Reconocimiento de Voz Codificada sobre Redes IP

José Luis Carmona Maqueda

Dpto. de Teoría de la Señal Telemática y Comunicaciones

Universidad de Granada

Editor: Editorial de la Universidad de Granada
Autor: José Luis Carmona Maqueda
D.L.: GR. 2612-2009
ISBN: 978-84-692-3873-8

D. Antonio M. Peinado Herreros y **D. José L. Pérez Cordoba**, Profesores Titulares de Universidad del Departamento de Teoría de la Señal, Telemática y Comunicaciones

CERTIFICAN:

Que la memoria titulada: **"Reconocimiento de Voz Codificada sobre Redes IP"** ha sido realizada por **José Luis Carmona Maqueda** 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 con Mención Europea en Ingeniería de Telecomunicación.

Granada, a 6 de Mayo de 2009

Fdo. Antonio M. Peinado Herreros
Director de la Tesis

Fdo. José L. Pérez Córdoba
Director de la Tesis

Fdo. José Luis Carmona Maqueda
Doctorando

A mis padres...
...y a Oliva por darme lo más preciado del mundo.

No hay más que un modo de dar una vez en el clavo,
y es dar ciento en la herradura.
Miguel de Unamuno

Lo que no te mata te hace fuerte.
Friedrich Nietzsche

Agradecimientos

Haciendo balance de los años que han supuesto la elaboración de este trabajo, las dificultades, las dudas y, por qué no, los buenos y malos momentos, me gustaría, en primer lugar, agradecer a Ángel Gómez García y José Andrés González López su inestimable colaboración, sin la cual esta tesis no habría podido llevarse a cabo, a Victoria Sánchez Calle su importante apoyo, tanto profesional como personal, y a todo el resto de compañeros de este departamento su cercanía personal.

Por supuesto, tengo la obligación de agradecer a Antonio Peinado Herreros y José Luis Pérez Córdoba su enorme interés y dedicación, así como su ilusión y guía.

Además, tengo también una deuda impagable con mis familiares, amigos y todas las demás personas que han estado a mi lado durante todo este tiempo. En particular, con mi *compañera de piso* Oliva que siempre estuvo a mi lado durante las horas más bajas. Tampoco quiero olvidar a mi hermano César por su cariño y comprensión ingenieril. Finalmente, quiero agradecer, como no podía ser de otra forma, a mis padres el haberme brindado las oportunidades que ellos no tuvieron.

Resumen

El rápido desarrollo de diversas redes inalámbricas, como por ejemplo *3G*, redes *WiFi* y *Bluetooth*, ha propiciado que los terminales móviles incrementen sustancialmente su conectividad. Paralelamente, estos dispositivos tienden a reducir sus dimensiones para aumentar su portabilidad, lo que dificulta la accesibilidad de sus interfaces. Esta tendencia imposibilita, en cierta medida, el desarrollo de nuevos servicios sobre redes inalámbricas. Por ello, se hace necesario el desarrollo de nuevos interfaces, que proporcionen una fácil interacción multimodal, para la próxima generación de dispositivos móviles. En este escenario, el reconocimiento automático del habla es un camino prometedor para un acceso fácil y natural a nuevas aplicaciones. Sin embargo, los terminales móviles se caracterizan por tener una capacidad de cómputo restringida, así como una duración de batería limitada. El reconocimiento remoto de voz permite salvar estas restricciones ubicando las tareas de mayor coste computacional en un servidor remoto, es decir, fuera del propio dispositivo móvil.

Uno de los aspectos claves del panorama actual de las telecomunicaciones es la convergencia de las distintas redes inalámbricas hacia las redes IP. Así, en un futuro cercano, este tipo de redes presentarán un alcance prácticamente global, posibilitando el reconocimiento remoto de la voz de forma ubicua. Como desventaja, estas redes resultan propensas a pérdidas de paquetes, ya que no fueron originariamente diseñadas para la transmisión de datos en tiempo real.

El interés de esta tesis se centra en el análisis de las degradaciones ocasionadas por las pérdidas de paquetes sobre el reconocimiento de voz codificada, así como la propuesta y posterior desarrollo de soluciones para prevenir, reducir y compensar los efectos degradantes. El rendimiento de la arquitectura de reconocimiento remoto vendrá supeditado a la robustez del esquema de codificación de voz utilizado. Los codificadores convencionales consiguen reducir

sustancialmente la tasa de transmisión haciendo uso de técnicas predictivas que explotan las correlaciones temporales de la voz. No obstante, estas técnicas predictivas introducen fuertes dependencias intertrama, de modo que una pérdida de un paquete no sólo afecta al segmento de voz correspondiente, sino que además genera una propagación de error en los paquetes posteriores, reduciendo severamente la precisión del reconocimiento. Por otro lado, los decodificadores integran sus propios algoritmos de mitigación de pérdidas, los cuales están basados en consideraciones perceptuales que no son adecuadas para las tareas de reconocimiento. Para combatir estas degradaciones, proponemos diferentes técnicas, las cuales pueden dividirse en dos clases en función de si actúan en el emisor o en el receptor.

En cuanto a las técnicas basadas en el emisor, en esta tesis realizamos dos propuestas. La primera de ellas consiste en llevar a cabo una combinación de diferentes esquemas de codificación, mezclando tramas independientes y dependientes, de modo que, por un lado, se consigue limitar la posible propagación de error (tramas independientes) y, por otro, se obtiene una tasa de codificación moderada (tramas dependientes). La segunda de las propuestas emplea codigos FEC específicos, basados en la codificación multipulso, que permiten reducir la propagación de error mediante un incremento limitado de la tasa de codificación. Ambas técnicas, además de aumentar la precisión de reconocimiento, consiguen mejorar la calidad perceptual en la síntesis de voz.

Por otra parte, también proponemos un conjunto de técnicas de mitigación basadas en el receptor. En este caso, el proceso de reconstrucción se lleva a cabo mediante estimación MMSE. Esta técnica consigue buenos resultados ya que emplea un modelado estadístico de la evolución temporal de la voz y de las distorsiones originadas por las pérdidas. Además, esta técnica permite determinar valores de confianza asociados a las reconstrucciones realizadas, los cuales pueden ser utilizados para tratar las pérdidas en el propio reconocedor.

Finalmente, en esta tesis se proponen esquemas para la transformación directa de los parámetros de voz codificada en vectores de características para el reconocimiento, a los que nos referimos como transaparametrizadores. Estas soluciones permiten soslayar ciertas consideraciones perceptuales del proceso de decodificación de voz, las cuales no son oportunas para el reconocimiento, así como adaptar eficientemente las técnicas basadas en el receptor.

Abstract

The fast development of diverse wireless networks, such as 3G, wireless LAN or Bluetooth, favours that networking facilities are becoming a standard component on mobile devices. Nevertheless, the accessibility of these terminals, which tend to be lighter and smaller, hampers the development of new services over wireless networks. This trend makes it more difficult, or even frustrates, the interaction of the user with the service. Thus, the development of new user interfaces, providing a ubiquitous, pervasive and multimodal interaction, is a necessary step for the next generation of mobile services. In this scene, automatic speech recognition is a promising way for an easy and natural user access to network services. However, mobile devices are characterized by a restricted computing power, small limited-speed memories and short battery life. Remote speech recognition allows circumventing these hardware constraints by moving the most complex computational tasks of speech recognition to a remote server.

Under this approximation, the user device has to send coded speech or speech parameters through a communication channel. One of the key aspects in the current state of telecommunications is the convergence of wireless networks towards IP. Thus, in the near future, IP networks will have an almost global deployment that will allow remote speech recognition and access to information services whenever and wherever it is required. However, as disadvantage, these networks exhibit a loss-prone packet transmission, since they were not designed for real time communications. As can be expected, these packet losses have a very negative impact on recognition performance.

In this dissertation, the influence of the aforementioned packet losses on speech recognition is analyzed and different solutions to prevent, reduce and conceal their effects are developed. The performance of remote speech recognition will depend on the robustness of the speech coding scheme employed. Conventional speech codecs reduce the bit-rate by means of predictive techniques that

exploit the temporal speech correlations. Thus, to decode a frame, a correct decoding of the previous ones is required. However, this inter-frame dependency considerably reduces the robustness against packet losses because it originates an error propagation in addition to the associated information loss. Furthermore, speech decoders integrate their own packet loss concealment algorithms, which are based on perceptual considerations that are unsuitable for speech recognition. In order to combat these degradations, we propose a set of mechanisms which can be grouped into sender-driven and receiver-based techniques.

In the present Ph.D. thesis we propose two sender-driven techniques. The first one combines intraframe-coded and interframe-coded speech frames in order to, on the one hand, limit the possible error propagation (intraframe-coded frames) and, on the other, obtain a moderate bit-rate (interframe-coded frames). The second proposal uses specific FEC codes, based on multi-pulse coding, that reduce the error propagation. In addition to an improvement of the recognition accuracy, both techniques improve the perceptual quality of synthesized speech.

On the other hand, we also propose a set of receiver-based reconstruction techniques. In this case, the reconstruction process is carried out by means of MMSE estimation. This technique offers good results since it models the temporal speech evolution and the distortion caused by packet losses. Also, this technique allows obtaining confidence values about the estimates, which can be used in order to deal with packet losses in the recognizer itself.

Finally, in this Ph.D. Thesis, we propose schemes that directly transcode the codec parameters into recognition-oriented feature vectors. These solutions have the advantage of avoiding some perceptual issues of speech synthesis that are inappropriate for speech recognition. Furthermore, they allow an efficient adaptation of the receiver-based techniques previously proposed.

Índice general

Índice de figuras

ÍNDICE DE FIGURAS

ÍNDICE DE FIGURAS

Índice de tablas

Capítulo 1

Introducción

1.1. Motivación

El reconocimiento automático del habla ha experimentado fuertes avances en el transcurso de los últimos años. El esfuerzo investigador realizado en el campo del reconocimiento por una parte, y por otra la evolución de los computadores (cada día más potentes y baratos) han hecho posible que aplicaciones que parecían propias de la ciencia ficción sean posibles. En la actualidad, aplicaciones de dictado automático, la gestión del sistema operativo por voz o la incorporación de sistemas de control en automóviles activados por voz son una realidad. Además, con el exponencial desarrollo de Internet y de las distintas tecnologías de acceso, se abre un amplio abanico de posibilidades para los sistemas de reconocimiento del habla.

Las redes de conmutación de paquetes basadas en el protocolo IP (*Internet Protocol*), gracias a su enorme capacidad para la conexión de redes diversas, han dado origen a una red de redes global o Internet. La transmisión de voz sobre este tipo de redes, también denominada VoIP (*Voice over IP*), se ha visto significativamente incrementada en los últimos años, convirtiéndose en uno de los aspectos claves del panorama actual de las telecomunicaciones. Prueba de ello son los datos estimados por la ITU (*International Telecommunication Union*) en 2007 [1] (véase la figura 1.1) que apuntan a que en el año 2011 el número de subscriptores VoIP superará los 200 millones, alcanzando casi un 18 por cierto del número total de líneas telefónicas fijas.

Paralelamente a la convergencia de datos y voz que suponen las plataformas VoIP, la creación de los nuevos estándares de acceso inalámbrico a Internet ha propiciado la convergencia de las redes IP con las redes de telefonía celular, tal y como se muestra en

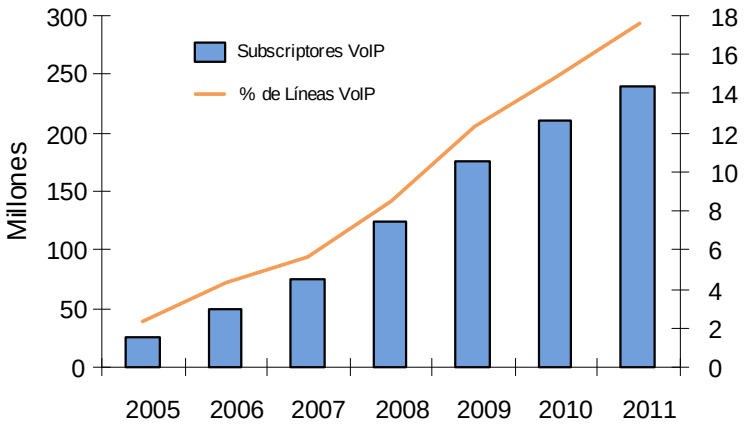


Figura 1.1: Estimación del número de subscriptores de VoIP a nivel mundial y su proporción sobre el número total de líneas de telefonía fija.

el diagrama de la figura 1.2, hacia una futura cuarta generación (4G). Así, en los últimos años, han aparecido teléfonos móviles celulares que incluyen conexión *bluetooth* (red PAN, *Personal Area Network*), wi-fi (red LAN, *Local Area Network*) y UMTS (*Universal Mobile Telecommunications System*). Concretamente, el gran crecimiento del número de redes WLAN (*Wireless LAN*) y la evolución de los servicios ofrecidos por éstas hacen prever que, en un futuro no muy lejano, las tecnologías del habla sobre IP se extiendan al dominio inalámbrico a través de estas redes locales [2]. Esta tendencia se verá reforzada a medio plazo con el desarrollo de otras tecnologías de acceso radio, tales como *Wi-MAX* (IEEE 802.16), *Mobile Broadband Wireless Access* (MBWA también conocida como IEEE 802.20), *Ultra Wideband* (UWB ó IEEE 802.15) ó *Long Term Evolution* (LTE), que en principio cubren distintos segmentos del mercado, pero cuyos límites de red son cada vez más difusos propiciando que estas tecnologías, inicialmente competidoras, se conviertan en complementarias. Bajo esta perspectiva, el terminal móvil del futuro podrá integrar varias de estas tecnologías y escoger aquella que le ofrezca mejores condiciones de acceso en cada situación. Este predecible paradigma proporcionará un nuevo concepto de acceso nómada, en un punto intermedio entre el acceso fijo y el acceso móvil, ofrecido por los proveedores de estas nuevas tecnologías y ligado a la incorporación de la tecnología IP.

La convergencia de las tecnologías inalámbricas hace que el acceso a la información mientras se está en movimiento sea hoy una realidad tecnológica que se encuentra en auge. No obstante, el pequeño tamaño de los nuevos dispositivos móviles dificulta la incorporación de interfaces, como por ejemplo el teclado. La interacción oral con dichos servicios se propone como un nuevo medio de acceso a la información, más rápido y mu-

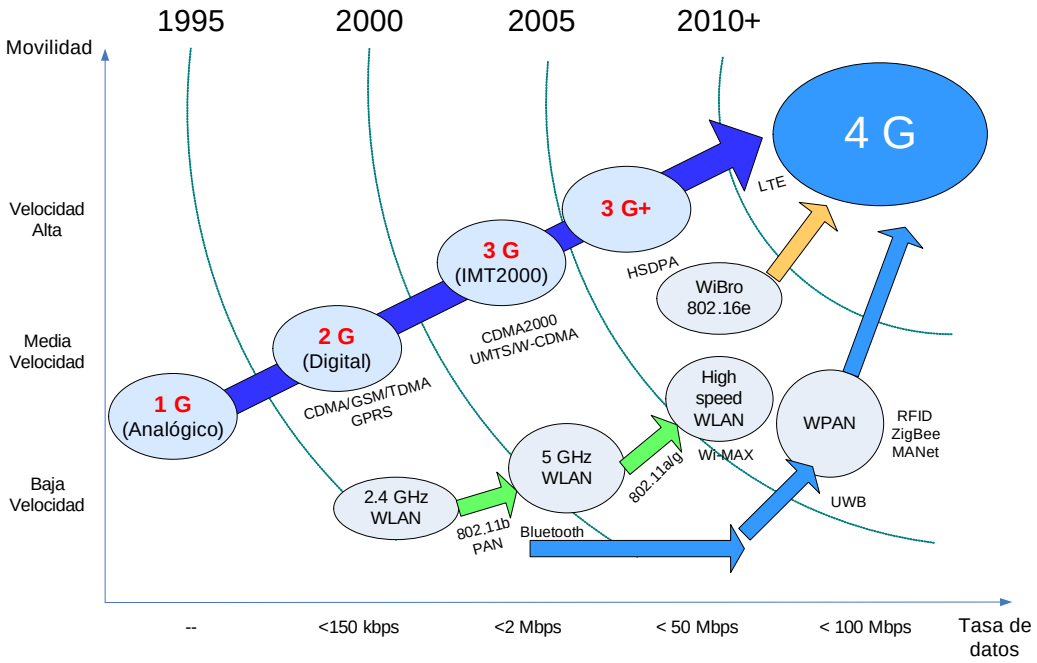


Figura 1.2: Diagrama de convergencia de las tecnologías inalámbricas hacia la cuarta generación.

cho más natural, donde el reconocimiento automático del habla encajaría perfectamente ofreciendo un servicio interactivo y de acceso rápido a la información (beneficiando al usuario), sin la necesidad de que al otro lado exista una persona que lo atienda (beneficiando al proveedor). Desgraciadamente, existen serios inconvenientes a la hora de proveer a los terminales móviles de un subsistema de reconocimiento automático de la voz. Principalmente la restricción en el tamaño de los dispositivos móviles limita la capacidad de cómputo condicionando la potencia y flexibilidad del reconocedor. Con el fin de superar estas limitaciones, surgió el planteamiento de realizar de forma remota el reconocimiento, es decir, fuera del propio terminal.

