat{\mathbf{c}}_T)$ is the vector sequence at the receiver and where $\hat{\mathbf{c}}_0$ and $\hat{\mathbf{c}}_T$ are the last and first correctly received vectors before and after the burst, respectively. Thus, we can now estimate the feature vector at time t as

$$\tilde{\mathbf{c}}_t = E[\mathbf{c}_t|\hat{\mathbf{c}}_0, \hat{\mathbf{c}}_1, \ldots, \hat{\mathbf{c}}_T] = \sum_{i=0}^{2^M-1} \mathbf{c}^{(i)} \gamma_t(i) \quad (0 < t < T) \tag{5}$$

where $\gamma_t(i)$, the conditional probabilities, can be computed as

$$\gamma_t(i) = P(\mathbf{c}_t^{(i)}|\hat{\mathbf{c}}_0, \hat{\mathbf{c}}_1, \ldots, \hat{\mathbf{c}}_T) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=0}^{2^M-1} \alpha_t(j)\beta_t(j)} \tag{6}$$

where $\alpha_t(i)$ and $\beta_t(i)$ are the forward and backward conditional probabilities, respectively, defined as,

$$\alpha_t(i) = P(\mathbf{c}_t^{(i)}|\hat{\mathbf{c}}_0, \hat{\mathbf{c}}_1, \ldots, \hat{\mathbf{c}}_t) \tag{7}$$

$$\beta_t(i) = P(\hat{\mathbf{c}}_{t+1}, \ldots, \hat{\mathbf{c}}_T | \mathbf{c}_t^{(i)}) \tag{8}$$

These conditional probabilities can be computed from the transition probabilities a_{ij} and observation probabilities $b_i(\hat{\mathbf{c}}_t)$ of the HMM model through the following forward and backward recursions

$$\alpha_t(i) = \left[\sum_{j=0}^{2^M-1} \alpha_{t-1}(j) a_{ji} \right] b_i(\hat{\mathbf{c}}_t)/K_t \quad (t > 0) \tag{9}$$

$$\beta_t(i) = \sum_{j=0}^{2^M-1} a_{ij} b_j(\hat{\mathbf{c}}_{t+1}) \beta_{t+1}(j) \quad (t < T) \tag{10}$$

where K_t is a normalization factor applied at each step in the recursion (further details can be found in [23]).The following initial conditions are applied to $(t = 0)$ and $(t = T)$,

$$\alpha_0(i) = P_i b_i(\hat{\mathbf{c}}_0)/K_0 \qquad \beta_T(i) = 1 \qquad (i = 0, 1, \ldots, 2^M - 1) \tag{11}$$

where P_i is the *a priori* probability of $\mathbf{c}^{(i)}$.

In our previous work [23] we named this technique as forward-backward MMSE (FBMMSE) estimation in order to remark the use of past and future frames by introducing the forward and backward probabilities and a decoding delay. Further details of this technique, along with an analysis of the complexity of the HMM-based MMSE estimation, can be found in [24].

B. Obtaining the HMM model probabilities for lossy channels

In order to apply the FBMMSE estimation for lossy channels, the transition probabilities $a_{ij} = P(\mathbf{c}_t^{(j)} | \mathbf{c}_{t-1}^{(i)})$ of the HMM model are determined from the training database as in [23]. Regarding the observation probabilities $b_i(\hat{\mathbf{c}}_t) = P(\hat{\mathbf{c}}_t | \mathbf{c}^{(i)})$, we will consider that all feature pairs during a burst $(\hat{\mathbf{c}}_1, \ldots, \hat{\mathbf{c}}_{T-1})$ have been received. The observation probabilities will be determined depending on the type of feature vector considered (received, replica or definitively lost):

- Assuming that the vector at time $t = 0$ or $t = T$ has been correctly received, the corresponding observation probabilities must be set as

$$b_i(\hat{\mathbf{c}}_0), b_i(\hat{\mathbf{c}}_T) = \begin{cases} 0 & \mathbf{c}^{(i)} \neq \hat{\mathbf{c}}_0, \mathbf{c}^{(i)} \neq \hat{\mathbf{c}}_T \\ 1 & \mathbf{c}^{(i)} = \hat{\mathbf{c}}_0, \mathbf{c}^{(i)} = \hat{\mathbf{c}}_T \end{cases} \tag{12}$$

  When a burst starts at the beginning or ends at the end of an utterance, then $\hat{\mathbf{c}}_0$ or $\hat{\mathbf{c}}_T$, respectively, will not be correctly received. They will be lost or a VQ replica and we will be in one of the two cases considered below.
- In the case where a VQ replica is available at time t $(0 \leq t \leq T)$, it is divided into feature pairs that are again SVQ quantized, obtaining $\mathbf{c}_t^{(j)}$, as Figure 6 illustrates. Note that SVQ quantization does

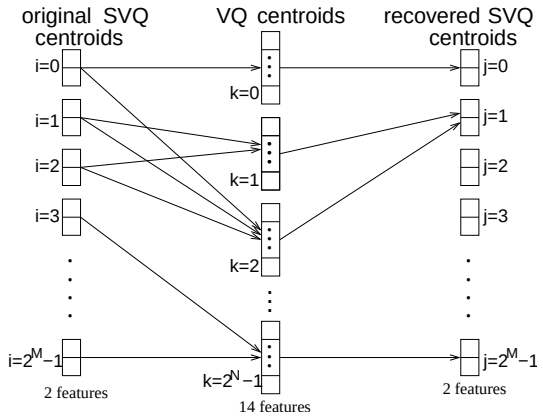


Fig. 6. Example of the sequence of quantizations applied to the replicas corresponding to one of the SVQ feature pairs.

not involve any reduction in recognition performance [23], [26]. It is observed that an original SVQ centroid (representing only one feature pair) can correspond to several VQ centroids (representing a complete vector) depending on the other features different from those of the feature pair being considered. At the same time, each VQ centroid corresponds to one recovered SVQ centroid, although the contrary may not be true (especially when a large VQ codebook is used). In other words, a recovered SVQ centroid can correspond to several VQ centroids, which can also correspond to several original SVQ centroids. Therefore, given an original SVQ centroid $\mathbf{c}^{(i)}$ we can observe several recovered SVQ centroids $\mathbf{c}^{(j)}$ after the double quantization process. It is then possible to determine the observation probabilities from the training database as frequencies of appearance, as follows:

$$b_i(\hat{\mathbf{c}}_t = \mathbf{c}^{(j)}) = P(\mathbf{c}^{(j)}|\mathbf{c}^{(i)}) = \frac{\text{No. recovered symbol } j \text{ given original } i}{\text{No. original symbol } i} \tag{13}$$

If $\mathbf{c}^{(i)}$ represents an empty cell (no training vector is quantized in that cell), an insufficient training threshold ϵ is assigned to $b_i(\mathbf{c}^{(j)})(\forall j = 0, \ldots, 2^M - 1)$.

This scheme implicitly involves the use of a discrete HMM in the FBMMSE estimation. It would also be possible to model these observation probabilities by probability density functions (the HMM model would be continuous and the second SVQ process would be unnecessary), by which a suitable parametric form would be selected for the corresponding pdf's. However, for the sake of simplicity, the discrete version has been selected.

- Finally, when neither an SVQ vector nor a VQ replica is available at time t ($0 \leq t \leq T$), a

TABLE II

WORD ACCURACY OBTAINED BY VQ AND FBMMSE (FOR $T_{fec} = 10$) IN COMPARISON WITH AURORA (AUR), DATA-SOURCE MODELLING (DS) AND MAP ESTIMATION (MAP) FOR DIFFERENT CHANNEL CONDITIONS (CHAN).

Chan	FBMMSE + FEC (Codebook Size)					AUR	DS	MAP
	Mit	4	16	64	256			
1	VQ	97.72	98.66	98.83	98.96	98.98	99.01	99.00
	FB	99.01	99.00	99.01	99.01			
2	VQ	94.78	98.12	98.44	98.67	98.08	98.24	98.30
	FB	98.45	98.73	98.85	98.82			
3	VQ	88.96	95.90	96.95	97.55	90.92	93.17	92.99
	FB	94.18	96.83	97.37	97.68			
4	VQ	77.09	87.91	90.08	91.23	76.61	81.29	81.13
	FB	83.22	88.83	90.30	91.09			
5	VQ	61.99	71.91	74.15	75.84	59.63	63.36	63.09
	FB	66.45	72.12	73.82	75.02			

degenerated VQ quantization with 0 bits is assumed. Thus, all the original SVQ centroids correspond to only one VQ centroid, the overall mean feature vector, and the observation probabilities are assigned as

$$b_i(\hat{\mathbf{c}}_t) = \begin{cases} 1 & \text{if } \mathbf{c}^{(i)} \text{ is not an empty cell} \\ \epsilon & \text{if } \mathbf{c}^{(i)} \text{ is an empty cell} \end{cases} \tag{14}$$

where ϵ is a very small real number. In this case the forward-backward algorithm mainly progresses guided by the transition probabilities as if the observation probabilities were not used.

Table II shows the word accuracies obtained with the proposed mitigation techniques for several codebook sizes and $T_{fec} = 10$. Since the ETSI-DSR standard mitigation is a rather low baseline, results obtained using more advanced mitigation methods, such as data-source modelling [10] and MAP estimation [11], are also shown in Table II for comparison. The best results correspond, as expected, to the FBMMSE (FB) technique, which obtains excellent results even with codebook sizes as low as 4 and 16 (2 and 4 bits, respectively).

It can be argued that comparing FEC and non-FEC schemes is fundamentally unfair, since in FEC schemes the bandwidth is always greater. However, it must be noted that FEC codes as small as 2 or 4

bits can be introduced without any bandwidth increase in the DSR payload format for IP channels, as is discussed in Section VII. Larger FEC codes would indeed require a bandwidth increase, but this will be still a very limited one, while a significant improvement in performance is achieved.