El reconocimiento remoto es una solución atractiva ya que presenta numerosas ventajas, como el empleo de terminales cliente más sencillos, disponibilidad de varios idiomas, actualización y mantenimiento de forma centralizada del servidor o la posibilidad de incorporar nuevas técnicas de reconocimiento más potentes. Además, la estructura de un sistema de reconocimiento remoto encaja perfectamente en el modelo IP, ya que el proveedor de servicio es el que implementa el reconocedor adaptándolo a sus necesidades. De este modo, el proveedor propone tanto los servicios como el reconocedor, adaptando este último a las necesidades de los primeros, lo que permite añadir fácilmente nuevos servicios

liberando al usuario de su mantenimiento. Ateniéndonos a este paradigma, terminales de bajo coste y prestaciones limitadas se conectan a potentes computadores remotos que realizan por ellos las tareas más complejas, lo que además supone un uso óptimo de los recursos centralizados.

La implementación más directa de un sistema de reconocimiento remoto consiste en transmitir la voz hasta el extremo servidor donde se lleva a cabo la tarea de reconocimiento. En este caso, la voz se comprime mediante un codificador de voz lo que permite obtener una baja tasa de transmisión. De este modo, el reconocimiento remoto se entiende como un servicio de valor añadido sobre VoIP, de modo que se transmite la voz codificada pero no con el objetivo de establecer una conferencia, sino de acceder a algún tipo de servicio. La principal ventaja de este tipo de implementación reside en que hace uso de las cada vez más extendidas plataformas VoIP, no siendo necesario introducir ningún tipo de modificación sobre el terminal cliente. No obstante, esta aproximación presenta a su vez ciertas desventajas ya que la pérdida de información que conllevan los esquemas de codificación de voz puede originar una pérdida de rendimiento del reconocedor. Además, se presentan ciertos problemas implícitos en el reconocimiento remoto. Entre estos problemas cabe destacar la presencia de ruido acústico (el contexto acústico del terminal puede ser muy diverso) y las degradaciones introducidas por el canal de comunicación. En este trabajo nos centramos en este segundo problema, ya que el diseño de las redes IP, que ofrecen un servicio *best effort* de entrega de paquetes, en general no garantiza que se satisfagan los requisitos de tiempo real (retardos y fluctuación de retardos acotados) y de fiabilidad (probabilidad de pérdidas acotadas, disponibilidad de mecanismos de recuperación) que la transmisión de flujos multimedia continuos requiere.

1.2. Objetivos

El objetivo general de esta tesis es el de proporcionar una serie de mecanismos que mejoren el rendimiento de los sistemas de reconocimiento remoto a partir de voz codificada frente a las degradaciones propias de las redes IP. En concreto, los mecanismos a diseñar deben hacer frente a las pérdidas de paquetes VoIP en la red, incrementando el rendimiento de las tareas de reconocimiento. En este sentido, es posible adoptar dos tipos de medidas en función de si son tomadas en el emisor o en el receptor. El primer tipo de estas medidas consiste en robustecer los esquemas de codificación de voz frente a la pérdida de paquetes. En este caso, las técnicas propuestas suponen introducir modificaciones sobre el esquema de codificación, de modo que deben tener presente un doble objetivo: incrementar la

calidad subjetiva perceptual de los sistemas de codificación y mejorar el reconocimiento de voz. El segundo tipo de medidas consistirá en el desarrollo de técnicas de mitigación de pérdidas para el reconocimiento de voz codificada. En este segundo caso, las técnicas implementadas modifican la estructura del servidor, de ahí que sólo contemplen la mejora del rendimiento del reconocedor dejando a un lado las consideraciones perceptuales.

Los objetivos particulares de esta tesis se pueden desglosar como sigue:

- Caracterizar el rendimiento de los sistemas de reconocimiento remoto a partir de voz decodificada en redes IP.
- Desarrollar esquemas de codificación de voz robustos frente a las degradaciones propias de las redes IP.
- Proporcionar técnicas de prevención de pérdidas en los esquemas de codificación con un incremento limitado de la tasa de transmisión.
- Evaluar las técnicas robustas de codificación desarrolladas de forma perceptual y en el reconocimiento de voz.
- Dotar de algoritmos de mitigación de pérdidas sobre la voz decodificada para mejorar el rendimiento de los sistemas de reconocimiento.
- Desarrollar esquemas de transparametrización que permitan extraer características de reconocimiento directamente de los parámetros de voz codificada.
- Adaptar eficientemente los algoritmos de mitigación desarrollados a los esquemas de transparametrización propuestos.

1.3. Estructura de la tesis

La presente tesis se estructura en 8 capítulos. En el capítulo 2 se introducen los conceptos implicados en el reconocimiento del habla, centrándonos en aquellos aspectos que resulten relevantes para el reconocimiento remoto, como la parametrización de la señal de voz y los enfoques de reconocimiento existentes. Dentro de éstos últimos, prestaremos especial interés al reconocimiento de voz basado en modelos ocultos de Markov, ya que se trata de la solución más extendida y en la que se basa el presente trabajo.

En el capítulo 3 se lleva a cabo una descripción detallada de las posibles arquitecturas de un sistema de reconocimiento de voz en terminales móviles, así como las posibles

implementaciones de éstas. Concretamente, veremos que las características y rendimiento de los sistemas de reconocimiento remoto a partir de voz codificada están intrínsecamente ligados al esquema de codificación de voz que se utilice. Por este motivo, en este capítulo se realiza una revisión sobre la tipología y fundamentos de los codificadores de voz existentes.

En las redes IP la información se envía de forma segmentada en paquetes. En particular, las exigencias de tiempo real en las comunicaciones VoIP conllevan que el envío de voz se realice de una forma no orientada a la conexión en la que parte de los paquetes enviados son susceptibles de ser perdidos por el canal. En el capítulo 4 se realiza un breve estudio sobre los orígenes de estas pérdidas y sus efectos sobre la codificación y reconocimiento de voz codificada, resultados que establecerán el punto de partida o *baseline* de este trabajo.

El hecho de utilizar técnicas predictivas en el desarrollo de los codificadores de voz conlleva que los efectos degradantes ocasionados por una pérdida se extiendan más allá del segmento perdido en sí. El capítulo 5 recoge diferentes propuestas para el robustecimiento de los esquemas de codificación más extendidos. Las soluciones propuestas se traducen en ciertas modificaciones de los codificadores que incrementan tanto la calidad perceptual de la voz reconstruida como el reconocimiento a partir de ésta.

Sin embargo, modificar el esquema de codificación presenta ciertas desventajas puesto que es necesario introducir modificaciones sobre el cliente. Por contra, en el capítulo 6 se proponen un conjunto de técnicas de mitigación de pérdidas en reconocimiento que no exigen realizar cambios en el transmisor.

En los capítulos mencionados, los parámetros de reconocimiento son extraídos a partir de la voz decodificada. No obstante, es posible llevar a cabo la extracción de características para el reconocimiento directamente a partir de los parámetros de codificación. Este esquema de reconocimiento recibe el nombre de transparametrización y es abordado en el capítulo 7, donde se proponen soluciones basadas en dicho esquema para los codificadores de voz más utilizados en redes IP.

Finalmente, el capítulo 8 se dedica a presentar las conclusiones, contribuciones realizadas y líneas futuras de investigación de este trabajo.

Capítulo 2

Reconocimiento Automático del Habla

2.1. Introducción

Una de las formas de comunicación más eficientes y naturales que poseemos es el habla. Sin duda, este medio de transmisión de ideas ha sido uno de los pilares en los que se ha apoyado nuestra civilización para alcanzar el actual grado de desarrollo social, intelectual y tecnológico. De ahí que el hombre siempre haya soñado con dotar a las máquinas con la capacidad de comprender mensajes orales. Ya en los cuentos de *Las mil y una noches*, Alí babá y los cuarenta ladrones utilizaban la voz para abrir la puerta de su cueva a la orden de "¡Ábrete, Sésamo!". Más recientemente, en multitud de películas aparecen máquinas con la capacidad de comunicación oral propia del ser humano.

Esta ficción no se comenzó a plasmar en la realidad hasta mediados del siglo XX cuando los laboratorios Bell presentaron un sistema electrónico capaz de reconocer, dentro de un margen de error aceptable, los dígitos pronunciados en inglés por un único locutor. No obstante, la gran revolución de estas tecnologías comenzó a gestarse con el advenimiento de los computadores digitales y la informática. Así, en la década de los 70 aparecen grandes proyectos de reconocimiento del habla por parte de países como EE.UU., Japón, Francia y Alemania. Sería a principios de esta década cuando el departamento de defensa de los EE.UU. decide financiar un ambicioso proyecto de investigación con el objetivo de desarrollar sistemas de reconocimiento de habla continua para varios locutores. Aunque los resultados fueron decepcionantes, ya que los objetivos no lograron alcanzarse, se realizaron importantes contribuciones ya que empezó a considerarse el conocimiento

sintáctico, semántico y contextual como útiles fuentes de información en el proceso de reconocimiento.

Es ya en los años 80 cuando se produce el giro metodológico fundamental con el modelado estadístico y el uso de los modelos ocultos de Markov o HMMs (*Hidden Markov Models*). A partir de ese momento, el reconocimiento del habla continua ha mejorado, aumentándose el tamaño de los vocabularios, diversificándose las aplicaciones y enfrentándose a situaciones cada vez más reales. En la actualidad, el desarrollo de nuevas técnicas, junto al espectacular desarrollo de los computadores, ha permitido grandes avances en el reconocimiento automático del habla, dando como resultado la aparición de aplicaciones comerciales de reconocimiento de voz continua.

En este capítulo se lleva a cabo una revisión de los principales conceptos implicados en el problema del reconocimiento automático del habla (RAH), así como de los distintos enfoques de reconocimiento existentes. Dentro de estos últimos, prestaremos especial atención al reconocimiento basado en modelos ocultos de Markov puesto que constituyen la solución más extendida y la base del presente trabajo.

2.2. Planteamiento General del Problema

Desde el punto de vista más simple, el RAH puede verse como una caja negra cuya entrada es la voz y cuya salida fuera el texto correspondiente al mensaje. De este modo, se conseguiría la interacción más natural entre hombre-máquina, tal y como se da fundamentalmente entre los humanos, mediante la voz. Por analogía con el propio ser humano, se suelen distinguir varios niveles de procesamiento [3]:

- 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 la etapa anterior se comparan y se identifican con las unidades sonoras básicas (palabras, fonemas, sílabas...) que conoce el receptor.

- Nivel Sintáctico. Empleando una serie de reglas, que conforman la gramática del lenguaje utilizado, las unidades sonoras básicas se combinan formando unidades conceptuales sintácticamente correctas.

- Nivel Semántico. En esta etapa se pretende realizar una comprensión del mensaje, eliminando interpretaciones sin sentido.

Las fronteras entre los diferentes niveles son difusas y, en todo caso, los niveles se encuentran absolutamente interrelacionados entre sí, no siendo posible aislar y estudiar un nivel de forma independiente, ya que no existe una secuencialización única entre estos niveles de operación.

Las tecnologías existentes suelen considerar el RAH como un problema de reconocimiento de formas, estructurado en un diagrama de bloques como el que se muestra en la figura 2.1. Las funciones de los distintos bloques son las siguientes:

- Adquisición. En este primer bloque se lleva a cabo la adquisición de la señal sonora, obteniendo finalmente una representación digital de ésta. Este bloque comprende el transductor acústico, así como los sistemas de amplificación, filtrado, muestreo y codificación precisos para la obtención de la señal digital.

- Análisis. Este bloque tiene como objetivo la obtención de un conjunto de características adecuado para llevar a cabo la tarea de reconocimiento. Por tanto, este bloque realiza un conjunto de operaciones que conducen a la vectorización de la señal de voz. Cuanto más discriminatoria sea esta información, tanto mejor pues los *vectores de características* finales permitirán una mayor precisión en el reconocimiento.

- Reconocimiento. Este último bloque consta de un conjunto de referencias que representan las unidades a reconocer. Así, mediante la comparación de los vectores de características extraídos durante la fase de análisis, se reconstruye el texto correspondiente al mensaje oral.

Desde una perspectiva general, podemos decir que los módulos de adquisición y análisis se encargan de adecuar la señal de voz para el procesado posterior, mientras que el reconocimiento constituye el núcleo del sistema. Dada la importancia de este último, podemos dividirlos en tres subbloques: un conjunto de referencias que representan las distintas unidades a reconocer; un módulo de comparación; y un módulo de decisión (véase el bloque de reconocimiento de la figura 2.1). Dependiendo de qué tipo de unidades básicas de reconocimiento se empleen, estas referencias podrán representar desde fonemas a frases completas. El módulo de comparación debe obtener, mediante algún tipo de métrica, una medida de similitud entre los vectores de características y el conjunto de referencias considerado. Finalmente, un mecanismo de decisión, que no tiene por qué formar una

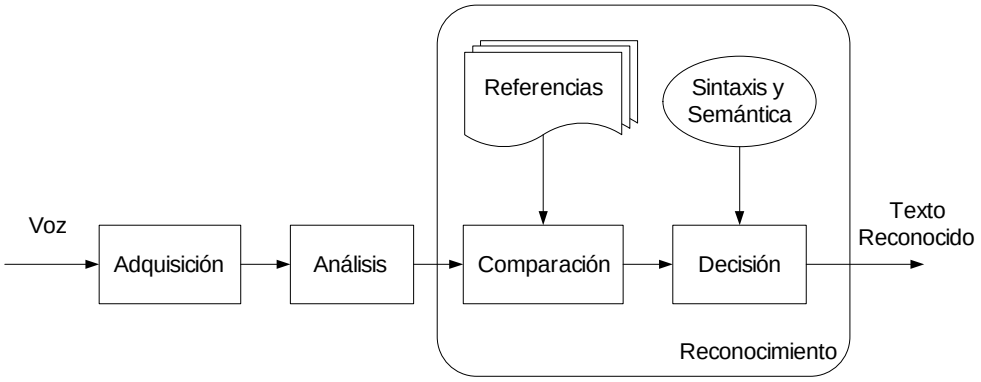


Figura 2.1: Diagrama de bloques de un sistema básico de reconocimiento.

unidad independiente, se encarga de decidir qué referencia aproxima mejor la entrada y, por tanto, cuál es el texto reconocido, teniendo en cuenta la sintaxis y semántica de la tarea de reconocimiento.

2.2.1. Clasificación de los Sistemas de Reconocimiento

El conjunto de referencias, que representan las unidades básicas a reconocer, se obtiene mediante una fase previa al reconocimiento denominada *fase de entrenamiento*. Así pues, en función de los condicionantes que se tengan en cuenta durante la fase de entrenamiento es posible distinguir diferentes categorías dentro de los sistemas de reconocimiento [4]. En primer lugar, podemos distinguir los siguientes dos tipos de sistemas:

- Dependientes del locutor. El sistema de reconocimiento sólo es capaz de tratar voz procedente de un conjunto cerrado de locutores y, en consecuencia, el entrenamiento se realiza a partir de locuciones de éstos. Estos sistemas son implementados cuando se cumple este tipo de restricción.

- Independientes del locutor. El conjunto de locutores de entrenamiento debe ser amplio, de modo que el reconocimiento de voz de un nuevo locutor sea suficientemente preciso.