The differences between direct VQ application (VQ) and FBMMSE tend to diminish as the codebook size is increased. This fact is more noticeable for very bad channel conditions, under which VQ may even outperform FBMMSE; this is the consequence of having very long gaps (without replicas) in the middle of the bursts. In these cases, FBMMSE is only guided by the transition probabilities a_{ij}, while the raw VQ technique becomes the Aurora mitigation algorithm. Under these conditions, vector repetition is better than FBMMSE with no channel information.

VI. Weighted Viterbi Recognition with VQ replicas

The previous technique shows how, with very short FEC codes, recognition robustness can be improved with the help of a model of the speech source. This model allows us to obtain better estimations for the lost vectors than the naive replacement by VQ replicas. An alternative approach to palliate the effect of lost frames consists of treating those losses at the recognition stage. When losses are treated at the recognition stage, the speech source model present within the recognizer can be exploited. Weighted Viterbi Recognition (WVR) [27] is based on a modification of the Viterbi Algorithm (VA) whereby the confidence in the decoded feature can be taken into account. The inclusion in the recognizer of reliability values which represent our confidence in a decoded feature is particularly attractive since it would then be possible to inform the recognizer that we have some information about lost frames (the VQ replicas), but that our confidence in them is lower, as they have been degraded by a strong quantization process.

The basic idea of WVR is to incorporate a time-varying reliability γ_t in the VA, obtaining the following state metrics update equation [28]:

$$\phi_t(j) = \max_i [\phi_{t-1}(i) a_{ij}][b_j(\mathbf{x}_t)]^{\gamma_t} \tag{15}$$

where $\phi_t(j)$ is the maximum likelihood of observing the feature vector $\mathbf{x}_t$ in state j at time instant t. Note that, unlike in the previous section, a_{ij} and $b_j(\mathbf{x})$ correspond to the transition and the observation probabilities of the recognition model, respectively. Such weighting has the advantage of becoming a simple multiplication in the logarithmic domain often used for scaling purposes. When a feature vector is reliable, γ_t is set to 1 and equation (15) becomes the original state metrics of the VA update equation. On the other hand, when a feature vector is unreliable, γ_t is set to 0. In such a case, the output probability $[b_j(\mathbf{x}_t)]^{\gamma_t}$ becomes 1 for every state and the feature vector has no influence on the selection of the best path.

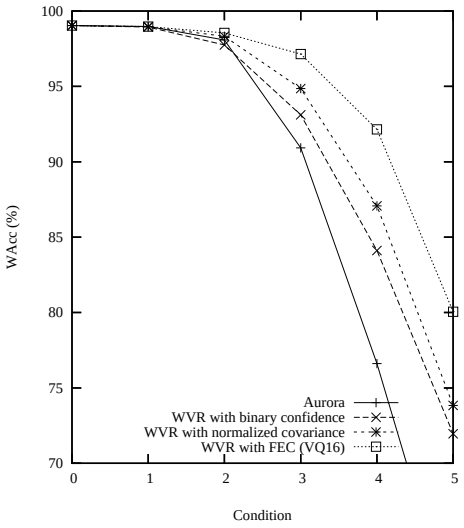


Fig. 7. Comparison of different mitigation procedures: Aurora, WVR and WVR extended with FECs ($VQ = 16, T_{fec} = 10$).

A. WVR applied to lossy packet channels

In lossy packet channels, feature vectors can be considered as reliable or unreliable depending on whether they have been received or lost. Thus, state metrics are continuously updated by virtue of the state transition probability matrix and, unlike the simple concatenation of the vectors received, the timing information of the observation sequence is conserved. However, when bursts are short, reconstruction based on the nearest vector received achieves better results, due to the high short-term correlation of the speech signal. Figure 7 shows the results obtained by the standard mitigation technique (AURORA) and by WVR with binary confidence, under the conditions described in Section 2. As can be seen, WVR with binary confidence gives better results than AURORA mitigation, except for condition 2, when bursts are short (and the loss ratio is high enough for differences to be apparent), where AURORA performs better.

Some authors [12], [13] have reported that this scheme can be refined by using time-varying continuous reliability ($\gamma_t \in [0, 1]$) along with a reconstruction technique for lost vectors. In these papers, the Aurora mitigation technique is applied and a reliability value is independently assigned to each repeated feature. In order to do so, the hypothesis of a diagonal covariance matrix is assumed, and the overall weighted probability can be computed as

$$b_j(\mathbf{x}_t) = \sum_{m=1}^{M} C_{j,m} \prod_{k=1}^{K} \mathcal{N}\left(x(k); \mu_{j,m}(k), \sigma^2_{j,m}(k)\right)^{\gamma_{k,t}} \tag{16}$$

where M is the number of mixture components, C_m is the mixture weight and $\mathcal{N}\left(x(k); \mu_{j,m}(k), \sigma^2_{j,m}(k)\right)$

represents a univariate Gaussian distribution function for the k^{th} feature with mean $\mu_{j,m}(k)$ and variance $\sigma_{j,m}(k)$.

An important hurdle to be overcome in WVR with continuous reliability is, indeed, how to determine the reliability function. It is clear that it is beneficial to decrease the weighting factor $\gamma_{k,t}$ as the feature is consecutively repeated, since the speech signal may have evolved to another sound and the repetition of the received features is no longer valid. However, the problem is how to measure this variation. An empirically good estimate of the reliability function is based on the normalized auto-correlation of each feature, $\rho_k(\tau)$, and is defined as [12]

$$\gamma_{k,t} = \begin{cases} \sqrt{\rho_k(t-t_s)}, & t_s < t \leq (t_e - t_s)/2 \\ \sqrt{\rho_k(t_e - t)}, & (t_e - t_s)/2 < t < t_e \end{cases} \tag{17}$$

where t_s and t_e are, respectively, the time instants of the last and first vectors received before and after the burst, and

$$\rho_k(\tau) = R_k(\tau)/R_k(0) \tag{18}$$

$$R_k(\tau) = E[x_t(k)x_{t+\tau}(k)] \tag{19}$$

However, the normalized auto-correlation function $\rho_k(\tau)$ cannot give coherent reliability values when VQ replicas are used. This is why in the present paper the normalized auto-correlation function has been generalized to the normalized cross-covariance between the original lost feature, x, and its estimate, $\tilde{x}$, defined as

$$\bar{C}[x, \tilde{x}] = C_{x\tilde{x}}/\sigma_x^2 \tag{20}$$

where σ_x^2 is the feature variance and $C_{x\tilde{x}}$ is the cross-covariance between x and $\tilde{x}$, defined as

$$C_{x\tilde{x}} = E[(x - \mu)(\tilde{x} - \tilde{\mu})] \tag{21}$$

where $\mu = E[x]$ and $\tilde{\mu} = E[\tilde{x}]$.

When a reconstruction method is defined and temporal independence is assumed, the normalized cross-covariance can be pre-estimated using the training database. As an example, if the estimation method for a feature, $\tilde{x}_t(k)$, is the repetition of the feature at another time instant, that is, $x_{t+\tau}(k)$, it can be derived that the normalized cross-covariance is a function of the delay τ, obtained as

$$\bar{C}[x_t(k), \tilde{x}_t(k)] = \bar{C}[x_t(k), x_{t+\tau}(k)] = \frac{R_k(\tau) - \mu_k^2}{R_k(0) - \mu_k^2} \tag{22}$$

When the overall variations of a feature are small in comparison with its mean value, the autocorrelation function normally gives high reliability values (this is particularly true for the Log-energy coefficient).

This dependency on the mean of the feature can be avoided by the normalized covariance, as shown in equation (22). Figure 7 also shows the performance obtained by WVR with normalized covariance. It outperforms AURORA in every condition and significantly improves the results obtained by WVR with binary confidence.

B. Reliability of VQ replicas

Once some lost vectors are recovered through VQ replicas from FEC codes, vectors that are definitively lost are replaced by the nearest vector received (either an SVQ vector or a VQ replica). With this recovery scheme, the reconstructed burst is composed of VQ replicas and repetitions of VQ replicas and SVQ vectors. The reliabilities for all those features can be obtained as particular cases of equation (20) as follows:

- When SVQ features are repeated, their reliabilities are obtained as in equation (22),

$$\bar{C}[x_t(k), \tilde{x}_t(k)] = \bar{C}[x_t(k), x_{t+\tau}(k)] = \bar{C}_{SVQ}(\tau; k) \tag{23}$$

- Otherwise, the recovered feature $\tilde{x}_t(k)$ is a repetition of the VQ replica at $t+\tau$ time instant or the replica itself ($\tau = 0$), that is,

$$\tilde{x}_t(k) = VQ[x_{t+\tau}(k)] \tag{24}$$

  therefore,

$$\bar{C}[x_t(k), \tilde{x}_t(k)] = \bar{C}[x_t(k), VQ[x_{t+\tau}(k)]] = \bar{C}_{VQ}(\tau; k) \tag{25}$$

  where $VQ[]$ represents vector quantization.

Thus, finally, the estimate of the reliability function can be defined as,

$$\gamma_{k,t} = \begin{cases} \sqrt{\bar{C}_{SVQ}(\tau_1; k)} & \text{when } \tilde{x}_t(k) = x_{t+\tau_1}(k) \\ \sqrt{\bar{C}_{VQ}(\tau_2; k)} & \text{when } \tilde{x}_t(k) = VQ[x_{t+\tau_2}(k)] \end{cases} \tag{26}$$

Since the cross-covariance decays relatively quickly, only the normalized cross-covariances for a few τ values must be precalculated and stored, while the remaining ones can be assumed to be 0.

As a result of this scheme, a reliability evolution similar to that of the example depicted in Figure 8 would be obtained. As can be seen, reliability decreases as the last received vector is further away. Prior to the appearance of a VQ replica, repetitions of it are used and reliability recovers, reaching a maximum at the replica itself. Depending on the quantization, this peak will be more or less close to one.

Finally, it is possible that replacing a lost vector by a copy of the nearest SVQ would be better than using a copy of a VQ replica or even the VQ replica itself. This is particularly true when replicas are

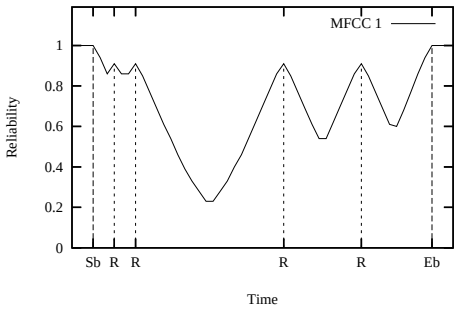


Fig. 8. Example of the reliability evolution of MFCC(1) with time when VQ replicas (R) are used within a burst (starting at S_b and ending at E_b).

strongly quantized and are used close to the beginning or the end of a burst. At these points, given the high short-term correlation of the speech signal, the repetition of the features received could be more reliable than their replacement by a replica. Since an estimate of the reliability for each one is available, the most reliable recovery method can be selected for each lost feature. Thus:

- If the substitution of a lost feature with the nearest SVQ feature received gives a higher reliability value than does the repetition of a VQ replica or substitution with the replica itself, SVQ repetition will be applied.
- Otherwise, the repetition of the nearest VQ replica (or the replica itself, if available) will be used.

As a result of this recovery scheme, the reliability is maximized:

$$\gamma_{k,t} = max\left\{\sqrt{\bar{C}_{SVQ}(\tau_1;k)}, \sqrt{\bar{C}_{VQ}(\tau_2;k)}\right\} \tag{27}$$

Table III shows the word accuracy obtained by this technique for different codebook sizes and channel conditions in comparison with the WVR scheme with normalized covariance but without using replicas. As can be seen, even with the very poor information offered by a replica encoded with only 2 bits (4 centroids), significant improvement is achieved (around 2.5% of WAcc for condition 5). Better results can be obtained with only a 4 bit (16 centroids) codebook, where word accuracy can be improved from 73.84% to 80.06% in condition 5. Figure 7 also shows the performance obtained by this codebook in comparison with the other techniques described above. Finally, as is to be expected, better results are obtained by increasing the codebook size.

VII. Payload Format and Implementation

It has been shown that even with very few FEC bits, the proposed techniques are very effective in mitigating the effect of packet losses in a DSR architecture. In this section, we propose several alternatives

TABLE III

WORD ACCURACY OBTAINED BY WVR WITH FEC (FOR $T_{fec} = 10$) IN COMPARISON WITH WVR WITH NORMALIZED COVARIANCE, FOR DIFFERENT CHANNEL CONDITIONS (CHAN).

Chan	WVR + FEC (Codebook Size)				WVR
	4	16	64	256	
1	98.97	98.97	99.01	99.00	98.97
2	98.45	98.55	98.60	98.69	98.33
3	96.03	97.15	97.58	97.78	94.86
4	89.15	92.15	93.33	93.79	87.07
5	76.36	80.06	82.00	82.83	73.84

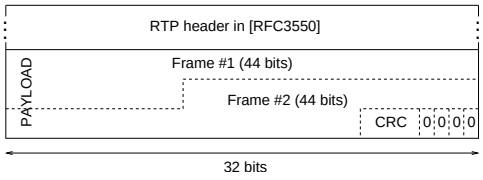


Fig. 9. Payload format for the DSR standard with one frame pair per packet.

for the introduction of the required FEC bits into the DSR payload. According to the recommendations of reference [5], regarding the payload format for the DSR standard, packets are aligned into words of 8 bits as shown in Figure 9. This alignment is required by the IP routers in order to facilitate an easy access to headers. As a result, a padding of 4 bits is required at the end of the DSR payload. Initially, these bits are free. This suggests several ways by which the FEC information required to apply the proposed mitigation methods can be included:

- These four free bits can be used to introduce two VQ replicas quantized with a 4-centre codebook (2 bits/replica). In this case, the replicas are coarsely quantized and, as we can see in Tables II and III, it would be necessary to apply the enhanced error mitigation techniques introduced in this work in order to improve the Aurora results.
- Assuming that the 16 bit checksum of the User Datagram Protocol (UDP) [29], along with the codes usually applied at the physical layer and the coherency test performed at the back end, is sufficient to ensure data integrity, the system performance would be meaningfully improved if we could reuse the 4 bits devoted to the CRC code of the payload to introduce more FEC bits. In this case the

replicas would be quantized with a 16-centre codebook (4 bits/replica) and better results could be obtained.

- Finally, any other increase in the FEC code size would require us to include a new 8-bit word in the packet. Using the bits devoted to padding and CRC, 2 VQ replicas of 8 bits (256 centres) could be applied. In this case, a meaningful increase in performance could be achieved.

VIII. Conclusion

In this work we explore the application of FEC codes in order to improve speech recognition robustness against packet losses. As a sender-driven repair technique, the participation of the sender is required. Although receiver-based concealment improves robustness without assistance from the sender, sender-driven and receiver-based repair are complementary techniques and applications should use both methods to achieve the best performance.

We have focused on media-specific FEC codes. The most important factor which degrades recognition accuracy seems to be burst length. In fact, if only very short bursts appear, a 50% loss ratio could be reached with no reduction in recognition accuracy. For this reason, we have developed a media-specific FEC scheme seeking to split the bursts into shorter ones. In this scheme, feature vectors corresponding to previous and subsequent frames (called *replicas*) are vector quantized and sent in the current packet.

As a result of this scheme, recognition performance can be significantly improved. However, when replicas are strongly quantized in order to minimize the additional bit-rate required, the redundant information sent in each packet is highly degraded. Under these conditions, FEC codes may even prejudice the recognition process if no additional post-processing is applied. Part of the success of the proposed FEC strategy depends on the mitigation algorithm which deals with the degraded replicas at the decoder. In this paper we have introduced and modified two techniques to be used along with VQ replicas, namely HMM-based MMSE estimation and Weighted Viterbi Recognition, that succeed in exploiting the information contained in the codes even when this information is minimal.

In HMM-based MMSE estimation with VQ replicas, a speech source model is used to enhance the information contained in the replicas. In this work we have extended the formulation previously proposed for wireless channels [24] in order to include any type of bursty distortion. While in wireless channels distortion is due to bit errors, when VQ replicas are used, distortion is caused by the strong quantization process. As a result, this technique improves the estimates for lost vectors.

In the case of weighted Viterbi recognition, a reliability function based on the normalized cross covariance is proposed as a confidence estimate. This reliability function can offer a confidence estimate

for a recovered lost feature given the reconstruction method, including those based on VQ replicas. Thus, the recognizer can be informed that we have some information about lost frames, the VQ replicas, but that our confidence in them is lower, as they have been degraded by a strong quantization process.

Both methods can significantly improve recognition accuracy, even when coarsely quantized replicas are used. For example, it is possible to use the four free bits available at the end of the DSR payload (padding bits) to transmit two VQ replicas of two bits. At this point, no additional bandwidth is required. In the worst channel condition, recognition performance can be improved from 59.63% to 66.45% or 76.36% using HMM-based MMSE estimation or Weighted Viterbi Recognition, both with VQ replicas, respectively. More impressive results would be obtained if the CRC bits along with the padding bits could be reused to transmit two VQ replicas of four bits. Without any additional bandwidth, recognition performance can reach 72.12% or 80.06% word accuracy using HMM-based MMSE estimation or WVR, both with VQ replicas, respectively. As has been shown, WVR obtains better results than does HMM-based MMSE estimation, but WVR is limited to speech recognition tasks while HMM-based MMSE estimation could be directly extended to speech transmission. Also, in WVR it is assumed that the speech recognition engine is based on a VA algorithm.

REFERENCES

[1] ETSI ES 201 108 v1.1.3. Distributed speech recognition; Front-end Feature Extraction Algorithm; Compression Algorithms. Sept. 2003.

[2] ETSI ES 202 050 v1.1.3. Distributed speech recognition; Advanced front-end feature extraction algorithm; Compression algorithms. Nov. 2003

[3] ETSI ES 202 211 V1.1.1 Distributed speech recognition; Extended front-end feature extraction algorithm; Compression algorithms. Nov. 2003

[4] ETSI ES 202 212 V1.1.1 Distributed speech recognition; Extended advanced front-end feature extraction algorithm; Compression algorithms. Nov. 2003

[5] ”RTP Payload Format for DSR ES 201 108”, IETF Audio Video Transport WG, RFC3557, July 2003.

[6] ”RTP Payload Formats for European Telecommunications Standards Institute (ETSI) European Standard ES 202 050, ES 202 211, and ES 202 212 Distributed Speech Recognition Encoding”, IETF Audio Video Transport WG, RFC4060, May 2005.

[7] ”RTP: A Transport Protocol for Real-Time Applications”, IETF Audio Video Transport WG, RFC3550, 2003.

[8] D. Pearce, H. Hirsch: ”The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions”. *Proceedings of ICSLP-2000*, vol. 4, pp. 29-32, Beijing, China, Oct. 2000.

[9] W. Jiang, H. Schulzrinne: ”Modeling of Packet Loss and Delay and Their Effect on Real-Time Multimedia Service Quality”, *Proc. NOSSDAV 2000*, June 2000.

[10] A.M. Gómez, A.M. Peinado, V. Sánchez, A.J. Rubio: ”A Source Model Mitigation Technique for Distributed Speech Recognition over Lossy Packet Channels”, *Proc. Eurospeech*, 2003.