En segundo lugar, los sistema de reconocimiento pueden ser también clasificados atendiendo a la unidad lingüística que utilicen. En principio, la unidad más natural es la palabra, sin embargo, esto puede originar un conjunto de referencias enorme que sería difícil de manejar. Cuando este problema se produce recurrimos a una unidad lingüística inferior a la palabra. Como unidad inmediatamente inferior podemos utilizar unidades fonéticas

independientes del contexto, las cuales corresponden a los fonemas básicos del lenguaje [5]. No obstante, estas unidades presentan el problema de que no modelan adecuadamente el efecto coarticulatorio entre fonemas consecutivos. Como respuesta a este problema, se han propuesto unidades del tipo bifonema, sílabas, semi-sílabas o trifonemas [6].

Atendiendo a los tipos de locuciones empleadas, los sistemas de reconocimiento de voz se pueden clasificar en:

- Reconocedores de palabras aisladas. En este caso las locuciones están formadas por palabras aisladas, o fácilmente aislables debido a la existencia de silencios entre palabras consecutivas, de modo que la tarea de reconocimiento se lleva a cabo de una en una (normalmente, utilizando la palabra como unidad).

- Reconocedores de voz continua. Al contrario del caso anterior, las locuciones forman frases que no necesariamente contienen pausas entre las palabras. En principio, puede utilizarse cualquier unidad de reconocimiento, aunque es común emplear unidades inferiores a la palabra. Este tipo de reconocedores proporciona textos ajustados a un conjunto de reglas o gramática. A su vez, el conjunto de frases generado por una cierta gramática es conocido como lenguaje. Así, para llevar a cabo correctamente el reconocimiento, el resultado obtenido debe pertenecer al lenguaje utilizado.

Finalmente, los sistemas de reconocimiento 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 llevar a cabo una fase de entrenamiento en entornos acústicos similares a aquellos en los que se va a operar. De este modo, por la forma de afrontar el problema del ruido acústico, los sistemas se pueden clasificar de forma general en robustos y no robustos dependiendo de que apliquen o no alguna técnica de compensación o robustecimiento frente al ruido [7].

2.3. Análisis de la Señal de Voz

El objetivo del bloque de análisis mostrado en la figura 2.1 es proporcionar una representación paramétrica de la señal de voz apropiada para el reconocimiento. El diagrama mostrado en la figura 2.2 representa el esquema de análisis comúnmente utilizado, el cual se encuentra basado en la representación espectral a corto plazo de la voz [8]. En primer

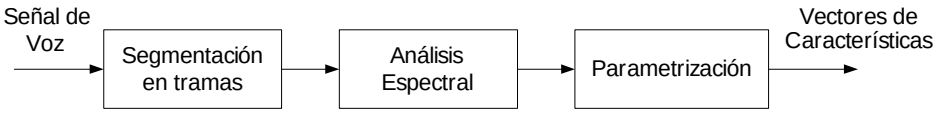


Figura 2.2: Esquema general de análisis de la señal de voz.

lugar, la voz es preprocesada y segmentada en tramas (20-40 ms) que suelen tener un cierto solapamiento. De este modo, si T_s representa el desplazamiento entre dos tramas consecutivas, entonces $1/T_s$ se corresponde con la tasa de trama. Aunque la voz se caracteriza por ser una fuente no estacionaria, la segmentación en tramas de corta duración permite llevar a cabo el análisis espectral a corto plazo de segmentos cuasi-estacionarios. Una vez obtenido el espectro, éste se transforma con el objetivo de obtener una representación paramétrica decorrelada y con una dimensionalidad reducida. Así pues, el resultado de la etapa de análisis es un vector de características $\mathbf{x}$ que representa los parámetros espectrales de cada trama. Adicionalmente, este vector se complementa con otro tipo de características, tales como la energía o parámetros dinámicos, obteniendo finalmente una secuencia de vectores de características $X = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)$.

El análisis espectral en reconocimiento de voz se lleva a cabo, principalmente, mediante dos métodos: *banco de filtros* y *predicción lineal*. El método de predicción lineal se ha utilizado clásicamente por varias razones. La primera de ellas es porque se basa en un potente modelo de producción de voz [3]. Además, cuando se aplica en condiciones de voz limpia (sin distorsiones) se consiguen resultados iguales o incluso superiores a los obtenidos con los métodos basados en bancos de filtros. A pesar de ésto, en los últimos años el método de predicción lineal ha sido reemplazado por los métodos basados en bancos de filtros ya que presentan un mejor comportamiento en presencia de ruido acústico [9]. En cualquier caso, la representación espectral obtenida se transforma al dominio cepstral [10]. Las siguientes subsecciones explican en detalle los procesos de análisis hasta ahora expuestos.

2.3.1. Preprocesado de la Señal de Voz

Normalmente, antes de iniciar el análisis de la señal de voz, ésta se somete a un filtrado de *preénfasis*, el cual consiste en filtrar la señal a través de un filtro del tipo [11],

$$P(z) = 1 - \mu z^{-1} \tag{2.1}$$

donde $\mu \leq 1$ es un factor real. Este filtro lleva a cabo una aproximación de la derivada de la señal, de modo que consigue eliminar la componente continua, además de realzar las componentes de alta frecuencia del espectro, las cuales tienen un factor de decaimiento de 6 dB/década característico de la señal de voz.

Después del preénfasis, la señal de voz se segmenta en tramas de corta duración, típicamente del orden de 20 a 40 ms (L muestras) considerando un cierto solapamiento. Así pues, dentro de una trama la señal se considera cuasi-estacionaria, de modo que los parámetros espectrales extraídos a partir de ésta pueden ser considerados constantes dentro de esa trama.

Justo antes de iniciar la fase de análisis espectral, la señal correspondiente a cada trama se enventana, normalmente utilizando una ventana de análisis diferente de la rectangular. Este proceso es aconsejable para reducir el fenómeno de *leakage* durante el análisis espectral [12]. Concretamente, es común utilizar la ventana de Hamming,

$$w(n) = 0{,}54 - 0{,}46\cos\left(2\pi\frac{n-1}{N}\right) \qquad 0 \leq n < N \tag{2.2}$$

ya que obtiene un buen compromiso entre resolución espectral y *leakage*.

2.3.2. Análisis de Predicción Lineal

La técnica LPC (*Linear Prediction Coding*) [13] establece un método para la extracción de la información espectral de segmentos de voz de corta duración. Este método se basa en un modelo que considera que la señal de voz $s(n)$ es la respuesta de un filtro todo-polos (el cual representa el tracto vocal) a una cierta excitación $e(n)$. La función de transferencia de este filtro es del tipo,

$$H(z) = \frac{\sigma}{1 - A(z)} \qquad A(z) = \sum_{k=1}^{p} a_k z^{-k} \tag{2.3}$$

donde los coeficientes a_k $(k = 1, \ldots, p)$ reciben el nombre de coeficientes LPC, mientras que σ representa a la ganancia del filtro y p es el orden del predictor. Partiendo de la expresión (2.3), podemos determinar la ecuación en diferencias,

$$s(n) = \sigma u(n) + \sum_{k=1}^{p} a_k s(n-k) \tag{2.4}$$

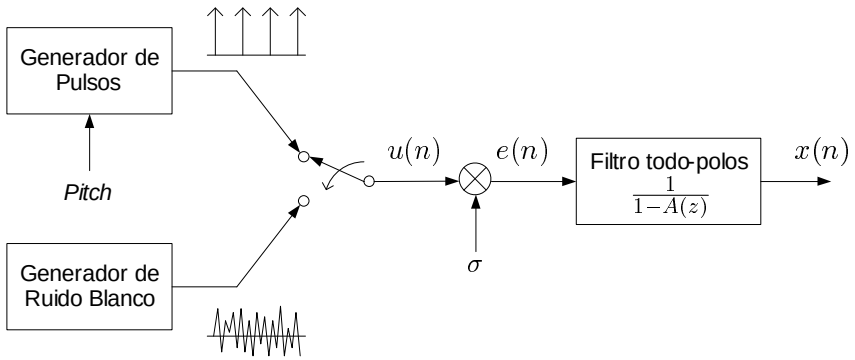


Figura 2.3: Modelo LPC de producción de voz.

Esta ecuación se puede interpretar como si estuviéramos representando la señal con un predictor lineal y un error de predicción $e(n) = \sigma u(n)$.

En la figura 2.3 presentamos el modelo LPC de producción de voz, donde podemos observar que la excitación se modela mediante dos posibles fuentes:

1. En el caso de que la señal de voz corresponda a un segmento sonoro, la excitación se modela mediante una secuencia de impulsos separados por un número constante de muestras. Esta separación viene determinada por el *periodo fundamental* o *pitch*, el cual corresponde a la inversa de la frecuencia de vibración de las cuerdas vocales.

2. Si el segmento considerado corresponde a un sonido sordo, la excitación se modela como un ruido estacionario blanco gausiano con media cero y varianza unitaria.

La extracción de los parámetros del modelo es similar a la extracción realizada para un modelo autoregresivo (modelo AR) o, equivalentemente, como los de un predictor lineal (para la obtención de los coeficientes LPC). Las principales vías de resolución son los métodos de covarianza y de autocorrelación. Ambos métodos se basan en la resolución de un sistema de ecuaciones lineales, y al igual que los algoritmos para la estimación del pitch, son ampliamente utilizados en el procesado de voz [14].

Puesto que $z = e^{j\omega}$, $H(\omega)$ establece la representación AR del espectro de la señal, la cual es conocida como *espectro LPC*. En la figura 2.4 se presenta la magnitud del espectro LPC de un segmento correspondiente al fonema/e/, así como el módulo de su transformada discreta de Fourier (DFT) o periodograma. El orden p del modelo LPC se escoge de modo que se puedan discernir claramente las frecuencias de resonancia del tracto vocal (también conocidas como *formantes*) y los valles del espectro. Normalmente, se escogen órdenes de predicción comprendidos entre 10 y 14 para modelar el comportamiento de la voz en los

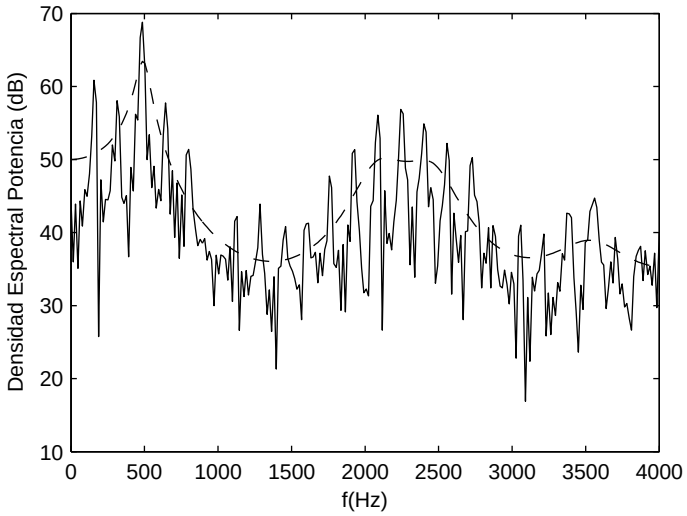


Figura 2.4: Representación espectral de un segmento de la vocal /e/. La línea continua representa el espectro DFT o periodograma, mientras que la línea discontinua se corresponde con el espectro LPC.

primeros 4 kHz. El espectro LPC mostrado en la figura 2.4 corresponde con $p = 10$. Como se puede observar, el módulo del espectro LPC proporciona una estima de la envolvente espectral, de modo que los coeficientes LPC constituyen una representación espectral de dimensión reducida.

2.3.3. Banco de Filtros en Escala mel

En la representación basada en banco de filtros la obtención de las características del espectro viene dada por las salidas de un conjunto de filtros paso-banda. Así pues, la salida de los filtros representa al espectro suavizado y diezmado, ya que cada filtro realiza un promedio ponderado de las componentes espectrales presentes en una cierta banda. Al igual que el análisis LPC, el análisis basado en banco de filtros se centra en la envolvente espectral y trata de caracterizar el filtro que modela el tracto vocal.

Como comentamos con anterioridad, actualmente los métodos de extracción de características suelen usar bancos de filtros ya que obtienen una parametrización más robusta frente a degradaciones acústicas [9]. Además, este método permite definir el ancho de banda de cada uno de los filtros del banco, de modo que se pueden tener en cuenta consideraciones perceptuales que recreen la forma en que la voz se percibe, es decir, mayor resolución para las bajas frecuencias que para las altas [15]. Debido a la anatomía del aparato auditivo humano, la respuesta subjetiva del oído no es uniforme con la frecuencia.

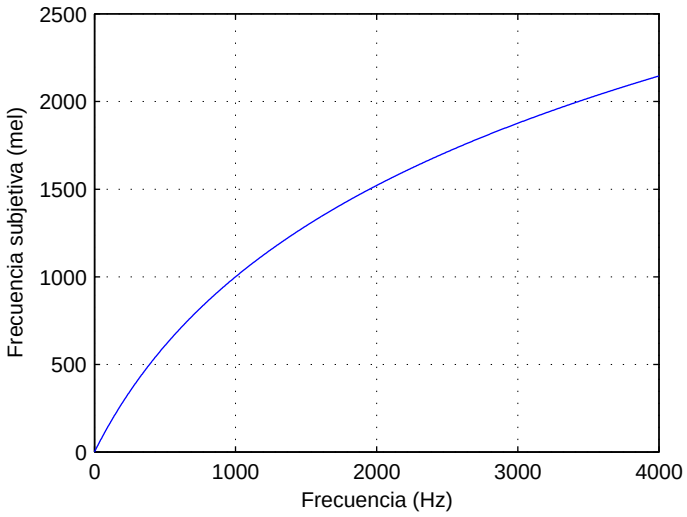


Figura 2.5: Escala mel.

El *mel* es una unidad de medida perceptual de la frecuencia que se obtiene por medio de experimentos psicoacústicos. A través de estos experimentos se obtiene una correspondencia entre la frecuencia ($f(Hz)$) y la escala mel (y_{mel}) o frecuencia subjetiva, cuya forma se puede aproximar a partir de diversas expresiones paramétricas [14, 16]. La figura 2.5 muestra la correspondencia definida por la siguiente expresión,

$$y_{mel} = 2595 \cdot \log_{10}\left(1 + \frac{f}{700}\right) \tag{2.5}$$

Partiendo de la escala mel, pues es la escala subjetiva percibida, las bandas de los filtros son distribuidas uniformemente. La figura 2.6 muestra un esquema de un banco de filtros triangulares en escala mel. Normalmente, se suelen seleccionar respuestas en frecuencia sencillas (como la triangular) para los distintos filtros, de modo que éstas se apliquen directamente sobre la DFT $|S(i)|$ $(i = 0, \ldots, N_{DFT})$ del segmento de señal considerado. Por tanto, las salidas del banco de filtros proporcionan la siguiente representación espectral:

$$B(k) = \sum_{i=\mathcal{I}(k)}^{\mathcal{F}(k)} |S(i)|^{\alpha} W_k(i) \qquad (k = 1, \ldots, F) \tag{2.6}$$

donde $B(k)$ y $W_k(i)$ son la salida y la respuesta en frecuencia del filtro k-ésimo, respectivamente, $\mathcal{I}(k)$ y $\mathcal{F}(k)$ corresponden a los índices de frecuencias que delimitan la banda de paso del filtro k-ésimo, y F es el número de filtros en el banco. El factor α determina si

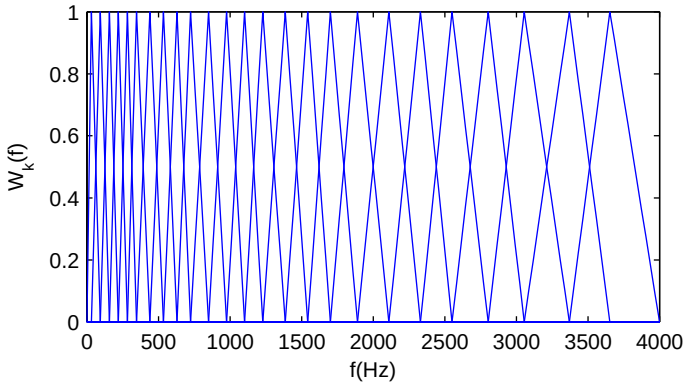


Figura 2.6: Esquema de un banco de filtros triangulares equidistribuidos en escala mel.

se considera la magnitud del espectro ($\alpha = 1$) o el espectro de potencia ($\alpha = 2$).