[11] A. James, A. Gómez and B. Milner: ”A Comparison of Packet Loss Compensation Methods and Interleaving for Speech Recognition in Burst-Like Packet Loss”, *Proc. ICSLP*, 2004.

[12] A. Bernard and A. Alwaan: ”Low-bitrate distributed speech recognition for packet-based and wireless communication” *IEEE Transactions on Speech and Audio Processing*, Vol. 10, Issue 8, Nov. 2002.

[13] A. Cardenal-Lopez, L. Docio-Fernandez, C. Garcia-Mateo: ”Soft decoding strategies for distributed speech recognition over IP networks” *Proc. ICASSP*, pp. 49-52 vol. 1, 2004.

[14] A.B. James and B.P. Milner: ”An analysis of interleavers for robust speech recognition in burst-like packet loss”. *Proc. ICASSP*, 2004.

[15] Postel, J.: ”Transmission Control Protocol”, RFC 793, USC/Information Sciences Institute, January 1980.

[16] C. Boulis, M. Ostendorf, E.A. Riskin, S. Otterson: ”Graceful Degradation of Speech Recognition Performance Over Packet-Erasure Networks”, in *IEEE Transactions on Speech and Audio Processing*, Vol. 10, No. 8, November 2002.

[17] N. Shacham, P. McKenney: ”Packet recovery in high-speed networks using coding and buffer management”, *Proc. INFOCOM ’90*, vol. 1, June 1990, pp. 124-31.

[18] I.S. Reed and G. Solomon: ”Polynomial codes over certain finite fields”, *J. SIAM*, vol. 8, no. 2, June 1960, pp. 300-4.

[19] E.A. Riskin, C. Boulis, S. Otterson, M. Ostendorf: ”Graceful Degradation of Speech Recognition Performance Over Lossy Packet Networks”, *Proceedings of Eurospeech-2001*, September 2001.

[20] C. Perkins, O. Hodson, V. Hardman: ”A Survey of Packet Loss Recovery Techniques for Streaming Audio”, *IEEE Network Magazine*, Sep. 1998.

[21] A.M. Peinado, A.M. Gómez, V. Sánchez, J.L. Pérez-Córdoba, A.J. Rubio: ”Packet Loss Concealment based on VQ Replicas and MMSE Estimation Applied to Distributed Speech Recognition”, *Proceedings of ICASSP’05*, Philadelphia, 2005.

[22] A.M. Peinado, P. Ramesh, and D.B. Roe: ”On the Use of Energy Information for Speech Recognition using HMMS”’. In *Proceedings of EUSIPCO-90*, vol.2, pp. 1243-6, Barcelona, Sept. 1990.

[23] A.M. Peinado, V. Sánchez, J.L. Pérez-Córdoba, A. de la Torre: ”HMM-Based Channel Error Mitigation and its Application to Distributed Speech Recognition”. *Speech Communication* Vol 41/4, 2003.

[24] A.M. Peinado, V. Sánchez, J.L. Pérez-Córdoba, A. Rubio: ”Efficient MMSE-Based Channel Error Mitigation Techniques. Application to Distributed Speech Recognition Over Wireless Channels ”. *IEEE Trans. On Wireless Communications*, Vol. 4, no. 1, January 2005.

[25] V. Vaishampayan, N. Farvardin: ”Joint design of block source codes and modulation signal sets”. *IEEE Trans. Inf. Theory* Vol 38, July 1992.

[26] D. Pearce: ”Enabling New Speech Driven Services for Mobile Devices: An overview of the ETSI standard activities for Distributed Speech Recognition Front-ends”. *AVIOS 2000: The Speech Aplications Conference*, San Jose (USA), May 2000.

[27] A. Potamianos, V. Weerackody: ”Soft-feature decoding for speech recognition for wireless channels”, in *Proc. ICASSP*, May 2001.

[28] N.B. Yoma, F.R. McInnes and M.A. Jack, ”Weighted Viterbi algorithm and state duration modeling for speech recognition in noise”, in *Proc. ICASSP*, vol. 2, May 1998.

[29] Postel, J.: ”User Datagram Protocol”, USC/Information Sciences Institute, RFC768, 1980.

Conclusions

In this dissertation we have addressed the problem of transmission channel robustness during remote speech recognition, particularly when this is performed over mobile GSM networks and packet switching networks based on IP protocol. In both cases, speech has to pass over an error-prone channel which causes distortions and reduces speech recognition accuracy. In this chapter we summarize our conclusions and findings based on our simulations and experiments. We review the major contributions of this work and present several suggestions for future work.

- We have described the current state of remote speech recognition, pointing out the available solutions and their corresponding advantages and limitations. Although DSR is the most promising architecture, since it has been specifically designed for remote speech recognition, its application to GSM requires hardware changes in the current mobile phones and modifications in the network. On the other hand, the IP networks do not exhibit this problem due to their inherent flexibility to transmit diverse media. Thus, two solutions become feasible nowadays: NSR working over GSM and DSR over IP networks.

- Both solutions can be jointly analyzed assuming an underlying lossy channel, wherein complete pieces of information are either loss by network congestion (in IP) or seriously damaged and discarded (in GSM). Depending on the applied architecture, NSR or DSR, and thus on the speech encoding, predictive or not, besides the frames losses, an additional degradation, the memory noise, can appear.

- Long term memories included within CELP type speech codecs (as EFR and AMR, in GSM) are the source of an additional noise following the frame losses. This noise has an important negative effect on the recognition accuracy and is inherent to the speech synthesis process, where samples of a previous frame are required to generate the current one. If the previous frame is not available, the synthesis of the next and following frames will be erroneous, although these frames had been received correctly.

- This noise can be modeled by additive factors in the cepstrum domain and logarithmic spectrum of the signal (in a similar way to acoustic noise). Due to this, the memory noise can be directly treated on the feature vectors obtained from the decoded speech signal. Since we can generate as much stereo information as we need,

we proposed an adaptation of the FCDCN technique (A-FCDCN) to conceal this noise. Moreover, this technique can slightly alleviate the degradation caused by the speech coding process.

- Alternatively, memory noise can be mostly avoided by means of the transcoding of encoded speech. In this dissertation we propose two new DSR-compatible transcoders, one for EFR, the other for the adaptive set of AMR codecs. Cepstral coefficients from 1 to 12, corresponding to frames after a loss, can be obtained through transcoding without errors, avoiding memory noise and improving the recognition accuracy. However, since MFCC 0 and Log-energy depend on the excitation energy, they are unavoidably affected by memory noise. A-FCDCN technique can be successfully applied to them.

- In addition to being very effective in avoiding the appearance of memory noise, transcoding can be also beneficial against codec distortion caused by AMR with low bitrates.

- Tandem Free Operation (TFO) protocol allows to perform speech transcoding without changing neither the user terminal nor the GSM network. A TFO upgrade would not only offer an improved speech quality since codec tandemming is avoided, but also it would allow a robust remote speech recognition. Without adding any new hardware to the user mobile, word accuracies similar or even better than DSR could be provided. Besides, it is not necessary a global network upgrade but only an upgrading of those ending parts of the network where, due to their localization, speech recognition could be a service to offer.

- On the other hand, the distortion caused by the loss of information (lost feature vectors) can be concealed at the receiver by the use of techniques based on MMSE estimation. Estimates obtained through these techniques can be improved by means of richer models of the speech temporal evolution. However, the memory requirements for these models grow exponentially with its order. By fixing a minimum frequency of appearance, these requirements can be drastically reduced while recognition accuracy is improved. As disadvantage, an auxiliary technique is needed for some reconstructions.

- Alternatively, lost vectors can be concealed by assuming that the whole sequence of feature vectors is the output of a stationary process with gaussian distribution,

being then estimated by maximizing their conditional probability given the observed (received) vectors. The huge computational burden of this estimation based on maximum a posteriori, can be significantly reduced by limiting the observations. Using a sliding window over the same cepstral band, it is possible to perform an efficient MAP estimation which provides meaningful improvements on recognition accuracy.

- MMSE and MAP based techniques provide similar results but have opposite requirements. Computational and memory requirements can be balanced between both techniques by means of the introduction of MAP estimation as auxiliary technique for the proposed MMSE technique with frequency threshold.

- Concealment techniques have the advantage of being applied at the receiver, so they are particularly useful in NSR architectures. However, receiver-based concealment is usually limited since it implicitly relies on the assumption that the signal segment to be recovered is in steady state. In this sense, system performance can be improved if sender-driven techniques are jointly applied.

- Consecutive losses of feature vectors are more damaging to speech recognition than isolated losses. Thus, decreases in average burst length cause significant improvements on recognition. In order to achieve a burst fragmentation, specific FEC codes or replicas can be introduced within the bitstream by the sender.

- However, the introduction of replicas can turn out counterproductive if these are coarsely quantized. In such a case, a post-processing technique which correctly manages the information contained within the replica must be applied. Thus, part of the success of media-specific FECs will depend on the mitigation algorithm applied at the decoder.

- FB-MMSE estimation is shown to be very effective in dealing with degraded replicas at the receiver. In this case, a channel model can be established whereby the "received" vectors, that is, those recovered through replicas, are assumed to be affected by quantization noise. Since a statistical model of speech is applied, FB-MMSE can meaningfully enhance the information contained within the replica, even when it was very distorted.

- By means of the assignment of confidence values which will be later taken into account during speech recognition, it is possible to treat the losses in the recognizer itself. As an advantage, a powerful speech model, the one employed by the recognizer, can be implicitly exploited. Confidence values can be assigned according to the normalized cross-covariance between lost feature and its estimate. This scheme provides coherent confidence values for replicas (and lost vectors), so that these can be post-processed by the recognizer itself. As a consequence, better results are obtained.