2.3.4. Coeficientes Cepstrales

Tras obtener la información espectral de la voz es necesario transformar esta información en un conjunto de parámetros compacto que sea idóneo para llevar a cabo el reconocimiento. En este sentido, normalmente se realiza la transformación de la información al dominio *cepstral* o del *cepstrum* [10]. Este nuevo dominio nos permite obtener una representación con un menor número de dimensiones donde la información de cada componente se encuentra más decorrelada, aportando así un evidente ahorro computacional en las etapas siguientes de reconocimiento. El cepstrum $c(n)$ se define como la función temporal obtenida de la transformada inversa del espectro logarítmico, o lo que es lo mismo,

$$\log S(\omega) = \sum_{n=-\infty}^{\infty} c(n) e^{j\omega n} \tag{2.7}$$

en donde $S(\omega)$ representa el espectro de la señal. Las muestras del cepstrum corresponden al dominio de la *cuefrencia* (aunque su transformación corresponda a un dominio temporal) y son normalmente conocidas como *coeficientes cepstrales*. Normalmente, las tareas de reconocimiento se realizan a partir de un conjunto reducido de los primeros coeficientes cepstrales. Además, es posible utilizar una ventana de ponderación de dichos coeficientes. Este enventanamiento recibe el nombre de *liftering*, y es equivalente a llevar a cabo un suavizado sobre el espectro, lo cual resulta útil a la hora de realizar comparaciones espectrales [14].

En función del análisis espectral que se realice se obtienen diferentes tipos de cepstrums. El modo más directo se obtiene a partir de la representación logarítmica del módulo de la DFT, el cual se somete a una DFT inversa para determinar su correspondiente cepstrum. En este caso, las componentes cepstrales reciben el nombre de LFCCs (*Linear Frequency Cepstrum Coefficients*) [9].

Cuando se utiliza el espectro LPC ($S(\omega) = H(\omega)$), los parámetros cepstrales obtenidos reciben el nombre de LPCCs (*Linear Prediction Cepstrum Coefficients*). Estos parámetros cepstrales presentan la ventaja de que pueden ser calculados por medio de las siguiente recursión [9],

$$c(n) = \begin{cases} \log \sigma & n = 0 \\ -a_{_1} & n = 1 \\ -a_n - \sum_{k=1}^{n-1} \frac{k}{n} c(k) a_{n-k} & 1 < n \leq L \end{cases} \tag{2.8}$$

También puede utilizarse la representación espectral basada en banco de filtros con escala mel, a partir de la cual se computan los parámetros MFCCs (*Mel Frequency Cepstral Coefficients*) [9]. Actualmente, esta es la representación más extendida y se obtiene de aplicar a las F salidas del banco de filtros en el dominio logarítmico, $\log B(k)$ ($k = 1, \ldots, F$), la transformada discreta del coseno (DCT, *Discrete Cosine Transform*), es decir,

$$c(n) = \sum_{k=1}^{F} \log B(k) \cos\left(\frac{\pi n}{F}(k - 0{,}5)\right) \qquad n = 0, \ldots, L \leq F \tag{2.9}$$

Una de las principales ventajas del cepstrum es que nos permite definir una sencilla medida de *distancia cepstral* entre dos espectros $S_{_1}(\omega)$ y $S_{_2}(\omega)$ como,

$$d_c(S_{_1}, S_{_2}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left|\log S_{_1}(\omega) - \log S_{_2}(\omega)\right|^2 d\omega = \sum_{n=-\infty}^{\infty} \big(c_{_1}(n) - c_{_2}(n)\big)^2 \tag{2.10}$$

Atendiendo al número de coeficientes cepstrales considerados, podemos aproximar esta distancia como una suma finita de los L primeros coeficientes cepstrales. Así, la expresión se reduce a considerar la distancia euclídea en el espacio vectorial $\mathbf{c} = (c(1), c(2), \ldots, c(L))$ del siguiente modo,

$$d_c(\mathbf{c}_{_1}, \mathbf{c}_{_2}) \approx \sum_{n=1}^{L} \big(c_{_1}(n) - c_{_2}(n)\big)^2 = \|\mathbf{c}_{_1} - \mathbf{c}_{_2}\|^2 \tag{2.11}$$

En la expresión anterior, la suma no tiene en cuenta el coeficiente cepstral de orden 0. Este coeficiente está relacionado con la energía y, por tanto, el hecho de no tenerlo en cuenta en la expresión (2.11) se traduce en la comparación de los espectros S_1 y S_2 normalizados en potencia. De esta forma, la anterior expresión establece una medida de similitud entre las formas espectrales.

2.3.5. Otras Características

Normalmente, el vector de componentes cepstrales **c** se complementa mediante la inclusión de la energía en escala logarítmica, también denominado *log-energía* [17],

$$\log E = 10 \log_{10} \sum_{n=0}^{N-1} s^2(n) \tag{2.12}$$

De este modo, el vector formado por los coeficientes cepstrales y la energía recibe el nombre de *vector de características estático*, puesto que sólo ofrece información de una cierta trama.

Bajo el supuesto de que las transiciones espectrales juegan un papel clave en la percepción humana de la voz, Furui [18] propuso el empleo de ciertas características dinámicas que incluyeran la información correspondiente a las transiciones de la voz. Para ello, cada característica instantánea $c(k)$ es considerada una función del tiempo $c_t(k)$. Así, considerando un entorno de $2M$ características centrado en la correspondiente al instante de tiempo t podemos obtener el correspondiente coeficiente dinámico como,

$$\Delta c_t(k) = \frac{\sum_{m=-M}^{M} w_m c_{t+m}(k)}{\sum_{m=-M}^{M} w_m^2} \qquad k = 1, \ldots, L \tag{2.13}$$

donde $w_m = m$. El vector $\Delta \mathbf{c} = (\Delta c_t(1), \Delta c_t(2), \ldots, \Delta c_t(L))$ es conocido como *delta cepstrum* o *vector de velocidad*, y representa la derivada del vector estático respecto al tiempo. De igual manera, las segundas derivadas (*componentes delta-delta* o *de aceleración*) permiten realzar el rendimiento de los sistemas de reconocimiento [18, 19]. Su cómputo se puede llevar a cabo aplicando la expresión (2.13) sobre las componentes delta o de velocidad.

2.4. Aproximaciones al Problema RAH

Una vez determinada una parametrización reducida de la señal de voz resta por llevar a cabo la tarea de reconocimiento en sí. A este efecto, tres han sido las propuestas más extendidas (en orden de aparición): reconocimiento de patrones, modelado estadístico y redes neuronales. Actualmente, las soluciones más utilizadas corresponden a las dos primeras. Concretamente, el reconocimiento de patrones utilizado en las décadas de los 60 y 70 fue siendo reemplazado por la aproximación estadística, basada en modelos ocultos de Markov, durante la década de los 80. A continuación expondremos brevemente las bases de estas dos técnicas, para posteriormente centrarnos en la técnica de reconocimiento más utilizada en la actualidad, es decir, la aproximación estadística basada en modelos ocultos de Markov.

2.4.1. Reconocimiento de Patrones

Partiendo del análisis espectral de la señal de voz que se realizó en la sección anterior, vimos como la voz finalmente es representada mediante una secuencia de vectores de características. De este modo, extraemos una secuencia de prueba, $\mathcal{T}$, dada por la concatenación de los vectores de características extraídos a partir de la voz a reconocer,

$$\mathcal{T} = \{\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_I\} \tag{2.14}$$

donde $\mathbf{t}_i$ representa el vector de características de entrada correspondiente a la trama i, siendo I el número total de tramas de voz. Igualmente, podemos definir un conjunto de patrones de referencia $\{\mathcal{R}^1, \mathcal{R}^2, \ldots, \mathcal{R}^V\}$, donde cada patrón $\mathcal{R}^j$ surge también de la concatenación de un conjunto de vectores de características,

$$\mathcal{R}^j = \{\mathbf{r}_1^j, \mathbf{r}_2^j, \ldots, \mathbf{r}_J^j\} \tag{2.15}$$

De este modo, la solución basada en el reconocimiento de patrones se reduce a computar una cierta medida de distancia o similitud entre el patrón $\mathcal{T}$ y cada uno de los patrones de referencia $\mathcal{R}^j$ $(1 \leq j \leq V)$, con el fin de identificar aquel patrón que minimiza la distancia y que, por tanto, se asociará a la voz de entrada. No obstante, es necesario establecer una medida de distancia que nos permita comparar patrones con diferentes longitudes. Así, para la resolución de este problema es necesario llevar a cabo la alineación entre los patrones de entrada y de referencia. Este alineamiento puede ser conseguido aplicando

técnicas de *programación dinámica* [20], las cuales reciben el nombre de técnicas DTW (*Dynamic Time Warping*) cuando son aplicadas al reconocimiento de voz [21]. Esta herramienta nos permite no sólo comparar patrones de diferentes longitudes, sino también obtener los patrones de referencia durante la fase de entrenamiento.

2.4.2. Aproximación Estadística

El modelado estadístico es la solución más extendida ya que permite establecer el problema de reconocimiento en términos estadísticos. Como punto de partida establecemos que $\mathcal{W} = \{W_i\}$ representa el conjunto de posibles frases dadas para un determinado lenguaje, mientras que X representa el conjunto de vectores de características correspondientes a la locución a reconocer. Por tanto, el problema de reconocimiento consiste en determinar la frase $W(X)$ correspondiente a la secuencia observada X. Aplicando la regla de decisión basada en el *máximo a posteriori* (MAP), la unidad reconocida puede expresarse como,

$$W(X) = \underset{j}{\operatorname{argmax}}\, P(W_j|X) \tag{2.16}$$

Esta maximización requiere el cómputo de las probabilidades condicionales $P(W|X)$ que, mediante la regla de Bayes, pueden ser calculadas como,

$$P(W|X) = \frac{P(X|W)P(W)}{P(X)} \tag{2.17}$$

Puesto que el problema de maximización expuesto en (2.16) es independiente de $P(X)$, podemos reescribir el planteamiento MAP como,

$$W(X) = \underset{j}{\operatorname{argmax}}\, P(X|W_j)P(W_j) \tag{2.18}$$

En la solución alcanzada, $P(X|W_j)$ es proporcionada por el modelo acústico de una cierta frase W_j, mientras que $P(W_j)$ representa la probabilidad *a priori* de esa unidad y es proporcionada por un modelo de lenguaje. En la siguiente sección introducimos los modelos ocultos de Markov como principal herramienta del modelado acústico.

2.5. Reconocimiento de Voz mediante Modelos Ocultos de Markov

La base teórica de los modelos ocultos de Markov (HMM, *Hidden Markov Model*) fue desarrollada en los años 60. Sin embargo no fueron aplicados al reconocimiento de voz hasta los 70, convirtiéndose en la solución más utilizada por la comunidad científica en los años 80 [22, 23, 24]. Actualmente, la aproximación estadística utilizando HMMs para el modelado acústico constituye la solución más extendida al problema de reconocimiento. De ahí que, dada su importancia, dediquemos esta sección a establecer las bases de los modelos ocultos de Markov y su aplicación al reconocimiento de voz.

2.5.1. Formulación de los HMMs

Un *proceso* o *cadena de Markov* viene definido por un conjunto de estados y un conjunto de probabilidades de transición desde un estado a otro. Los procesos de Markov, en los cuales el estado actual depende de los estados previos, nos permiten modelar un proceso aleatorio con memoria. Los procesos ocultos de Markov son una generalización de los procesos de Markov. Concretamente, los modelos HMM introducen adicionalmente un nuevo proceso estadístico sobre el proceso de Markov. De este modo, un HMM es un modelo estadístico que describe la producción de una secuencia de observaciones $O = (o_1, o_2, \ldots, o_T)$, generada por una secuencia "oculta" de estados $Q = (q_1, q_2, \ldots, q_T)$. Así, tenemos un proceso oculto que viene dado por la secuencia de estados y otro observable que lo constituye la secuencia de observaciones, es decir, la superposición de dos procesos estocásticos.

Podemos definir un modelo HMM λ mediante los siguientes elementos:

1. Un conjunto S con N estados, $S = \{s_1, s_2, \ldots, s_N\}$ interconectados entre sí. En cada momento t el modelo se encuentra en un cierto estado que denotaremos como q_t.

2. Una matriz de transición $A = \{a_{ij}\}$ que contiene las probabilidades de transición entre estados, las cuales vienen definidas del siguiente modo,

$$a_{ij} = P(q_{t+1} = s_j | q_t = s_i) \qquad i, j = 1, \ldots, N \tag{2.19}$$

que verifican la siguiente condición,

$$\sum_{j=1}^{N} a_{ij} = 1 \tag{2.20}$$

3. Dependiendo de la naturaleza del conjunto de observaciones tendremos un modelo HMM continuo o discreto. Cuando las observaciones, o_t, pertenecen a un conjunto V dado por M posibles símbolos, $V = \{v_1, v_2, \ldots, v_M\}$, estaremos ante un HMM discreto (DHMM). En este caso, la probabilidad de observación viene dada por una matriz $\mathbf{B} = \{b_i(v_k)\}$, en la que cada elemento representa la probabilidad de generación de un cierto símbolo en un cierto estado,

$$\begin{aligned} b_i(v_k) = P(o_t = v_k | q_t = s_i) \qquad & i = 1, \ldots, N \\ & k = 1, \ldots, M \end{aligned} \tag{2.21}$$

Estas probabilidades deben verificar que,

$$\sum_{k=1}^{M} b_i(v_k) = 1 \qquad i = 1, \ldots, N \tag{2.22}$$

Por contra, si la variable observada $o_t = \mathbf{x}$ pertenece a un espacio continuo de p dimensiones, $\mathbf{x} \in \mathbb{R}^p$, tendremos un HMM continuo (CHMM). En este caso las probabilidades de salida $b_i(v_k)$ deben de ser sustituidas por funciones densidad de probabilidad (pdf, *probability density function*) $b_i(\mathbf{x}) = p(o_t = \mathbf{x} | q_t = s_i)$, las cuales deben de cumplir la siguiente condición de normalización,

$$\int_{\mathbb{R}^p} b_i(\mathbf{x}) d\mathbf{x} = 1 \qquad i = 1, \ldots N \tag{2.23}$$

4. Una matriz $\Pi = \{\pi_i\}$ en la que cada elemento determina la probabilidad de que un cierto estado sea el estado inicial del modelo,

$$\pi_i = P(q_1 = s_i) \qquad i = 1, \ldots, N \tag{2.24}$$

Estas probabilidades deben verificar la siguiente condición,

$$\sum_{i=1}^{N} \pi_i = 1 \tag{2.25}$$

El modelado DHMM parte de la premisa de que las observaciones pertenecen a un conjunto discreto V. No obstante, los vectores de características extraídos de la voz no se encuentran discretizados. Por tanto, para poder utilizar el modelado discreto sería necesario llevar a cabo una cuantización previa de las observaciones, con la consabida pérdida de información.

Por contra, el modelado CHMM no exige la cuantización de las observaciones, aunque es necesario llevar a cabo un correcto modelado de las distribuciones de probabilidad de observación. Normalmente, cuando consideramos un modelo CHMM la distribución de probabilidad de las observaciones suele aproximarse mediante una mezcla de pdfs del tipo,

$$b_i(\mathbf{x}) = p(o_t = \mathbf{x}|q_t = s_i) = \sum_{v_k \in V(s_i)} c_{ik} p(\mathbf{x}|v_k, s_i) \tag{2.26}$$

donde cada pdf $p(\mathbf{x}|v_k, s_i)$ es etiquetada mediante un índice v_k que varía en un conjunto $V(s_i)$ específico para el estado s_i, mientras que $c_{ik} = P(v_k|s_i)$ corresponde al peso de cada pdf en la mezcla (la suma de los pesos debe ser 1). Frecuentemente, la probabilidad de observación se modela mediante una mezcla de gausianas multivariadas [25] obteniendo la siguiente expresión,

$$b_i(\mathbf{x}) = p(o_t = \mathbf{x}|q_t = s_i) = \sum_{k=1}^{M} c_{ik} \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik}) \tag{2.27}$$

donde $\boldsymbol{\mu}_{ik}$ es el vector media y $\boldsymbol{\Sigma}_{ik}$ es la matriz de covarianza correspondientes a la k-ésima gausiana multivariada del estado s_i.

2.5.2. Aplicación de los HMMs al Reconocimiento de Voz

La principal aplicación de los HMMs en el reconocimiento de voz es el modelado acústico de las unidades de reconocimiento, donde cada unidad de reconocimiento diferente es representada mediante un HMM. En la figura 2.7 se muestra un ejemplo sencillo de un sistema de reconocimiento de palabras aisladas. En el ejemplo, las distintas palabras W_i que forman el vocabulario actúan como unidades de reconocimiento λ_i, de modo que

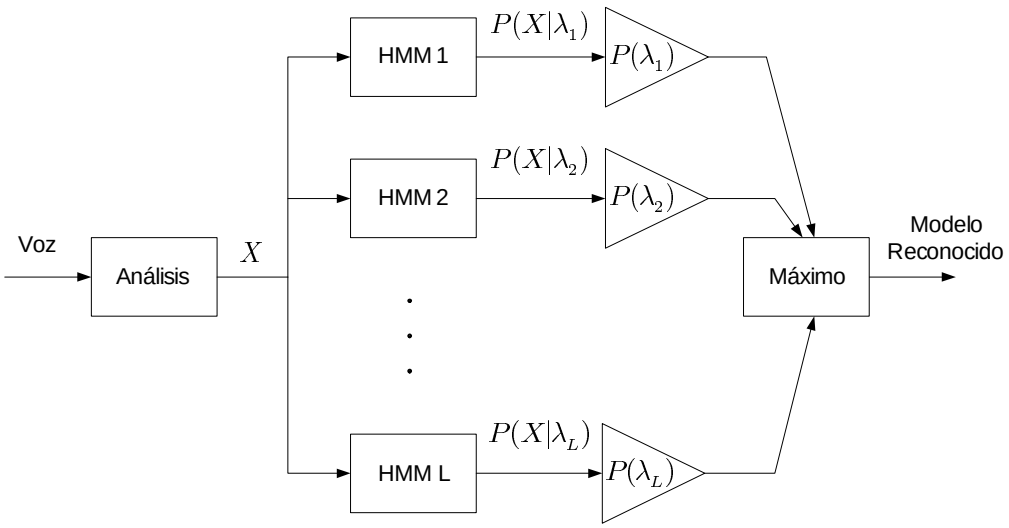


Figura 2.7: Reconocimiento de palabras aisladas basado en el uso de HMMs.

se asocia un HMM a cada una de ellas. Así, el reconocimiento de una cierta secuencia de características X equivale a obtener aquel modelo λ_i que maximiza la probabilidad conjunta $P(X, \lambda_i)$. Esta probabilidad podemos expresarla, aplicando el teorema de Bayes, como $P(X, \lambda_i) = P(X|\lambda_i)P(\lambda_i)$, es decir, que la probabilidad conjunta vendrá dada por la probabilidad condicional de que dado un modelo λ_i se produzca la secuencia observada X, $P(X|\lambda_i)$, ponderada ésta a su vez por la probabilidad a priori de ese modelo, $P(\lambda_i)$.

La topología más general de un modelo HMM es la ergódica, la cual asegura la existencia de una transición desde un estado a otro cualquiera. No obstante, la topología más extendida en el reconocimiento de voz es la de *izquierda a derecha* o *de Bakis* [26], mostrada en la figura 2.8, ya que se adapta perfectamente a la naturaleza secuencial de la voz en la que los estados consecutivos representan la variación dinámica de una cierta locución. En este modelo se incluyen dos estados nulos (denominados I y F), los cuales no emiten observación alguna ni consumen ninguna unidad temporal, pero que sirven para modelar el principio y fin de una secuencia, así como su concatenación con otras en sistemas de reconocimiento de habla continua. 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 (se mantiene en el estado actual) hasta un valor máximo Δ (el ejemplo de la figura 2.8 corresponde a $\Delta = 2$). Cuando las unidades de reconocimiento son fonemas o trifonemas independientes del contexto se suele utilizar este tipo de topología con tres estados no nulos que representan una transición inicial, una parte estacionaria y una transición final.

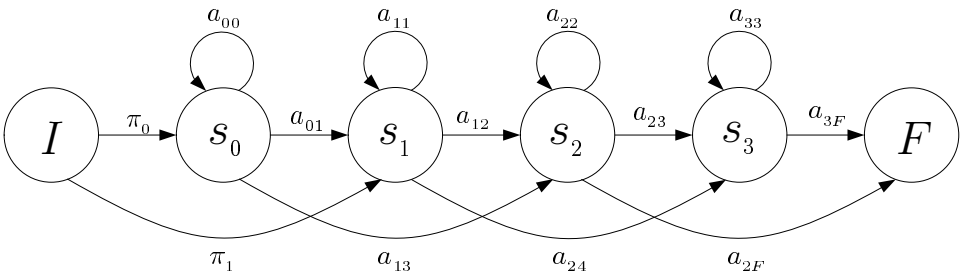


Figura 2.8: Topología de izquierda a derecha.

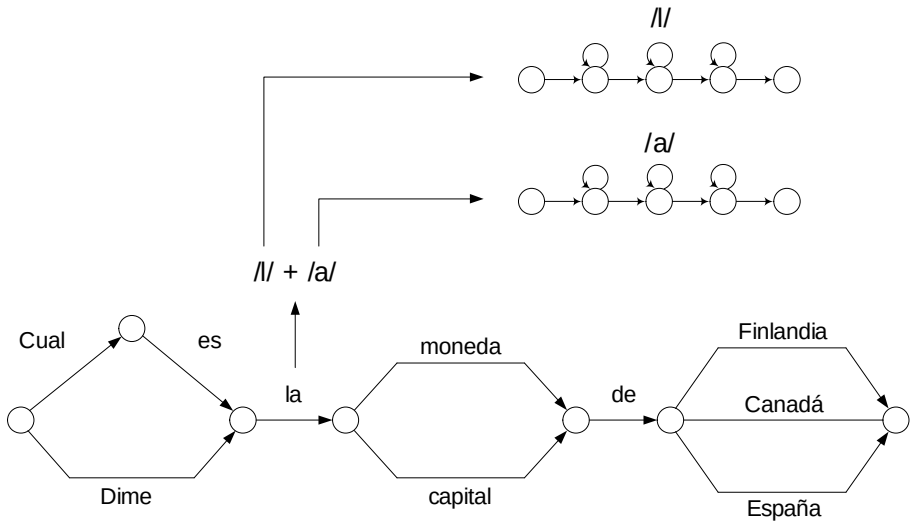


Figura 2.9: Sistema de reconocimiento de habla continua basado en un macromodelo de HMMs.

En principio, el desarrollo de un sistema de reconocimiento de habla continua se podría llevar a cabo asignando un modelo HMM a cada frase. Sin embargo, el número de posibles frases es tan grande que esta solución es inviable. En la práctica, se recurre a la construcción de un macromodelo que incluye todas las frases de una cierta gramática. La figura 2.9 muestra un ejemplo de una sencilla gramática recogida por un macromodelo que utiliza una topología de Bakis para el modelado de fonemas. El macromodelo se obtiene al concatenar los correspondientes modelos de los fonemas (atendiendo a la gramática) a través de los estados nulos. La unidad básica del macromodelo no tiene por qué ser el fonema, sino que podríamos recurrir a una unidad superior como el trifonema o la palabra. En este caso, el reconocimiento de una cierta frase consistirá en determinar aquel camino Q más probable a lo largo de todo el macromodelo, es decir, el problema de reconocimiento consiste en determinar qué frase maximiza la probabilidad $P(Q|X, \lambda)$.

Así, el reconocimiento del habla mediante HMMs requiere la resolución de tres problemas básicos [4, 27]:

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

2. Problema de *decodificación*. Dado el modelo λ y el conjunto de observaciones X, consiste en hallar la secuencia de estados $Q = (q_1, q_2, \ldots, q_T)$ más probable, o lo que es lo mismo determinar el siguiente camino óptimo,

$$Q^* = \underset{Q}{\operatorname{argmax}}\, P(Q|X, \lambda) \tag{2.28}$$

3. Problema de *estimación*. En este caso, consiste en determinar el conjunto de parámetros de cada HMM que mejor se ajusta a un conjunto de observaciones acústicas. Concretamente, consiste en determinar las probabilidades de transición dadas por la matriz (2.19), las probabilidades a priori establecidas en la ecuación (2.24) y, dependiendo de si el modelo considerado es discreto o continuo, las probabilidades de observación definidas por las ecuaciones (2.21) ó (2.26), respectivamente.

El primer problema es el problema más básico de reconocimiento, el cual nos permite llevar a cabo el reconocimiento de palabras aisladas, y que es posible resolver mediante el algoritmo adelante-atrás (o *algoritmo forward-backward*) [28]. El segundo problema consiste en obtener la secuencia de estados oculta más probable, el cual coincide exactamente con el objetivo del reconocimiento continuo del habla donde teníamos que obtener el camino óptimo sobre el macromodelo. El *algoritmo de Viterbi* nos permite solucionar este problema de una forma eficiente [29]. Finalmente, el problema de *estimación* corresponde con el entrenamiento de los HMMs, es decir, con la obtención de los parámetros de los HMMs a partir de la base de datos de entrenamiento acústica y sus transcripciones. La resolución de este problema se consigue mediante el algoritmo de *Baum-Welch* [30, 31].

Capítulo 3

Reconocimiento Remoto y Codificación de Voz

3.1. Introducción

El rápido desarrollo de diversas redes inalámbricas, tales como 3G, redes WiFi o Bluetooth, ha propiciado el hecho de que la conectividad de los dispositivos móviles se convierta en una característica intrínseca de éstos. No obstante, los terminales tienden a reducir sus dimensiones con el objetivo de aumentar su portabilidad, lo que reduce la accesibilidad de sus interfaces. En este sentido, esta tendencia imposibilita el desarrollo de nuevos servicios sobre redes inalámbricas ya que se hace más difícil la interacción con el usuario. Así pues, se hace necesario el desarrollo de nuevos interfaces de usuario que provean de una fácil interacción multimodal para la próxima generación de dispositivos móviles.

En este escenario, el reconocimiento automático del habla es un camino prometedor para un acceso fácil y natural a las nuevas aplicaciones de red. Sin embargo, los terminales móviles se caracterizan por tener una capacidad de cómputo restringida, así como una corta duración de batería. El reconocimiento remoto de voz (RSR, *Remote Speech Recognition*) permite salvar estas restricciones *hardware* ubicando las tareas de mayor coste computacional del reconocimiento en un servidor remoto. De este modo se eliminan las restricciones de tamaño y consumo, habilitando el uso de potentes computadores que contengan sistemas de interacción oral sofisticados, los cuales incluyen sistemas de diálogo, módulos de lenguaje natural, síntesis de voz y acceso a bases de datos. En este paradigma los terminales actúan como clientes activados por voz que realizan peticiones de reconocimiento a un servidor centralizado, el cual controla un cluster compartido de

servidores de reconocimiento [32].

En este capítulo presentaremos las arquitecturas disponibles para el reconocimiento de voz en terminales móviles, haciendo especial hincapié en las arquitecturas de reconocimiento remoto. Dentro de estas últimas existen dos posibles implementaciones, arquitecturas sólo-servidor y cliente-servidor, de las que se expondrán sus ventajas e inconvenientes, justificando la selección de la arquitectura sólo-servidor para el desarrollo de este trabajo. Puesto que el rendimiento de esta arquitectura se encuentra supeditado al rendimiento del esquema de codificación de voz seleccionado, el capítulo se finalizará presentando las principales técnicas de codificación de voz existentes.

3.2. Arquitecturas de Reconocimiento

La arquitectura de un sistema de reconocimiento del habla, tal y como vimos en la sección 2.2, se compone principalmente de tres bloques: un bloque de adquisicióndedicado a la captura de la señal de voz, un bloque de análisis y un bloque final de reconocimiento. El bloque de análisis se encarga de extraer una serie de vectores de características, a partir de la señal de voz adquirida, que constituyen la entrada del bloque de reconocimiento. Este último se encarga de determinar la secuencia de palabras más probable confrontando los vectores de características con los modelos acústicos y de lenguaje de la tarea de reconocimiento.

Puesto que los componentes del sistema de reconocimiento realizan funciones independientes, la partición del sistema de reconocimiento posibilita diferentes arquitecturas de reconocimiento. Si entendemos el sistema de reconocimiento como un interfaz que nos da acceso a algún tipo de servicio, dependiendo de dónde situemos la red de comunicaciones en la cadena de procesamiento y reconocimiento de voz, podemos distinguir tres posibles arquitecturas [33]: sistema de reconocimiento empotrado (ESR, *Embedded Speech Recognition*), arquitectura sólo-servidor (NSR, *Network-based Speech Recognition*) y arquitectura cliente-servidor (DSR, *Distributed Speech Recognition*), tal y como se muestra al pie de la figura 3.1. En todas ellas, la captura de la voz se realiza en el terminal cliente, mientras que la aplicación podría residir en el cliente o en el servidor. La primera de las arquitecturas, ESR, integra el sistema de reconocimiento al completo en el cliente. Sin embargo, como comentamos con anterioridad, los clientes tienen unos recursos y capacidades de cómputo limitadas, lo que hace que este tipo de arquitectura sólo sea adecuada para aplicaciones sencillas con un bajo nivel de complejidad (reconocimiento de dígitos, palabras clave,

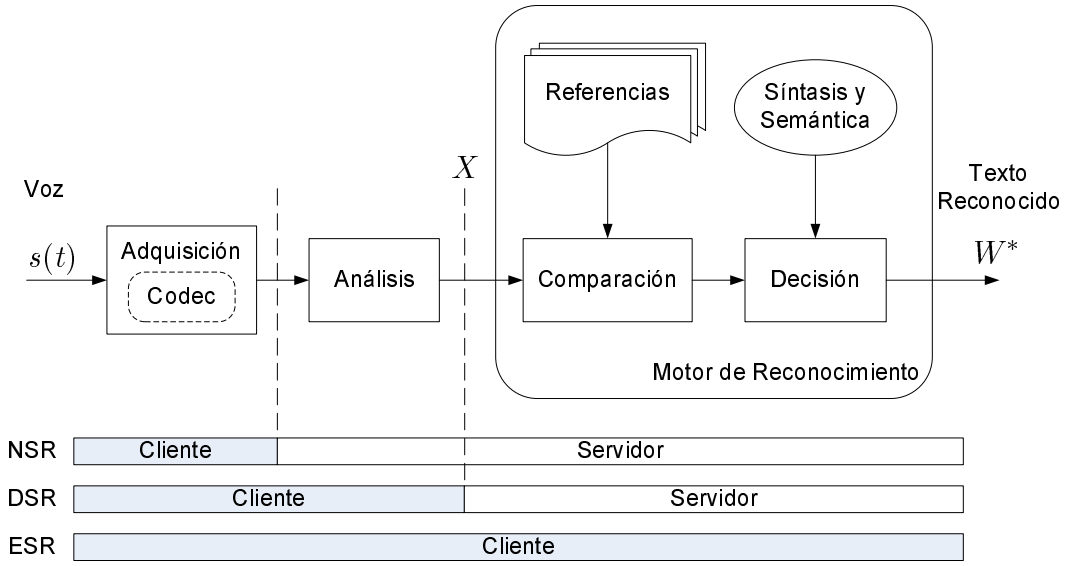


Figura 3.1: Posibles arquitecturas de un sistema de reconocimiento basado en el paradigma cliente-servidor.

etc...). Por contra, las arquitecturas NSR y DSR posibilitan la construcción de un sistema RSR donde la carga de cómputo del sistema de reconocimiento sería demasiado grande para el terminal móvil. Además, los sistemas RSR ofrecen algunas oportunidades que los sistemas ESR no pueden ofrecer. Piénsese, por ejemplo, en un sistema de reconocimiento de voz semi-automático asistido por operadores humanos.