- Alternatively, frame interleaving is especially useful to break up bursts of losses and, as an additional advantage, it does not imply a bandwidth increase. However, minimum latency block interleaving applied by other authors does not allow to control the spreading of isolated losses, which turns out to be excessive. On the contrary, convolutional interleavers described by Ramsey allow to control the distance the isolated losses are spread apart. By reducing this distance, it is possible to counteract longer bursts with the same latency, improving the recognition accuracy.

- Finally, the proposed schemes based on replicas and interleaving, as independent techniques, can be combined and provide better results than applying them separately. A double stream scheme, whereby replicas make up an additional virtual stream which is interleaved differently to feature vectors, allows that latencies caused by both techniques do not sum up. Instead, the total latency is the maximum of them. This scheme, combined with WVA, provides the best results shown in this dissertation.

Contributions

The main contributions of this dissertation can be summarized as follows:

- Study of the effect of channel noise on remote speech recognition over mobile telephony [166], particularly when the EFR codec (Enhanced Full Rate) and its traffic channel TCH/EFS (Traffic Channel/Enhanced Full Speech) are used.

- Concealment of channel errors in a remote recognition system based on EFR coded speech [167]. The channel error analysis reveals that the decoded speech signal exhibits tree different types of distortion: background, substitution-and-muting and

memory noise, being the two last ones the main responsible for the reduction on recognition accuracy. Substitution-and-muting noise can be concealed through a simple interpolation technique, while memory noise can be compensated with an adaptation of the FCDCN technique to this kind of noise. This scheme enables a remote speech recognition with accuracies close to DSR, but with EFR decoded speech.

- Development of concealment techniques based on MMSE and MAP estimation for remote speech recognition over lossy channels [109, 169], as well as the comparison and joint application of both types of techniques [170, 171].

- Development and application of transcoders to the recognition of coded speech transmitted over wireless channels. Through EFR [168] and AMR transcoding, memory noise is mostly avoided and recognition performance approaches or even improves that obtained by DSR. The developed transcoders are also DSR-compatible so that retraining or modifying the back-end is not needed. Besides, some proposals are described on how to apply the transcoding to the GSM network, performing only centralized changes (at the TRAU) or even avoiding them (TFO protocol) [168].

- Packet loss concealment based on media-specific FEC and MMSE estimation for robust distributed speech recognition over IP networks [172, 173] and description of some proposals to introduce these replicas in the RTP payload format for DSR without modifying the actual size of packets.

- Development of a scheme that combines VQ replicas with Weighted Viterbi Recognition [173].

- Application of the Ramsey class of interleavers to robust speech recognition over IP networks [159].

- Comparison and joint application of MMSE estimation with VQ replicas and frame interleaving by means of a double stream strategy [174].

Future work

In addition to the speech transmission through an error-prone channel, the likely presence of acoustic noise is another problem that must be addressed in ubiquitous speech recog-

nition. Depending on the system architecture, acoustic noise can worsen the degrading effects of the transmission channel. The NSR architecture could be particularly susceptible to this problem, since it depends on the encoder performance in noisy environments. For this reason, it would be interesting to evaluate the performances of transcoding and A-FCDCN techniques when the speech signal is distorted by acoustic noise. Also, the extension of the A-FCDCN technique to jointly treat channel and acoustic noise would be interesting.

On the other hand, although front-ends that conceal acoustic noise at the sender (AFE and XAFE) can be applied in the DSR architectures, reducing the amount of this, it would be interesting to evaluate the performance of the recovery techniques proposed along this dissertation in noisy environments. On the other hand, a worthwhile research area would be the translation of the noise reduction methods applied in the ETSI advanced front-end to the transcoding approach.

Regarding to the treatment of lost information, frame interleaving has shown up to be very promising. Interleaving has not been completely exploited yet. In this dissertation we have shown that the minimum latency block interleavers previously proposed by other authors do not take advantage of the particularities of recognition-oriented speech transmission. Thus, a future research work is the development of interleavers, for example, based on the Ramsey convolutional interleavers, which exploit these particularities. In this sense, the double stream scheme opens up a big amount of possibilities, such as the optimal combination of interleavers and the application of non permutation-like functions as interleavers for VQ replicas.

In addition, it would be interesting to apply of more advanced coding techniques for the replicas. This would complicate channel modeling in the FB-MMSE technique with replicas and confidence assignment in weighted recognition. However it would increase the information transmitted within the replicas and could improve the recognition.

Finally, the treatment of errors using the speech model within the recognizer, that is, during the speech recognition process itself, is a very promising research area. The development of more detailed heuristics and techniques for the assignment of confidence values, not only for static features but also for the dynamic ones, would be clearly very useful to this end.

Bibliografía

[1] *The W3C Voice Browser Working Group*, 2003. Disponible en linea: http://www.w3.org/Voice/ 1

[2] L.R.Rabiner y B.H.Juang, *Fundamentals of Speech Recognition*. Prentice-Hall, 1993. 2.1.1, 2.1.3, 2.2.1, 2.2.1, 2.2.2

[3] J. T. Tou y R. C. González, *Pattern Recognition Principles*. Addison-Wesley Publishing Company Inc., 1974. 2.1.1

[4] K. F. Lee, *Automatic Speech Recognition (The Development of the Sphinx System)*. Kluwer Academic Publishers, 1989. 2.1.2, 2.1.2, 2.3.3

[5] L. R. Rabiner, J. G. Wilpon, y F. K. Soong, "High performance connected digit recognition using hidden markov models," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, págs. 1214–1225, 1989. 2.1.2

[6] K. F. Lee, "Context-dependent phonetic hidden markov models for speaker-independent continuous speech recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38, págs. 599–623, 1990. 2.1.2, 2.1.2

[7] M.Boros, W.Echert, F.Gallwitz, G.Görz, G.Hanrieder, y H.Niemann, "Towards understanding spontaneous speech: word accuracy vs. concept accuracy," *in proceedings of INTERSPEECH-ICSLP*, págs. 1009–1012, 1996. 2.1.2

[8] W. Minker, "Stochastically-based natural language understanding across tasks and languages," *in proceedings of EuroSpeech*, vol. 3, págs. 1423–1426, 1997. 2.1.2

[9] J.G.Wilpon, L.R.Rabiner, C-H.Lee, y E.R.Goldman, "Automatic recognition of keywords in unconstrained speech using Hidden Markov Models," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38, págs. 1870–1878, 1990. 2.1.2

[10] M.C.Benitez, "Reconocimiento de palabras clave en sistemas independientes de la tarea," Tesis doctoral, Universidad de Granada, 1998. 2.1.2

[11] Y. Gong, "Speech recognition in noisy environments: A survey," *Speech Communication*, vol. 16, núm. 3, págs. 261–291, 1995. 2.1.2

[12] A. Quilis y J. A. Fernández, *Curso de Fonética y Fonología Españolas.* C.S.I.C., 1989. 2.1.2

[13] C. Lee, L. Rabiner, R. Pieraccini, y J. Wilpon, "Acoustic modeling for large vocabulary speech recognition," *Computer Speech and Language*, vol. 4, págs. 127–165, 1990. 2.1.2

[14] R.E.Bellman, *Dynamic Programming.* Princenton Univ. Press, 1957. 2.1.3

[15] T. K. Vintsjuk, "Recognition of words of oral speech by dynamic programming," *Kibernetica*, vol. 4, núm. 1, págs. 81–88, 1968. 2.1.3

[16] H. Sakoe, "Twolevel DP matching - a dynamic programming based pattern matching algorithm for connected word recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 27, págs. 588–595, 1979. 2.1.3

[17] C. S. Myers y L. R. Rabiner, "A level building dynamic time warping algorithm for connected word recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 29, págs. 284–297, 1981. 2.1.3

[18] J.E.Díaz, J.C.Segura, A.J.Rubio, A.M.Peinado, y J.L.Pérez-Cordova, "A new neuron model for an alphanet-semicontinuous HMM," *in proceedings of IEEE ICASSP*, vol. 1, págs. 529–532, 1993. 2.1.3

[19] T.Wessels y C.W.Omlin, "Refining hidden markov models with recurrent neural networks," *in proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks*, vol. 2, págs. 24–27, 2000. 2.1.3

[20] V.Abrash, M.Cohen, H.Franco, y I.Arima, "Incorporating linguistic features in a hybrid HMM/MLP speech recognizer," *in proceedings of IEEE ICASSP*, vol. ii, págs. II/673–II/676, 1994. 2.1.3

[21] N.Morgan y H.Bourlard, "Continuous speech recognition using multilayer perceptrons with hidden markov models," *in proceedings of IEEE ICASSP*, págs. 413–416, 1990. 2.1.3

[22] R.O.Duda y P.E.Hart, *Pattern Classification and Scene Analysus.* J. Wiley and Sons, 1973. 2.2

[23] O.Brigham, *The Fast Fourier Transform.* Prentice-Hall, 1974. 2.2, 2.5

[24] M.R.Schroeder, "Vocoders: Analysis and synthesis of speech," *in proceedings of the IEEE*, págs. 720–773, 1966. 2.2.1, 4.2

[25] A.V.Oppenheim y R.W.Schafer, *Digital signal Processing.* Prentice-Hall Inc., 1975. 2.2.1, 2.2.1

[26] J.R.Deller, J.G.Proakis, y J.H.L.Hansen, *Discrete-time processing of speech signals.* Prentice-Hall, 1993. 2.2.2

[27] O.Ghitza, *Advances in Speech Signal Processing.* Dekker, 1992. 2.2.3

[28] O.Ghitza, "Auditory models and human performance in task related to speech coding and speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 2, págs. 115–132, 1994. 2.2.3

[29] C.R.Jankowski, J. Doan, y R.P.Lippmann, "A comparison of speech processing front ends for automatic word recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 3, págs. 282–293, 1995. 2.2.3

[30] J.C.Junqua y J.P.Haton, *Robustness in automatic speech recognition.* Kluwer Academic Publishers, 1996. 2.2.3

[31] S.Young, "A review of large-vocabulary continuous speech recognition," *IEEE Signal Processing Magazine*, vol. 13, págs. 45–57, 1996. 2.2.4

[32] P. Moreno, "Speech recognition in noisy environments," Tesis doctoral, Carnegie Mellon University, 1996. 2.2.4, 4.3.2

[33] S. Furui, "Speaker-independent isolated word recognition using dynamic featers of speech spectrum," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 34, págs. 52–59, 1986. 2.2.4

[34] P.J.Moreno y R.Stern, "A new algorithm for robust speech recognition: the delta vector taylor series approach," *in proceedings of EuroSpeech*, vol. 5, págs. 2599–2602, 1997. 2.2.4

[35] L. R. Rabiner, S. E. Levinson, y M. M. Sondhi, "On the application of vector quantization and hidden markov models to speaker-independent, isolatedword recognition," *The Bell System technical Journal*, vol. 62(4), págs. 1075–1105, 1983. 2.3

[36] S. Levinson, L. Rabiner, y M. Sondhi, "An introduction to the application of the theory of probabilistic functions of a markov process to automatic speech recognition," *The Bell System technical Journal*, vol. 62(4), págs. 1035–1074, 1983. 2.3

[37] L.R.Rabiner, B. Juang, S.E.Levinson, y M.M.Sondhi, "Some properties of continuous hidden markov model representation," *AT&T Technical Journal*, vol. 64, págs. 1251–1269, 1985. 2.3, 2.3.1

[38] L.R.Rabiner y B.H.Juang, "An introduction to Hidden Markov Models," *IEEE Acoustics, Speech and Signal Processing Magazine*, págs. 4–16, 1986. 2.3, 2.3.2, 2.3.3

[39] B.H.Juang y L.R.Rabiner, "Mixture autorregressive hidden markov models for speech signals," *IEEE Transactions on ASSAP*, vol. 33, págs. 1404–1413, 1995. 2.3.1

[40] X.Huang, K.F.Lee, y H.W.Hon, "On semi-continuous hidden markov modeling," *Proceedings of ICASP-90*, págs. 689–692, 1990. 2.3.1

[41] J.C.Segura y A.J.Rubio, *Variantes del Modelado Oculto de Markov para Señales de Voz*. Monografias del Dpto.de Electrónica, Universidad de Granada, 1991, vol. 24. 2.3.1

[42] J.C.Segura, A.J.Rubio, A.M.Peinado, P.García, y R.Román, "Multiple VQ hidden markov modeling for speech recognition," *Speech Comunication*, págs. 163–170, 1994. 2.3.1

[43] A.M.Peinado, J.C.Segura, A.J.Rubio, y M.C.Benítez, "Using multiple vector quantization and semicontinuous hidden markov models for speech recognition," *in proceedings of ICASP*, 1994. 2.3.1

[44] A.M.Peinado, "Selección y estimación de parámetros en sistemas de reconocimiento automático de voz basados en modelos ocultos de markov," Tesis doctoral, Universidad de Granada, 1994. 2.3.1

[45] J.F.Jelinek, R.L.Mercer, y S.Roukos, *Advances in Speech Signal Processing*. Dekker, 1992. 2.3.2

[46] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *in proceedings of the IEEE*, vol. 77, págs. 257–285, 1989. 2.3.3

[47] L. Baum, "An inequality and associated maximization technique in statistical estimation of probabilistic functions of markov processes," *Inequalities*, págs. 1–8, 1972. 2.3.3

[48] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. IT-13, págs. 260–269, 1967. 2.3.3

[49] L. Bahl, F. Jelinek, y R. Mercer, "A maximum likelihood approach to continuous speech recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-5, núm. 2, págs. 179–190, 1983. 2.3.3

[50] C. Boulis, M. Ostendorf, E. Riskin, y S. Otterson, "Graceful degradation of speech recognition performance over packet-erasure networks," *IEEE Transactions on Speech and Audio Processing*, vol. 10, núm. 8, 2002. 2.4.3, 6.2.1, 6.3

[51] D.Pearce, "Enabling new speech driven services for mobile devices: An overview of the ETSI standards activities for distributed speech recognition front-ends," *in proceedings of AVIOS*, 2000. 2.5, 3.4.2, 3.6, 5.3.6, 6.2.3

[52] *Aurora project database 2&3*. Disponible en linea: http://www.icp.inpg.fr/ELRA/aurora2.html 2.5

[53] *Transmission performance characteristics of pulse code modulation channels*, ITU-T Recomendation G.712, 1996. 2.5

[54] W. Zhang, L. He, Y.-L. Chow, R. Yang, y Y. Su, "The study on distributed speech recognition system," *in proceedings of IEEE ICASSP*, vol. 3, págs. 1431–1434, 2000. 3.1

[55] T.S.Rappaport, *Wireless communications, Principles and Practice.* Prentice Hall PTR, 1996. 3.2.1

[56] M.D.Yacoub, *Cell Desing Principles.* Ed. Suthan and S. Suthersan, CRC Press LLC, 1999. 3.2.1

[57] S.M.Redl, M.K.Weber, y M.W.Oliphant, *An Introduccion to GSM.* Artech House Publishers, 1995. 3.2.3

[58] *Channel Coding*, ETSI EN 300 909 v8.5.1, 2000. 3.2.4

[59] *Substitution and muting of lost frames for Full Rate (FR) speech traffic channels*, ETSI ETS 300 580, 1994. 3.2.5

[60] *Substitution and muting of lost frames for Half Rate (HR) speech traffic channels*, ETSI ETS 300 581, 1998. 3.2.5

[61] *Substitution and muting of lost frames for Enhanced Full Rate (EFR) speech traffic channels*, ETSI EN 300 727 v8.0.1, 2000. 3.2.5

[62] *Substitution and muting of lost frames for Adaptive Multi Rate (AMR) speech traffic channels*, ETSI EN 301 705 v7.1.1, 2004. 3.2.5, 4.5

[63] J.Postel, "Internet protocol," *RFC 791*, 1981. 3.3.3

[64] P.Mockapetris, "Domain names - implementation and specification," *RFC 1035*, 1987. 3.3.3

[65] P.Mockapetris, "Domain names - concepts and facilities," *RFC 1034*, 1987. 3.3.3

[66] J.Postel, "Transmission control protocol," *RFC 793*, 1981. 3.3.3, 3.3.3

[67] J.Postel, "User datagram protocol," *RFC 768*, 1980. 3.3.3

[68] H.Schulzrinne, R.Frederick, y V.Jacobson, "RTP: A transport protocol for Real-Time applications," *RFC 1889*, 1996. 3.3.3

[69] S.Shenker y J.Wroclawski, "General characterization parameters for integrated service network elements," *RFC 2215*, 1997. 3.3.4

[70] S.Blake, D.Black, M.Carlson, E.Davies, Z.Wang, y W.Weiss, "An architecture for Differentiated Services," *RFC 2475*, 1998. 3.3.4

[71] B.Fox y B.Gleeson, "Virtual private networks identifier," *RFC 2685*, 1999. 3.3.4

[72] B.Gleeson, A.Lin, J.Heinanen, G.Armitage, y A.Malis, "A framework for IP based virtual private networks," *RFC 2764*, 2000. 3.3.4

[73] P.Haavisto, "Speech recognition for mobile comunications," *in proceedings of the COST Workshop on Robust Methods for Speech Recognition in Adverse Conditions*, 1999. 3.4.2, 3.5, 3.6

[74] *Distributed speech recognition; Front-end feature extraction algorithm; Compression algorithms*, ETSI ES 201 108 v1.1.2, 2000. 3.4.2, 3.5, 5.2.2, 5.3.6, 6.2.2, 6.2.4, 6.4.4

[75] *Advanced front-end feature extraction algorithm*, ETSI ES 202 050 v1.1.1, 2002. 3.4.2

[76] D. Pearce, "Developing the ETSI aurora advanced distributed speech recognition front-end & what next?" *IEE Colloquium on Interactive Spoken Dialogue Systems for Telephony Applications*, págs. 131–134, 2001. 3.4.2

[77] *Distributed Speech Recognition; Extended Front-end Feature Extraction Algorithm; Compression Algorithm, Back-end Speech Reconstruction Algorithm*, ETSI ES 202 211, 2003. 3.4.2, 3.5

[78] *Distributed Speech Recognition; Extended Advanced Front-end Feature Extraction Algorithm; Compression Algorithm*, ETSI ES 202 212, 2005. 3.4.2

[79] Q.Xie, D.Pearce, S.Balasuriya, Y.Kim, S.H.Maes, y H.Garudari, "RTP payload format for DSR ES 201 108," *IETF Audio Video Transport WG, Internet RFC 3557*, 2002. 3.4.2, 3.5, 3.7.2, 5.2.2, 5.3.6, 6.2.2, 6.2.4, 6.2.5, 6.4.4

[80] C. Bormann, C. Burmeister, M. Degermark, H. Fukushima, H. Hannu, L.-E. Jonsson, R. Hakenberg, T. Koren, K. Le, Z. Liu, A. Martensson, A. Miyazaki, K. Svanbro, T. Wiebke, T. Yoshimura, y H. Zheng, "RObust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed," *RFC 3095*, 2001. 3.4.2

[81] S.Casner y V.Jacobson, "Compressing IP/UDP/RTP headers for low-speed serial links," *RFC 2508*, 1999. 3.4.2