3.2.1. Arquitectura Remota DSR

En la arquitectura DSR, o sólo-servidor, las tareas de procesamiento y computación se encuentran distribuidas entre el terminal y el servidor remoto de reconocimiento de voz. En este tipo de arquitectura se evita transmitir directamente la señal de voz ya que ésta posee cierta información redundante (características intrínsecas del locutor) para la tarea de reconocimiento. En su lugar el dispositivo cliente lleva a cabo la adquisición de voz y las tareas propias del bloque de análisis, es decir, la extracción de los vectores de características. Estas características, y no la voz, son codificadas y transmitidas a través de la red. De esta forma, se evita la posible degradación de la voz en el proceso de codificación/decodificación, a la vez que se reduce la cantidad de información a transmitir. En el receptor los vectores de características son decodificados e introducidos en el núcleo reconocedor. Se distingue así entre un *front-end* que realiza la adquisición y extracción

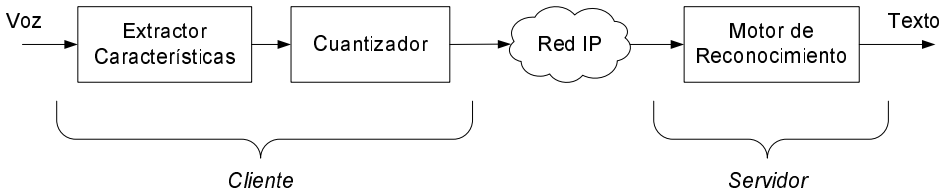


Figura 3.2: Esquema de reconocimiento remoto DSR.

de características, y un *back-end* que lleva a cabo el proceso de reconocimiento en sí. La figura 3.2 muestra un diagrama de bloques de la arquitectura de reconocimiento remoto DSR.

El principal inconveniente de esta arquitectura reside en que precisa de un front-end normalizado común para todas las aplicaciones [34], de modo que todos los terminales cliente lo integren. A este respecto, el Aurora DSR Working Group desarrolló un conjunto de front-ends que permiten el reconocimiento distribuido en redes de telefonía móvil GSM (*Global System for Mobile communications*) y que se encuentran estandarizados por el organismo ETSI (*European Telecommunications Standards Institute*). El primer estándar fue el *Front-End* básico (FE) [35], el cual obtiene muy buenos resultados en entornos acústicamente limpios [36]; posteriormente, se propuso el *Advanced Front-End* (AFE) [37], que mejora las prestaciones del FE en entornos acústicos adversos [38]; el *eXtended Front-End* (XFE)[39], el cual introduce ciertos parámetros sobre el FE que permiten el reconocimiento de lenguajes tonales y la síntesis de voz; y el *eXtended Advanced Front-End* (XAFE) [40], que es una versión del XFE robusta al ruido acústico (combinación de AFE y XFE).

Posteriormente a la aparición de los estándares, éstos se extendieron a redes IP mediante la definición del formato de carga útil de DSR para el protocolo RTP (*Real Time Protocol*) [41]. El protocolo RTP [42] proporciona la transmisión de datos extremo a extremo entre aplicaciones que requieren transmisión en tiempo real, el cual se apoya sobre un servicio no orientado a la conexión de tipo datagrama ofrecido por los protocolos UDP (*User Datagram Protocol*) e IP (*Internet Protocol*). Además, los estándares DSR incluyen los algoritmos necesarios de mitigación de errores o pérdidas.

3.2.2. Arquitectura Remota NSR

La arquitectura NSR, o de sólo-servidor, se caracteriza por sólo llevar a cabo el proceso de adquisición en el terminal cliente. Por este motivo, dentro del bloque de adquisición de la figura 3.1 se muestra la posibilidad de que éste venga dado por un códec de voz. El

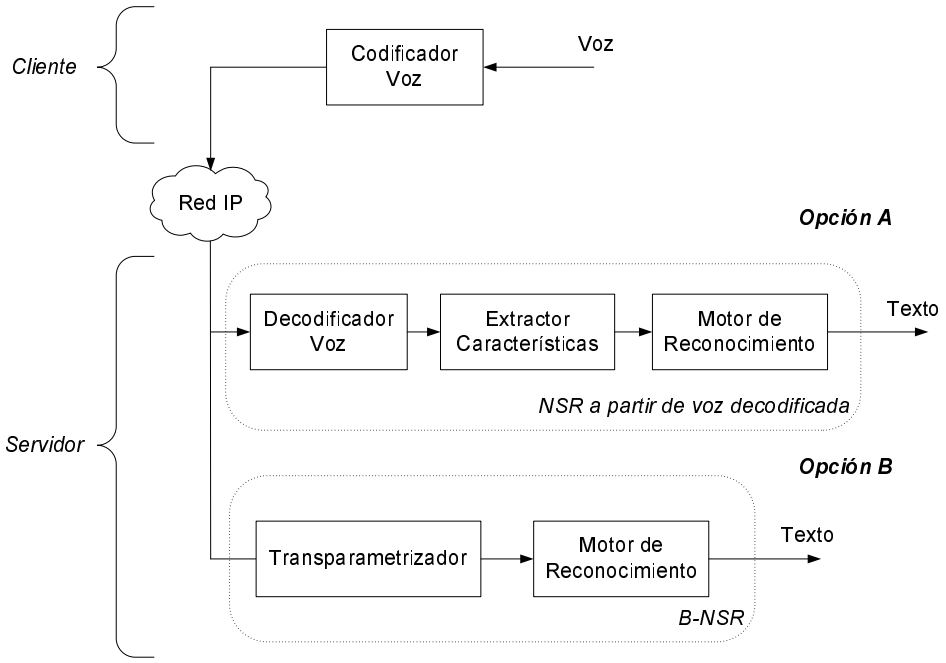


Figura 3.3: Implementaciones de un sistema de reconocimiento remoto NSR.

resto del sistema de reconocimiento se integra en el lado servidor, de forma que, desde el punto de vista del usuario, es la red la que se encarga de realizar todo el proceso de reconocimiento.

La figura 3.3 muestra las posibles implementaciones de un sistema de reconocimiento remoto NSR. En este caso, el cliente utiliza un codificador de voz convencional, lo que permite la transmisión a una baja tasa de bits. En función de cómo se realiza la etapa de análisis en el lado servidor podemos distinguir dos posibles implementaciones [43]:

- NSR básico, o a partir de voz decodificada. Este es el modo más directo para llevar a cabo el reconocimiento y consiste en decodificar la señal de voz y extraer a partir de ésta las características de reconocimiento (opción A en la figura 3.3).

- B-NSR (*Bitstream-based NSR*). En este caso la extracción de las características de reconocimiento se lleva a cabo directamente a partir de los parámetros de codificación, no siendo necesaria la síntesis de voz (opción B en la figura 3.3).

La implementación B-NSR puede ser vista como una *transparametrización*, o *transcodificación*, si tenemos en cuenta que la señal de voz podría reconstruirse a partir de las características de reconocimiento (tal y como hacen los estándares XFE y XAFE). De

hecho, algunos autores han propuesto sistemas de codificación de voz basados en parámetros cepstrales [44, 45], los cuales consiguen mejorar las prestaciones del reconocimiento remoto manteniendo una calidad perceptual adecuada en la señal reconstruida. Existen varias razones por las que un sistema B-NSR puede resultar ventajoso frente a un sistema de reconocimiento NSR básico:

- Los codificadores de voz están diseñados con el objetivo de proporcionar la mayor calidad perceptual posible. Durante el proceso de codificación es común incluir algún tipo de post-procesado en el decodificador, de modo que se obtenga una señal decodificada perceptualmente realzada. No obstante, este postprocesado no es óptimo desde el punto de vista del reconocimiento de voz [46].

- No es necesario reconstruir la señal de voz, lo que supone un ahorro computacional.

- En caso de que se produzcan degradaciones introducidas por el canal, se puede aplicar un algoritmo de mitigación apropiado para el reconocimiento de voz directamente sobre los parámetros del codificador.

No obstante, la implementación B-NSR puede tener también ciertos inconvenientes ya que el proceso de extracción de características es dependiente del codificador utilizado, además de que la tasa de envío de parámetros del codificador y la tasa de extracción de características no tienen por qué ser iguales.

3.2.3. NSR vs. DSR

La arquitectura DSR ha recibido más atención por parte de la comunidad científica durante los últimos años. La principal ventaja de esta arquitectura es que está diseñada, desde su inicio, con el objetivo del reconocimiento de voz. En contraste con NSR, DSR extrae directamente las características de reconocimiento a partir de la señal de voz original lo que conlleva una serie de ventajas cuando es aplicado en redes IP [43]:

1. No existe pérdida de rendimiento debida a la compresión de voz (proceso de codificación/decodificación). La cuantización de los parámetros de reconocimiento no introduce degradaciones apreciables [47].

2. La tasa de bits suele ser menor debido a que se eliminan aquellas redundancias que no son precisas en reconocimiento (información referente a la identidad del locutor).

3. En entornos móviles DSR se presenta como una opción más robusta ya que el extractor de características puede llevar a cabo algún tipo de procesado destinado a la compensación de ruido acústico.

4. En principio, los sistemas DSR son más robustos ante degradaciones del canal ya que las técnicas de mitigación de errores que integran están destinadas a optimizar el rendimiento de las tareas de reconocimiento.

Aunque como hemos visto las ventajas de DSR sobre NSR son obvias, es necesario hacer notar que NSR ofrece la posibilidad de ciertos servicios difíciles de implementar, o al menos no de una forma natural, en una arquitectura DSR. NSR se puede entender como un servicio de valor añadido sobre VoIP, de modo que se transmite la señal de voz pero no con el objetivo de establecer una conferencia, sino de suministrar algún tipo de servicio automático remoto. En este sentido, el hecho de transmitir la señal de voz posibilita la aparición de nuevos servicios. Aunque algunos estándares de DSR permiten llevar a cabo una reconstrucción de la voz, la arquitectura NSR consigue esta tarea con mucha más naturalidad, lo que permite llevar a cabo tareas de identificación de locutor. Otro posible servicio es el de asistencia a operadores, en el que el usuario establece una comunicación con un centro de atención de llamadas atendido por teleoperadores, los cuales, a su vez, reciben asistencia sobre las consultas del usuario mediante un sistema automático de reconocimiento.

Además, hay que resaltar que en la actualidad, los sistemas RSR, implementados para redes fijas y móviles, están basados en la arquitectura NSR. Esto se debe a que esta arquitectura, al utilizar un esquema convencional de codificación de voz, no precisa introducir ningún cambio en los terminales. La escasa aceptación por parte de la industria de los estándares DSR sobre las redes GSM hacen pronosticar que los futuros sistemas RSR basados en redes IP adoptarán una arquitectura NSR, ya que se encuentra soportada por las cada vez más extendidas plataformas VoIP. Esta es la principal razón que motivó el desarrollo de este trabajo, que tiene como objetivo el desarrollo de técnicas que mejoren el rendimiento de la arquitectura NSR sobre redes IP. El rendimiento de la arquitectura NSR está intrínsecamente ligado al comportamiento del esquema de codificación de voz seleccionado. Por este motivo, el resto del capítulo tiene como objetivo presentar los principales fundamentos de los codificadores utilizados en la transmisión de voz sobre redes IP.

3.3. Codificación de Voz

El objetivo de la codificación de voz es obtener una representación de la señal con el menor número de bits posibles, pero que permita la reconstrucción de ésta con un cierto nivel de calidad de modo que se pueda realizar su transmisión o almacenamiento de una forma eficiente [48]. El codificador se encarga de realizar las operaciones correspondientes de *análisis* sobre la voz original, mientras que el decodificador realiza la *síntesis* cuando extrae la señal de voz a partir de la secuencia de bits recibidos. No obstante, ambos procesos no son totalmente inversos ya que parte de la información se pierde, o bien por razones intrínsecas del propio proceso de codificación, o bien por las degradaciones introducidas por el canal.

La pérdida de información introducida por un códec de voz se puede reducir mediante la selección de un esquema de codificación apropiado. Normalmente, la voz se codifica mediante parámetros que representan la señal eficientemente, de modo que, mediante un conjunto reducido de bits, se puede reconstruir la voz con una pérdida mínima de calidad. Para realizar eficientemente esta función los codificadores aprovechan las características intrínsecas de la voz y del sistema de percepción auditivo. Una vez llevada a cabo esta transformación, la señal de voz se convierte en una fuente de información, de modo que, antes de ser enviada a través del canal, se le introduce una cierta redundancia mediante la codificación de canal, haciéndola así más inmune a las distorsiones introducidas por el medio de transmisión.

3.3.1. Tipos de Codificadores

Históricamente, el ancho de banda utilizado para los codificadores de voz ha sido el del canal telefónico (300-3400 Hz), lo que da lugar a una primera clasificación en función del ancho de banda utilizado para la transmisión de voz. Los codificadores que hacen uso del ancho de banda del canal telefónico son denominados de banda estrecha, mientras que aquellos que hacen uso de una banda mayor reciben el nombre de codificadores de banda ancha. Obviamente, el hecho de utilizar una banda mayor dota a la señal reconstruida de una mayor naturalidad, aunque el uso de codificadores de voz de banda estrecha es suficiente para establecer la inteligibilidad del mensaje, así como las principales características del locutor. De hecho, los principales codificadores utilizados en las arquitecturas actuales de VoIP siguen haciendo uso de codificadores de banda estrecha, principalmente por la

compatibilidad con las redes de telefonía fija, lo que motiva que centremos nuestro interés en estos últimos.

A lo largo de las últimas décadas se han desarrollado numerosos esquemas de codificación de la señal de voz. Clásicamente, en la literatura, éstos se dividen en los siguientes tres grupos.

- **Codificadores de forma de onda.** Presentan una estrategia basada en el procesado muestra a muestra de la voz. El objetivo final de estos codificadores es conseguir reconstruir una señal que se aproxime lo máximo posible a la original y que, por tanto, proporcione una reproducción aproximada.

- **Codificadores paramétricos o Vocoders.** En este caso, la estrategia seguida pretende alcanzar una señal perceptualmente equivalente a la original. Con este fin, se determinan ciertos parámetros que caracterizan a los distintos segmentos de voz individuales y que constituyen la información enviada por el codificador. A partir de esos parámetros el decodificador genera una señal, con forma de onda distinta a la original, que el oído humano percibirá como similar a la original. La principal ventaja obtenida es la reducción drástica de la tasa a transmitir. Como contrapartida, la voz reproducida pierde cierta naturalidad.

- **Codificadores híbridos.** Conjugan las características de los dos anteriores con el fin de obtener la calidad de voz de los primeros con una tasa de bits reducida. Hoy en día, la mayoría de los codificadores utilizados en los sistemas de transmisión digitales de voz utilizan distintas estrategias enmarcadas en esta tipología.

En las próximas secciones se resumen las principales características de los distintos tipos de codificadores, atendiendo principalmente a la tasa resultante de bits, la calidad de la voz decodificada y su robustez frente a posibles degradaciones. Además, dentro de cada tipología se presentarán los principales estándares de codificación de voz.

3.4. Codificadores de Forma de Onda

Los codificadores de forma de onda operan habitualmente en rangos de razones de transmisión superiores a 2 bits por muestra. Por tanto, para el caso de banda estrecha (utilizando una frecuencia de muestreo de 8 kHz) las tasas obtenidas son superiores a 16 kbits/s.

La forma más inmediata para los codificadores de forma de onda se basa en el empleo de la codificación PCM (*Pulse Code Modulation*) de las muestras de la señal de voz en

el dominio temporal, tal y como hace el estándar G.711 [49] desarrollado por la CCITT (*Comité Consultatif International Téléphonique et Télégraphique*), actual ITU. En este caso, se trata de un codificador de banda estrecha que adquiere la señal de voz mediante un proceso de muestreo a 8 kHz. Este codificador aplica un cuantizador logarítmico (ley μ en EE.UU. y ley A en Europa) que asigna códigos binarios de 8 bits a cada muestra, consiguiendo un tasa final de 64 kbps.

La tasa de bits de estos codificadores se puede reducir notablemente aplicando técnicas predictivas. De esta forma, aparecen los codificadores DPCM (*Differential Pulse Code Modulation*) con distintos órdenes de predicción [50], cuyos coeficientes se determinan a partir de estimaciones de la función de autocorrelación de la voz. Para aplicar estas técnicas eficientemente, y debido a la no estacionariedad de la señal de voz, los coeficientes deben de ir modificándose a medida que cambien las funciones de autocorrelación a corto plazo. Estos esquemas de codificación reciben el nombre de técnicas APC (*Adaptive Prediction Coding*) y comprenden un gran número de variantes [51]. Si la información de predicción se transmite explícitamente al receptor DPCM da lugar a las denominadas técnicas predictivas con adaptación hacia delante, APF (*Adaptive Prediction Forward*), mientras que si es derivada de la historia de la señal recientemente cuantizada, se denominan técnicas de adaptación hacia atrás, APB (*Adaptive Prediction Backward*). Además, si a la adaptación de los predictores se une la adaptación de los cuantizadores a la señal de entrada, se generan unas técnicas denominadas técnicas ADPCM (*Adaptive* DPCM), ampliamente utilizadas y que constituyen la base de los estándares G.721 [52] y G.723 [53].