[82] S.Euler y J.zinke, "The influence of speech coding algorithms on automatic speech recognition," *in proceedings of IEEE ICASSP*, págs. 621–624, 1994. 3.5, 3.6

[83] L.Fissore, F.Ravera, y C.Vair, "Speech recognition over GSM: Specific features and performance evaluation," *in proceedings of the COST workshop on robust methods for speech recognition in adverse conditions*, 1999. 3.5, 3.6

[84] B. T. Lilly, "Effect of speech coders on speech recognition performance," *in proceedings of INTERSPEECH-ICSLP*, vol. 4, págs. 2344–2347, 1996. 3.5

[85] J.M.Huerta y R.M.Stern, "Speech recognition from GSM codec parameters," *in proceedings of INTERSPEECH-ICSLP*, vol. 4, págs. 1463–1466, 1998. 3.5, 4.1

[86] B.Raj, J.Migdal, y R.Singh, "Distributed speech recognition with codec parameters," *in proceedings of IEEE Workshop on Automatic Speech Recognition and Underestanding (ASRU)*, págs. 127–130, 2001. 3.5

[87] G.N.Ramaswamy y P.S.Gopalakrishnan, "Compression of acoustic features for speech recognition in network environments," *in proceedings of IEEE ICASSP*, 1998. 3.5

[88] D.Chazan, G.Cohen, R.Hoory, y M.Zibulski, "Low bit rate speech compression for playback in speech recognition systems," *in proceedings of European Signal Processing Conference (EUSIPCO)*, 2000. 3.5

[89] D.Chazan, G.Cohen, R.Hoory, y M.Zibulski, "Speech reconstruction from mel-frequency cepstral coefficients and pitch frequency," *in proceedings of IEEE ICASSP*, 2000. 3.5

[90] J.V.Sciver, J.Ma, F.Vanpouke, y H. Hamme, "Investigation of speech recognition over IP channels," *in proceedings of IEEE ICASSP*, vol. 4, págs. 3812–3815, 2002. 3.5

[91] C.Pelaez-Moreno, A.Gallardo-Antolin, y F. de Maria, "Recognizing voice over IP: a robust front-end for speech recognition on the world wide web," *IEEE Transactions on Multimedia*, vol. 3, 2001. 3.5, 4.1

[92] D.Quercia, L.Docio-Fernandez, C.Garcia-Mateo, L.Farinetti, y J.C.Martin, "Performance analysis of distributed speech recognition over IP networks on the aurora database," *in proceedings of IEEE ICASSP*, 2002. 3.5

[93] T.Fingscheidt, S.Aalburg, S.Stan, y C.Beaugeant, "Network-based vs. Distributed Speech Recognition in Adaptative Multi-Rate wireless systems," *in proceedings of INTERSPEECH-ICSLP*, 2000. 3.5

[94] C. Mokbel, L. Mauuary, L. Karray, D. Jouvet, J. Monné, J. Simonin, y K. Bartkova, "Towards improving ASR robustness for PSN and GSM telephone applications," *Speech Communication*, vol. 23, págs. 141–159, 1997. 3.6

[95] J.M.Huerta y R.M.Stern, "Distortion-class modeling for robust speech recognition under GSM RPE-LTP coding," *Speech Communication*, vol. 34, 2001. 3.6

[96] H-G.Hirsch, "The influence of speech coding on recognition performance in telecommunication networks," *in proceedings of INTERSPEECH-ICSLP*, págs. 1877–1880, 2002. 3.6, 4.5.2

[97] I. Kiss, "A comparison of distributed and network speech recognition for mobile communication systems," *in proceedings of INTERSPEECH-ICSLP*, 2000. 3.6

[98] J.Bolot, "End-to-end packet delay and loss behavior in the internet," *ACM Sigcomm*, págs. 289–298, 1993. 3.7.1

[99] M.Yajnik, S.Moon, J.Kurose, y D.Towsley, "Measurement and modelling of the temporal dependence in packet loss," *in proceedings of IEEE INFOCOM*, 1999. 3.7.1

[100] W.Jiang y H.Schulzrinne, "Modeling of packet loss and delay and their effect on Real-Time multimedia service quality," *in proceedings of NOSSDAV*, 2000. 3.7.1

[101] M.Borella, D.Swider, y S.Uludag, "Internet packet loss: Measurement and implications for end-to-end QoS," *in proceedings of International conference on Parallel Processing*, 1998. 3.7.1

[102] N.F.Maxemchuk y S.Lo, "Measurement and interpretation of voice traffic on the internet," *in proceedings of ICC*, 1997. 3.7.1

[103] M.Yajnik, S.Moon, J.Kurose, y D.Towsley, "Measurement and modelling of the temporal dependence in packet loss," *in proceedings of IEEE INFOCOM*, 1999. 3.7.1, 3.7.1, 3.7.1

[104] J.Bolot, "Adaptive FEC-based error control for Internet telephony," *in proceedings of IEEE INFOCOM*, vol. 3, págs. 1453–1460, 1999. 3.7.1

[105] H.Sanneck y G.Carle, "A framework model for packet loss metrics based on loss runlengths," *in proceedings of IEEE Global Internet*, págs. 554–557, 1996. 3.7.1, 3.7.1

[106] R.Koodli y R.Ravikanth, "One-way loss pattern sample metrics," *Internet Draft, Internet Engineering Task Force*, 1999. 3.7.1, 3.7.1

[107] B.Milner, "Robust speech recognition in burst-like packet loss," *in proceedings of IEEE ICASSP*, vol. 1, págs. 261–264, 2001. 3.7.1

[108] B. Milner y A. James, "Robust speech recognition over mobile and IP networks in burst-like packet loss," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, págs. 223–231, 2006. 3.7.2, 6.3.2

[109] A. James, A.M.Gómez, y B. Milner, "A comparison of packet loss compensation methods and interleaving for speech recognition in burst-like packet loss," *in proceedings of INTERSPEECH-ICSLP*, 2004. 3.7.2, 5.2.3, 6.3.2, 6.4.2, 7.2, A

[110] J.M.Huerta, "Speech recognition in mobile environments," Tesis doctoral, Carnegie Mellon University, 2000. 4.1

[111] A. Gallardo-Antolin, F. D. de Maria, y F.Valverde-Albacete, "Recognition from GSM digital speech," *in proceedings of INTERSPEECH-ICSLP*, 1998. 4.1

[112] A. Gallardo-Antolin, F. D. de Maria, y F.Valverde-Albacete, "Avoiding distorsion due to speech coding and transmission errors in GSM ASR tasks," *in proceedings of IEEE ICASSP*, vol. 1, págs. 277–280, 1999. 4.1

[113] H.K.Kim y V.Cox, "A bitstream-based front-end for wireless speech recognition on IS-136 communications system," *IEEE Transactions on Speech and Audio Processing*, vol. 9, núm. 5, 2001. 4.1

[114] C. Pelaez-Moreno, "Reconocimiento de habla mediante transparametrización: Una alternativa robusta para entornos móviles e IP," Tesis doctoral, Universidad Carlos III, 2002. 4.1

[115] N.S.Jayant y P.Noll, *Digital Coding of Waveforms.* Prentice Hall, Englewood Cliffs, NJ, 1984. 4.2

[116] *Pulse code modulation (PCM) of voice frequencies*, ITU-T Recomendation G.726, 1988. 4.2

[117] J.D.Gibson, "Adaptative prediction in speech differential encoding system," *Proc.IEEE*, págs. 488–525, 1980. 4.2

[118] *40, 32, 24, 16 kbit/s adaptive differential pulse code modulation (ADPCM)*, ITU-T Recomendation G.726, 1990. 4.2

[119] H.Dudley, "The vocoder," Bell Labs Rec., Tech. Rep., 1939. 4.2

[120] J.L.Flanagan, *Speech Analysis, Synthesis and Perception.* Springer-Verlag, New York, 1972. 4.2

[121] E.Peterson y F.S.Cooper, "Peakpicker: A bandwidth compression device," *Journal of the Acoustical. Society of America*, págs. 777–782, 1957. 4.2

[122] J.N.Holmes, "The JSRU channel vocoder," *in proceedings of IEEE*, vol. 127-1, págs. 53–60, 1980. 4.2

[123] R.J.McAuley y T.F.Quatieri, "Speech analysis/synthesis based on a sinusoidal representation," *IEEE Transactions Acoustics, Speech Signal Processing*, vol. 34, págs. 744–754, 2003. 4.2

[124] L.R.Rabiner y R.W.Schafer, *Digital Processing of Speech Signals.* Prentice-Hall, 1978. 4.2.1, 5.3.1

[125] F. S. 1015, *Analog to digital conversion of radio voice by 2400 bit/second linear predictive coding*, National Communication System, Office of Technology and Standards, 1984. 4.2.1, 4.2.3

[126] R.P.Ramachandran y P.Kabal, "Pitch prediction filters in speech coding," *IEEE Transactions on Acoustics, Speech and Signal Processing*, págs. 467–478, 1989. 4.2.1, 4.2.3

[127] *Full rate speech;Transcoding*, ETSI EN 300 961 V8.0.2, 2000. 4.2.2

[128] M.R.Schroeder y B.S.Atal, "Code-excited linear prediction (CELP): High quality speech at very low bit rates," *in proceedings of IEEE ICASSP*, págs. 937–940, 1985. 4.2.2

[129] P.Kroon, E.F.Deprettere, y R.J.Sluyter, "Regular-pulse excitation: A novel approach to effective and efficient multipulse coding of speech," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 34, págs. 1054–1063, 1986. 4.2.3

[130] *Enhanced Full rate (EFR) speech transcoding*, ETSI EN 300 726 v8.0.1, 2000. 4.4.1