Una técnica muy relacionada con la anterior es la codificación en sub-bandas, donde se aproxima la señal en el dominio de la frecuencia en lugar de en el dominio temporal. En esta técnica la banda de la señal de voz se divide normalmente en cuatro o más sub-bandas mediante un banco de filtros paso-banda. La salida correspondiente a cada filtro es entonces diezmada, para tener una frecuencia de muestreo acorde con su ancho de banda, siendo finalmente codificada mediante un codificador PCM o DPCM. Asignando los bits de forma apropiada en las diferentes bandas se puede controlar el espectro del error de reconstrucción. Así, se pueden usar más bits en las bandas de frecuencia más baja para representar con mayor precisión la estructura de los formantes y el *tono fundamental*, o *pitch*, y, en general, hacer adaptable dicha asignación de bits. Un ejemplo de este tipo de codificadores es el estándar G.722 [54], también desarrollado por la ITU, que para una banda de 7 kHz consigue una tasa de bits de 64 kbit/s utilizando dos sub-bandas.

3.5. Codificadores Paramétricos

Los codificadores paramétricos operan a velocidades de transmisión menores que los codificadores de forma de onda, en la región de los 0.5 bits por muestra e inferiores, pero la calidad de la señal de voz reproducida, aunque resulta inteligible, normalmente sufre de una pérdida de naturalidad y de las características propias del locutor.

Estos codificadores también son conocidos como *vocoders* y se basan en el modelo de producción de voz que ya introdujimos en la sección 2.3.2. En este caso, los vocoders no trabajan muestra a muestra, sino que parten de una segmentación en tramas de la señal de voz, de modo que sólo transmiten los parámetros del modelo correspondientes a cada una de las tramas. En el receptor, la síntesis de voz se lleva a cabo mediante el modelo con los parámetros recibidos. En oposición a los codificadores de forma de onda, los vocoders no persiguen la reproducción de la señal de voz original, sino que su objetivo consiste en obtener una señal sintetizada que perceptualmente sea similar a la original.

3.5.1. Vocoders LPC

El vocoder más ampliamente estudiado es el basado en el modelo LPC (*Linear Prediction Coding*), el cual vimos en el capítulo anterior (figura 2.3). Este modelo se basa en la suposición de que la señal de voz es la salida de un filtro digital, como el mostrado en la figura 3.4, que representa el tracto vocal. A su vez, la excitación del tracto vocal puede estar formada por pulsos glotales casi periódicos (sonidos sonoros), modelados como un tren de pulsos, o por turbulencias del aire que proviene de los pulmones (sonidos sordos), modeladas por ruido blanco.

Filtro LPC

Así pues, la señal de voz $s(n)$ se forma como la respuesta de un sistema lineal a una cierta entrada $u(n)$,

$$s(n) = \sum_{k=1}^{p} a_k s(n-k) + \sigma \sum_{l=0}^{q} b_l u(n-l) \qquad b_0 = 1 \tag{3.1}$$

donde a_k, $1 \leq k \leq p$, b_l, $1 \leq l \leq q$, y la ganancia σ son los parámetros del sistema. La ecuación (3.1) nos dice que la salida $s(n)$ es una función lineal de las salidas anteriores y de las entradas actual y anteriores. Es decir, la señal $s(n)$ se puede predecir como una combinación lineal de anteriores salidas y entradas, de aquí el nombre de predicción lineal.

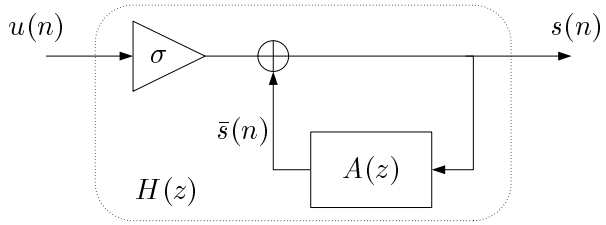


Figura 3.4: Representación del tracto vocal a través de un filtro LPC.

También podemos expresar la ecuación (3.1) en el dominio de la frecuencia haciendo la transformada z a ambos lados de la igualdad. Si $H(z)$ es la función de transferencia del sistema, entonces obtenemos la siguiente expresión,

$$H(z) = \frac{S(z)}{U(z)} = \sigma \frac{1 + \sum_{l=1}^{q} b_l z^{-l}}{1 - \sum_{k=1}^{p} a_k z^{-k}} \tag{3.2}$$

donde $S(z)$ es la transformada z de $s(n)$, y $U(z)$ es la transformada z de $u(n)$. $H(z)$ representa el modelo polos-ceros más general. Las raíces de los polinomios del numerador y del denominador son los ceros y los polos del modelo, respectivamente. Hay dos casos especiales en el modelo que son de interés:

1. El modelo todo-ceros: $a_k = 0$, $1 \leq k \leq p$
2. El modelo todo-polos: $b_l = 0$, $1 \leq l \leq q$

En el lenguaje estadístico, el modelo todo-ceros se conoce con el nombre de modelo de media móvil (MA, *moving average*) y el modelo todo-polos se denomina modelo autoregresivo (AR, *auto-regressive*). El modelo que se usa más frecuentemente para representar de forma paramétrica la señal de voz es el modelo todo-polos, donde la señal $s(n)$ viene dada como una combinación lineal de los valores anteriores y de una entrada $u(n)$,

$$s(n) = \sum_{k=1}^{p} a_k s(n-k) + \sigma u(n) \tag{3.3}$$

donde σ es un factor de ganancia, a_k los denominados coeficientes de predicción y p es el orden de predicción. La función de transferencia $H(z)$ queda como,

$$H(z) = \frac{\sigma}{1 - \sum_{k=1}^{p} a_k z^{-k}} = \frac{\sigma}{1 - A(z)} \tag{3.4}$$

Los coeficientes del filtro de predicción lineal $A(z)$ se obtienen llevando a cabo un proceso de minimización del error de predicción sobre un conjunto de N muestras (trama). Se considera que el filtro lineal permanece constante durante cierto intervalo de tiempo, que suele durar de 10 a 20 ms, modelando la evolución del tracto vocal como una sucesión de filtros lineales estacionarios ya que la velocidad de articulación del tracto vocal no es muy superior a 10 veces por segundo. La predicción realizada puede ser expresada como,

$$\bar{s}(n) = \sum_{k=1}^{p} a_k s(n-k) \tag{3.5}$$

Por tanto, el error entre el valor actual de $s(n)$ y el predicho $\bar{s}(n)$ es,

$$e(n) = s(n) - \bar{s}(n) = s(n) - \sum_{k=1}^{p} a_k s(n-k) \tag{3.6}$$

a partir del cual podemos obtener el error cuadrático E definido como,

$$E = \sum_{n=0}^{N-1} e^2(n) \tag{3.7}$$

Los coeficientes LPC se calculan minimizando este error cuadrático, es decir,

$$\frac{\partial E}{\partial a_k} = 0 \qquad k = 1, \ldots, p \tag{3.8}$$

expresión a partir de la cual obtenemos el sistema de ecuaciones normales,

$$\sum_{k=1}^{p} a_k R_s(|i-k|) = R_s(i) \qquad 1 \leq i \leq p \tag{3.9}$$

donde R_s representa la función de autocorrelación de la señal $s(n)$.

Como se observa en la expresión (3.9), para el cálculo de los distintos coeficientes a_k se precisa de una descripción matemática de cada segmento de $s(n)$ que no se encuentra disponible. Por este motivo, es necesario estimar los valores de la matriz de autocorrelación $R(|i-k|)$ a partir de las observaciones consecutivas de $s(n)$ con $0 \leq n \leq N-1$.

El método para la resolución del modelo todo-polos descrito anteriormente se denomina método de autocorrelación. Este método considera que se dispone de un conjunto infinito de muestras para determinar la función de autocorrelación R_s y, por consiguiente,

dicha función sólo depende de la diferencia $|i-k|$, obteniéndose así una matriz Toeplitz. Basándose en dicha característica se han desarrollado métodos muy eficaces de resolución de dicho sistema de ecuaciones, como, por ejemplo, el algoritmo recursivo de Levinson y Durbin [55]. Sin embargo, en la práctica no se dispone de infinitas muestras de un segmento, por lo que es necesario multiplicar la señal $s(n)$ por una función de ventana w_n para obtener otra señal $s_w(n)$ que es cero fuera del intervalo $0 \leq n \leq N-1$, siendo N el tamaño del segmento de análisis. Este enventanado hace del método de autocorrelación un método aproximado.

Una alternativa más precisa la proporciona el método de covarianza. En este método se calcula el error E en un intervalo finito de tamaño igual al del segmento de análisis, $0 \leq n \leq N-1$, obteniendo la siguiente expresión para los elementos de la matriz de covarianza,

$$R_s(i,k) = \sum_{n=0}^{N-1} s(n-k)s(n-i) \tag{3.10}$$

Hay que señalar que según se desprende de esta última expresión, en este método se ha de conocer la señal $s(n)$ en el intervalo $-p \leq n \leq N-1$. En este caso la matriz del sistema de ecuaciones ya no es una matriz Toeplitz aunque sí simétrica. Para resolver dicho sistema de ecuaciones se utiliza la denominada descomposición de Cholesky [8].

El método de autocorrelación tiene la ventaja, frente al de covarianza, de asegurar la estabilidad del filtro de síntesis usado durante la reconstrucción de la voz. Además, el procedimiento de covarianza requiere un mayor esfuerzo computacional [56]. Por estas razones, el método de covarianza es menos utilizado que el método de autocorrelación.

Excitación del Modelo LPC

En el modelo LPC, la excitación se modela por medio de la señal $u(n)$. Como mencionamos al principio de esta sección, en caso de que el segmento de voz sea sonoro, la señal de excitación adopta la forma de un tren de pulsos. Por contra, cuando se trata de un segmento sordo, la excitación se hace corresponder con ruido blanco. Si analizamos el espectro de este modelo, podemos concluir que la envolvente espectral la determina el filtro LPC, mientras que el espectro de la señal de excitación crea la estructura fina.

La excitación en el vocoder LPC es generada en el decodificador sólo a partir de una especificación relativamente simple del carácter general del segmento de voz actual, sin

enviar realmente información que especifique la forma de la excitación. La única información que se envía consiste en los parámetros del filtro (coeficientes LPC y ganancia), información de si el segmento es sordo o sonoro y una estimación del valor del periodo de *pitch* en el caso de que el segmento sea sonoro.

Cuantización de Parámetros LPC

En la literatura pueden encontrarse numerosas técnicas para cuantizar los parámetros LPC. Estos cuantizadores pueden ser uniformes o no uniformes, pero ya que los no uniformes presentan una menor distorsión son más utilizados en la actualidad. No sólo se busca que al cuantizar los parámetros LPC se introduzca la menor distorsión posible, sino que también es necesario que el filtro todo-polos se mantenga estable tras la cuantización.

Normalmente, no se utiliza la cuantización directa de los coeficientes LPC ya que pequeños errores pueden producir errores espectrales relativamente grandes, además de filtros todo-polos $H(z)$ inestables [56]. Por este motivo, se utilizan otras representaciones de los coeficientes LPC que aseguran la estabilidad. Dentro de estas representaciones destacan los RC (*Reflections Coefficients*) [57], los coeficientes LAR (*Log Area Ratio*) [58], los ASRC (*ArcSine Reflection Coefficients*) [59] y los LSF (*Line Spectral Frecuency*) [60]. Los más extendidos son los coeficientes LAR y los LSF, puesto que presentan las mejores características de cuantización. Además, los LSF presentan una ventaja adicional ya que si éstos se encuentran en orden ascendente, aseguran la estabilidad del filtro todo-polos [61].

Un ejemplo de vocoder LPC es el estándar del gobierno estadounidense FS1015 [62]. Este codificador muestrea la señal de voz a 8 kHz y trabaja sobre tramas de 22.5 ms de duración, obteniendo una tasa final de 2.4 kbps. No obstante, aunque estos codificadores reducen sustancialmente la tasa de bits, en la actualidad son poco utilizados debido a su pobre calidad. La señal sintetizada suena artificial o poco natural y la identidad del locutor resulta difícil de reconocer. Estos codificadores tienden a degradarse todavía más si la señal de voz original contiene ruido de fondo.

3.5.2. Codificadores Sinusoidales

Los codificadores sinusoidales constituyen un tipo de vocoder basado en el modelo de producción de voz, en el que se toma la voz como el resultado de pasar una excitación generada por la glotis a través de un filtro lineal variante en el tiempo (caracterizado

por la secuencia de coeficientes LPC) que modela las características resonantes del tracto vocal.

En el modelo de vocoder LPC, la excitación se generaba como un tren de pulsos (de periodo igual al del *pitch*) para los sonidos sonoros y como una señal ruidosa para los sordos. A diferencia de éstos, los codificadores sinusoidales generan la excitación en el decodificador generando una suma de sinusoides, cuyas frecuencias y fases son modificadas en segmentos sucesivos para representar el carácter cambiante del espectro de la señal de voz original [63].

La motivación de esta representación viene dada porque la excitación, cuando se asume que es periódica, puede ser representada por una descomposición en series de Fourier en la que cada componente armónica corresponde con una única onda sinusoidal. De hecho, la calidad de la voz sintetizada dependerá de la codificación que se introduzca de las fases y amplitudes de las ondas sinusoidales utilizadas. Una de las más exitosas aplicaciones de esta técnica de modelado para codificación a baja tasa ha sido el codificador IMBE (*Improved Multi-Band Excitation*), el cual fue utilizado para el sistema INMARSAT-M en el año 1990 [64] y para el INMARSAT-Mini-M en 1994 [65].

3.6. Codificadores Híbridos

Los codificadores híbridos surgen como una mezcla de los codificadores de forma de onda y los codificadores paramétricos. Éstos pueden considerarse codificadores paramétricos atendiendo a que emplean un modelo paramétrico, aunque también intentan preservar la forma de onda de la señal de voz. En el caso de que consideremos un modelo LPC, podemos plantear que el objetivo que persigue un codificador paramétrico sería conseguir que la señal de excitación, $\hat{e}(n) = \sigma\hat{u}(n)$, haga que la síntesis se aproxime lo máximo posible a la señal de voz original. El principal problema que surge es cómo determinar los parámetros óptimos de la nueva excitación $\hat{e}(n)$ una vez determinado el tipo de modelo que se utilizará. Normalmente, esta cuestión se resuelve mediante un procedimiento denominado *análisis por síntesis*, del que se muestra un diagrama básico en la figura 3.5. Este método consiste en obtener aquellos parámetros que logran ajustar mejor la señal sintetizada $\hat{s}(n)$ a la señal original $s(n)$. Normalmente, esta tarea se realiza aplicando un criterio LSE (*Least Square Error*), es decir, minimizando el error cuadrático definido como,

$$E = \sum_{n=0}^{N-1} \left(s(n) - \hat{s}(n)\right)^2 \tag{3.11}$$

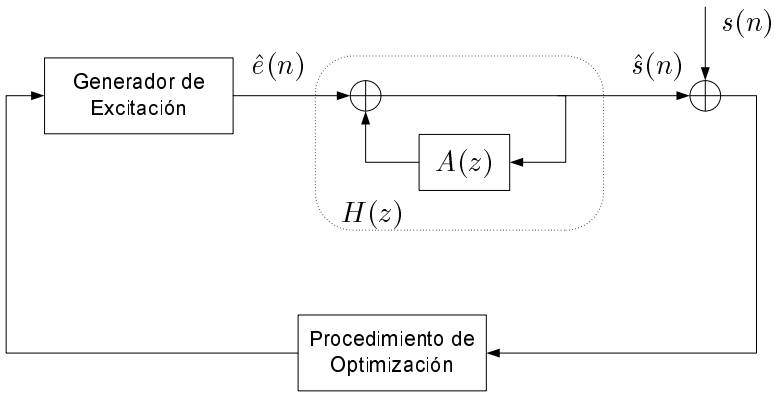


Figura 3.5: Esquema de selección de parámetros de un codificador híbrido mediante el proceso de análisis por síntesis.