[131] R.Salami, C.Laflamme, B.Bessette, J-P.Adoul, K.Jarvinen, J.Vainio, P.Kapaenen, T.Honkanen, y P.Haavisto, "Description of GSM enhanced full rate speech codec," *in proceedings of IEEE International Conference on Communications*, vol. 2, págs. 725–729, 1997. 4.4.1, 4.4.1

[132] *Pulse code modulation (PCM) of voice frequencies*, ITU-T Recomendation G.711, 1988. 4.6

[133] *Digital Cellular Telecommunications System;Inband Tandem Free Operation (TFO) of speech codecs; Service description.*, ETSI TS 128 062 v5.4.0, 2003. 4.6

[134] C.Perkins, O.Hodson, y V.Hardman, "A survey of packet-loss recovery techniques for streaming audio," *IEEE Network Magazine*, 1998. 5.1

[135] J.G.Gruber y L.Strawczynski, "Subjective effects of variable delay and clipping in dynamically managed voice systems," *IEEE Transactions on Communications*, vol. 33, núm. 8, págs. 801–808, 1985. 5.2.1

[136] A.Bernard y A.Alwan, "Low-bitrate distributed speech recognition for packet-based and wireless communication," *IEEE Transactions on Speech and Audio Processing*, vol. 10, págs. 570–579, 2002. 5.2.1

[137] V.Hardman, "Reliable audio for use over the internet," *in proceedings of INET*, 1995. 5.2.2

[138] G.A.Miller y J.C.R.Licklider, "The intelligibility of interrupted speech," *Journal of the Acoustic Society of America*, vol. 22, núm. 2, págs. 167–173, 1950. 5.2.2

[139] R.M.Warren, *Auditory Perception.* Pergamon Press Inc., 1982. 5.2.2

[140] D.Quercia, L.Docio-Fernandez, C.Garcia-Mateo, L.Farinetti, y J.C.Martin, "Performance analysis of distributed speech recognition over IP networks on the aurora database," *in proceedings of IEEE ICASSP*, 2002. 5.2.2

[141] B.Milner y S.Semnani, "Robust speech recognition over IP networks," *in proceedings of IEEE ICASSP*, vol. 3, págs. 1791–1794, 2000. 5.2.3

[142] A. Peinado, V. Sanchez, J. Perez-Cordova, y A. Torre, "HMM-based channel error mitigation and its application to distributed speech recognition," *Speech Communication*, núm. 41, págs. 549–561, 2002. 5.2.3, 5.3.1, 6.2.3, 6.2.3

[143] Z. Tan, B. Lindberg, y P. Dalsgaard, "A comparative study of feature-domain error concealment techniques for distributed speech recognition," *in proceedings of Robust 2004 (Cost 278 and ITRW workshop*, 2004. 5.2.3

[144] A.M.Peinado, V.Sánchez, J.C.Segura, y J.L.Pérez-Córdoba, "MMSE-based channel error mitigation for distributed speech recognition," *in proceedings of EUROSPEECH*, págs. 2707–2710, 2001. 5.3.1, 6.2.3

[145] A.M.Peinado, V.Sánchez, J.L.Pérez-Córdoba, J.C.Segura, y A.J.Rubio, "HMM-methods for channel error mitigation in distributed speech recognition," *in proceedings of INTERSPEECH-ICSLP*, págs. 2205–2208, 2002. 5.3.1, 6.2.3

[146] A. Aho, J. Hopcroft, y J. Ullman, *Estructuras de datos y algoritmos.* Addison-Wesley, 1988. 5.3.5

[147] G. Correa, *Desarrollo de Algoritmos y sus aplicaciones.* MacGraw - Hill Inc., 1992. 5.3.5

[148] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 3º ed. McGraw Hill, Inc., 1991. 5.4.1

[149] B. R. Ramakrishnan, "Reconstruction of incomplete spectrograms for robust speech recognition," Tesis doctoral, Carnegie Mellon University, 2000. 5.4.1, 5.4.1

[150] H. C. Thomas, E. L. Charles, y L. R. Ronald, *Introduction to Algorithms.* The MIT Press, Mc Graw-Hill, 1996. 5.4.4

[151] J.Rosenberg y H.Schulzrinne, "An RTP payload format for generic forward error correction," *IETF Audio/Video Transport WG*, 1998. 6.2.1

[152] I. Reed y G. Solomon, "Polynomial codes over certain finite fields," *J. SIAM*, vol. 8, núm. 2, págs. 300–4, 1960. 6.2.1

[153] E. Riskin, C. Boulis, S. Otterson, y M. Ostendorf, "Graceful degradation of speech recognition performance over lossy packet networks," *in proceedings of Eurospeech*, 2001. 6.2.1

[154] A. James y B. Milner, "An analysis of interleavers for robust speech recognition in burst-like packet loss," en *in proceedings of IEEE ICASSP*, Montreal, Canada, 2004. 6.2.1, 6.3.2

[155] A. Peinado, P. Ramesh, y D. Roe, "On the use of energy information for speech recognition using HMMS," *in proceedings of EUSIPCO*, vol. 2, págs. 1243–6, 1990. 6.2.2

[156] B. Delaney, "Increased robustness against bit errors for distributed speech recognition in wireless environments," *in proceedings of IEEE ICASSP*, 2005. 6.3

[157] K. Andrews, C. Heegard, y D. Kozen, "A theory of interleavers," Computer Science Department, Cornell University, Tech. Rep. 97-1634, 1997. 6.3, 6.3.1, 6.3.2, 6.3.2

[158] J. Ramsey, "Realization of optimum interleavers," *IEEE Transactions on Information Theory*, vol. 6, págs. 338–45, 1970. 6.3.1, 6.3.3

[159] A. Gómez, A. Peinado, V. Sánchez, y A. Rubio, "On the ramsey class of interleavers for robust speech recognition in burst-like packet loss," *Enviado para segunda revisión en IEEE Transactions on Speech and Audio Processing*, 2006. 6.3.3, A

[160] A.Bernard y A.Alwan, "Joint channel decoding - viterbi recognition for wireless applications," *in proceedings of Eurospeech*, vol. 4, págs. 2703–2706, 2001. 6.4, 6.4.1

[161] A.Bernard y A.Alwan, "Low-bitrate distributed speech recognition for packet-based and wireless communication," *IEEE Transactions on Speech and Audio Processing*, vol. 10, págs. 570–579, 2002. 6.4, 6.4.1, 6.4.2, 6.4.2

[162] A. Cardenal-Lopez, L. Docio-Fernandez, y C. Garcia-Mateo, "Soft decoding strategies for distributed speech recognition over IP networks," *in proceedings of IEEE ICASSP*, vol. 1, págs. 49–52, 2004. 6.4, 6.4.1, 6.4.2, 6.4.2

[163] A. Cardenal-Lopez y C. Garcia-Mateo, "Correlation based soft-decoding for distributed speech recognition over IP networks," *in proceedings of Robust 2004 (Cost 278 and ITRW workshop*, 2004. 6.4, 6.4.1, 6.4.2

[164] R. Schwartz, Y. Chow, O. Kimball, S. Roucos, M. Krasner, y J. Makhoul, "Context-dependent modeling for acoustic-phonetic recognition of continous speech," *in proceedings of IEEE ICASSP*, 1985. 6.4.1

[165] A. James y B. Milner, "Soft decoding of temporal derivatives for robust distributed speech recognition in packet loss," *in proceedings of IEEE ICASSP*, vol. 1, págs. 345–8, 2005. 6.4.2, 6.4.2, 6.4.2

[166] A.M.Gómez, A.M.Peinado, V.Sánchez, y A.J.Rubio, "Estudio preliminar de los errores de transmisión con codificación EFR y su influencia en el reconocimiento de voz," *II Jornadas en Tecnologías del Habla*, 2002. 7.2, A

[167] A. Gómez, A. Peinado, V. Sánchez, y A. Rubio, "Mitigation of channel errors in EFR-based speech recognition," *in proceedings of IEEE ICASSP*, vol. 1, págs. 1021–4, 2004. 7.2, A

[168] A. Gómez, A. Peinado, V. Sánchez, y A. Rubio, "Recognition of coded speech transmitted over wireless channels," *IEEE Transactions on Wireless Comunications (In Press)*, 2004. 7.2, A

[169] A.M.Gómez, A.M.Peinado, V.Sánchez, y A.J.Rubio, "A source model mitigation technique for distributed speech recognition over lossy packet channels," *in proceedings of EuroSpeech*, 2003. 7.2, A

[170] A. Gómez, A. Peinado, V. Sánchez, B. Milner, y A. Rubio, "Statistical-based reconstruction methods for speech recognition in IP networks," *in proceedings of the COST 278, Robustness Issues in Conversational Interaction*, 2004. 7.2, A

[171] A. Gómez, A. Peinado, V. Sánchez, J. Pérez-Córdoba, y A. Rubio, "Técnica combinada de reconstrucción estadística de la voz para reconocimiento robusto en redes IP," *III Jornadas en Tecnologías del Habla*, págs. 21–26, 2004. 7.2, A

[172] A. Peinado, A. Gómez, V. Sánchez, y A. Rubio, "Packet loss concealment based on VQ replicas and MMSE estimation applied to distributed speech recognition," *in proceedings of IEEE ICASSP*, vol. 1, págs. 329–332, 2005. 7.2, A

[173] A. Gómez, A. Peinado, V. Sánchez, y A. Rubio, "Combining media-specific FEC and error concealment for robust distributed speech recognition over loss-prone packet channels," *IEEE Transactions on Multimedia (In Press)*, 2005. 7.2, A

[174] A. Gómez, A. Peinado, V. Sánchez, J. Carmona, y A. Rubio, "Interleaving and MMSE estimation with VQ replicas for distributed speech recognition over lossy packet networks," *in proceedings of INTERSPEECH-ICSLP*, 2006. 7.2, A