Cada muestra de la señal de excitación $\hat{e}(n)$ afecta a muchas muestras de la señal de voz reconstruida $\hat{s}(n)$, debido a la estructura recursiva del filtro de síntesis expresada en la ecuación (3.3). Por consiguiente, la elección de una señal de excitación se realiza midiendo su efecto sobre la señal de voz reconstruida durante más de una muestra. Es decir, la decisión sobre la mejor representación cuantizada no se realiza de forma instantánea sino que se retarda durante un intervalo que incluye varias muestras. Este tipo de aproximación se denomina codificación de decisión retardada. Como dicha decisión depende del error entre la señal original y la sintetizada, estas técnicas requieren síntesis durante el análisis, y el procedimiento se conoce como codificación adaptable usando predicción y análisis por síntesis. La característica principal de un esquema de codificación basado en análisis por síntesis es que el codificador incluye al propio decodificador. De esta forma, el codificador puede conocer en cada momento la señal decodificada $\hat{s}(n)$. Así, mediante un proceso iterativo (representado por el bucle de la figura 3.5) se lleva a cabo el proceso de optimización. Finalmente, el codificador envía la información referente al tracto vocal (coeficientes del filtro LPC) junto a los parámetros que definen la excitación.

Este tipo de codificadores obtienen una buena calidad de la señal de voz en el rango de razones de transmisión que va de 0.5 a 2 bits/muestra, lográndose así tasas de bits comprendidas entre 4 y 16 kbps. Debido a las buenas relaciones entre calidad perceptual y las moderadas tasas de bit conseguidas, la mayoría de los codificadores actuales utilizados en telefonía móvil y VoIP hacen uso de esta tipología. Las subsecciones que se presentan a continuación desarrollan los principales esquemas de codificación híbridos basados en el modelo LPC.

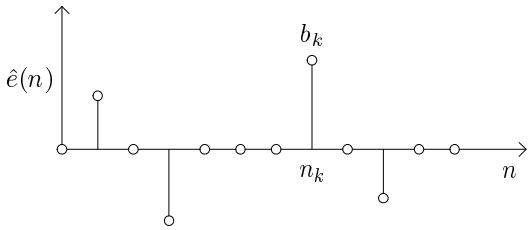


Figura 3.6: Ejemplo de excitación multipulso.

3.6.1. Codificadores Multipulso

El esquema de codificación multipulso [66] construye la señal de excitación $\hat{e}(n)$ mediante una serie de L pulsos dados por unas ciertas amplitudes b_l y posiciones n_l ($l = 0, \ldots, L-1$), de modo que podemos expresar la excitación como,

$$\hat{e}(n) = \sum_{l=0}^{L-1} b_l \delta(n - n_l) \tag{3.12}$$

donde $\delta(n)$ define la función impulso unidad. La figura 3.6 muestra un ejemplo de este tipo de excitación en el que podemos ver que la señal de excitación está constituida por ceros, excepto en algunas posiciones donde se ubican los pulsos.

Una vez que los parámetros LPC del modelo son seleccionados es necesario determinar los parámetros de la excitación, es decir, las posiciones y amplitudes de los pulsos. Atendiendo al criterio LSE, podemos decir que los parámetros de la excitación han de ser aquellos que minimicen la expresión,

$$E = \sum_{n=0}^{N-1} \left(s(n) - h(n) * \hat{e}(n)\right)^2 \tag{3.13}$$

respecto a las amplitudes b_k y las posiciones n_k ($k = 0, \ldots, L-1$). Así mismo, $h(n)$ corresponde con la respuesta impulsiva del filtro LPC utilizado, que normalmente suele corresponderse con un filtro todo-polos como el expresado en (3.4).

Un caso particular de la codificación multipulso es el codificador RPE (*Regular Pulse Excitation*). En este caso, los pulsos se distribuyen de forma uniforme o regularmente espaciados, de manera que no es necesario transmitir la posición de cada pulso, pues basta con la posición del primero. Dentro de este tipo de codificadores podemos encontrar el codificador FR (*Full Rate*) [67], muy popular por ser el primer estándar de codificación de voz digital utilizado por el sistema de telefonía móvil GSM. Este codificador de banda

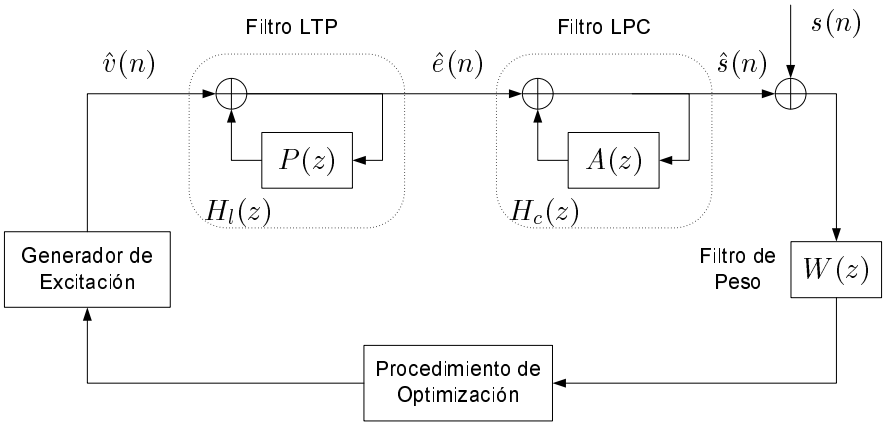


Figura 3.7: Esquema de codificación mejorado basado en el procedimiento de análisis por síntesis.

estrecha, que se caracteriza por obtener una tasa de 13 kbps, lograba en su momento (a principio de los años 90) un buen compromiso entre calidad y complejidad computacional.

Introduciendo ciertas modificaciones sobre el procedimiento de análisis por síntesis, los esquemas de codificación multipulso (por ejemplo, FR hace uso de ellas) consiguen realzar su rendimiento. Concretamente, el diagrama presentado en la figura 3.7 muestra un esquema de codificación de análisis por síntesis mejorado, donde se han introducido, respecto al diagrama presentado en la figura 3.5, dos nuevos bloques: el filtro de *pitch* y el filtro de peso. A continuación pasamos a detallar los principios de funcionamiento de ambos bloques.

Filtro de *Pitch* o LTP

El filtro de *pitch* introduce un nuevo termino de filtrado $H_l(z)$ sobre la respuesta del filtro LPC $H_c(z)$, resultando en una respuesta total $H(z) = H_l(z)H_c(z)$. Como vimos con anterioridad, el filtro LPC consiste en un predictor a corto plazo (STP, *Short-term Predictor*) que modela la envolvente del espectro. Del mismo modo, el filtro de *pitch* está constituido por un predictor pero que, en este caso, se encarga de modelar las correlaciones a largo plazo debidas al *pitch*. De esta forma, el predictor a largo plazo (LTP, *Long-term predictor*) intenta eliminar las redundancias a largo plazo del residuo o error de predicción $e(n)$, de forma que el nuevo residuo $v(n)$ pueda ser correctamente representado por una

aproximación $\hat{v}(n)$ con menos pulsos. La forma general de un filtro LTP es

$$P(z) = 1 - \sum_{k=-(q-1)/2}^{(q-1)/2} b_k z^{-(M+k)} \tag{3.14}$$

donde M es el retardo en muestras y b_k son los coeficientes de predicción de retardo largo, mientras que q determina el orden del filtro. El valor M, cuyo rango suele ser de 2 a 20 ms, coincide con el periodo de *pitch* en los segmentos sonoros, mientras que es un valor aleatorio en los segmentos sordos. El retardo M y los coeficientes b_k se determinan bien a partir de la señal de voz o a partir de la señal diferencia después de eliminar las correlaciones de retardo corto. El procedimiento para determinar el valor de los coeficientes b_k, para un M dado, es similar que para el caso de los coeficientes de predicción de retardo corto a_k, pudiendo determinarse mediante el método de autocorrelación o covarianza [68]. Sin embargo, es necesario determinar con antelación el valor del retardo M.

Típicamente se utilizan de uno a tres coeficientes de predicción y sus valores se adaptan con el tiempo a un ritmo que va de 50 a 200 veces por segundo. Si se aumenta el orden del predictor se obtiene una predicción mejor, pero a cambio se necesitan más bits para codificar los coeficientes adicionales. Para segmentos periódicos, la razón principal de disponer de múltiples coeficientes es suministrar interpolación entre las muestras en caso de que el periodo no se corresponda con un número entero de muestras. En lugar de usar un predictor de orden alto se puede también usar un predictor de primer orden con un retardo no entero [69], lo cual permite una cuantización más eficiente de los parámetros del predictor.

Filtro de Peso

El oído presenta una propiedad denominada enmascaramiento que consiste en que el sistema auditivo tiene poca capacidad de detectar ruido en las bandas de frecuencia en las cuales la señal de voz tiene una alta energía, tal y como ocurre en las bandas donde aparecen los formantes de la voz.

Cuando se reduce el número de bits para la codificación de la excitación se aumenta la energía del ruido, lo que hace que el ruido sea más audible. Sin embargo, teniendo en cuenta la propiedad de enmascaramiento del oído es posible reducir este efecto. Para ello se realiza el filtrado de la señal excitación mediante un filtro $W(z)$ antes de ser codificada, de forma que se reduzca el ruido de cuantización en los valles y se aumente en los picos del espectro. Así, al aplicar el proceso de minimización del error cuadrático medio pesado

mediante el filtro $W(z)$ se consigue aumentar la calidad de la voz sintetizada, a pesar de que la SNR sea menor.

Puede observarse que dicho procedimiento de peso no afecta ni a la razón de transmisión ni a la complejidad del procedimiento de síntesis. Aumenta sólo la complejidad del codificador. El uso de un procedimiento de enmascaramiento del ruido fue, inicialmente, investigado para los esquemas de codificación adaptable predictiva y se obtuvo que un filtro de peso apropiado [70] responde a la siguiente forma,

$$W(z) = \frac{F(z/\gamma_1)}{F(z/\gamma_2)} = \frac{1 - \sum_{k=1}^{p} f_k \gamma_1^k z^{-k}}{1 - \sum_{k=1}^{p} f_k \gamma_2^k z^{-k}} \qquad 0 \leq \gamma_2 \leq \gamma_1 \leq 1 \tag{3.15}$$

donde $F(z)$ es un predictor de retardo corto. Para la mayor parte de las aplicaciones se puede seleccionar $F(z) = A(z)$. Los parámetros γ_1 y γ_2 tienen un valor entre 0 y 1 que controla la energía del error en las regiones formantes. Nótese que al decrementar γ aumenta el ancho de banda de los ceros de $F(z/\gamma)$. Los valores de γ_1 y γ_2 se determinan heurísticamente mediante los correspondientes pruebas auditivas.

Cuando se utilizan tasas de transmisión pequeñas el efecto de pesar el error es menos perceptible. Una razón que explica este efecto es que el nivel de ruido es tan alto que, a pesar del enmascaramiento, el ruido sigue siendo audible. Además, la suposición de que el ruido de cuantización tiene un espectro plano ya no es válida, lo que hace que el resultado del proceso de enmascaramiento sea menos predecible.

3.6.2. Codificadores CELP

De todas las técnicas híbridas, la más extendida es la denominada codificación CELP (*Code Excited Linear Prediction*) [71], la cual se deriva de la codificación multipulso. Veíamos que el esquema de análisis por síntesis mejorado (figura 3.7) tiene un doble objetivo. En primer lugar, intenta obtener una señal de excitación $\hat{v}(n)$ con el menor número de redundancias posibles. Por este motivo, se introducen dos etapas de predicción que consisten en un predictor a corto plazo (filtro LPC) y otro a largo plazo (filtro LTP). De este modo, en caso de que la excitación $\hat{v}(n)$ fuera igual a la señal residual $v(n)$, se reproduciría la señal de voz original $s(n)$. El segundo objetivo que se persigue es la búsqueda de una excitación que represente el residuo de la forma más precisa posible. En este sentido, la innovación que introducen los codificadores CELP frente a los multipulso consiste en incorporar un cierto diccionario de M códigos fijos de excitación $c_k(n)$ ($k =$

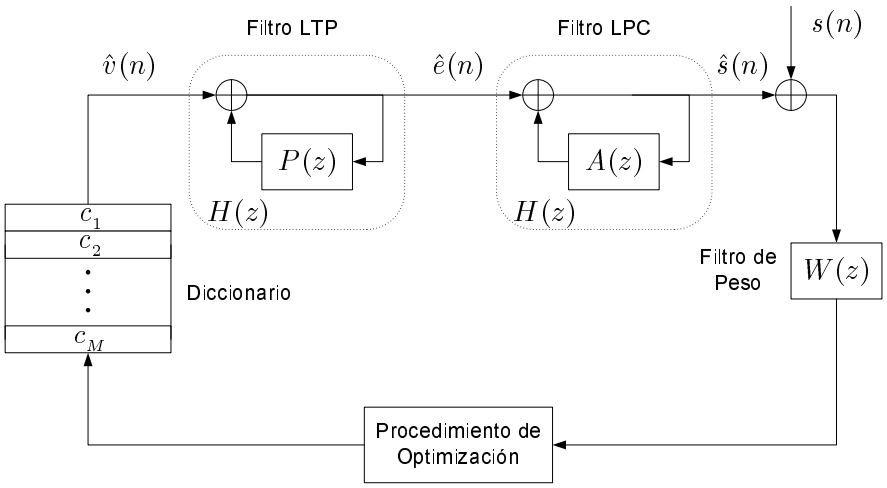


Figura 3.8: Esquema general de un codificador CELP.

$0, \ldots, M-1$). La formulación básica de la codificación CELP supone que las M entradas del diccionario están formadas por secuencias aleatorias Gausianas de varianza unitaria. De este modo, el esquema final de un codificador CELP se puede representar mediante el diagrama de la figura 3.8. De este esquema se desprende que la excitación se construye como $\hat{v}(n) = g_c c(n)$, donde la ganancia g_c se obtiene durante el proceso de optimización junto a $c(n)$ (código óptimo de los M posibles del diccionario).

En este caso, tanto el codificador como el decodificador almacenan un diccionario de M secuencias posibles para la excitación, donde cada secuencia tiene una longitud L. La excitación en cada segmento está completamente descrita por el índice del vector apropiado dentro del diccionario. Este índice se determina mediante una búsqueda exhaustiva sobre todos los vectores del diccionario y corresponde al índice de aquel vector del diccionario que produce el error más pequeño entre la señal original y la reconstruida. La transmisión del índice requiere $(log_2 M)/L$ bits/muestra, con valores típicos en el rango de 0.2 a 2 bits/muestra. Considerando una frecuencia de muestreo de 8 kHz, las tasas totales de transmisión obtenidas se sitúan entre 4 y 16 kbit/s.

Un ejemplo de implementación de un codificador CELP viene dado por el estándar estadounidense FS1016 [72], el cual obtiene una tasa final de 4.8 kbps. Este codificador utiliza una longitud de trama de 30 ms (con un nivel de subtrama de 7.5 ms). El diccionario que utiliza está formado por 512 códigos de 60 muestras ternarias (los posibles valores son -1, 0 y 1). Otro ejemplo de uso del esquema de codificación CELP lo constituye el codificador de bajo retardo ITU-T G.728 [73]. En este caso, la tasa de bits obtenida es de 16 kbps puesto que la longitud de la trama es de tan sólo 2.5 ms, obteniendo así un bajo

retardo algorítmico.

La técnica CELP ha derivado en un amplio número de variantes, como por ejemplo los esquemas VSELP (*Vector Sum Excited Linear Prediction*) [74], adoptado por los estándares IS-54/136 y GSM-HR (GSM *Half-Rate*) , QCELP (*Qualcomm* CELP) [75], adoptado por el estándar IS-95 de telefonía móvil, y el ACELP (*Algebraic* CELP) [76]. La variante ACELP es la que ha tenido más éxito durante los últimos años, la cual se basa en la selección de códigos ralos. Estos códigos están formados principalmente por ceros, lo que permite una rápida búsqueda en el diccionario. Los valores distintos de cero se hallan permutando una serie de pulsos en una serie de posiciones prefijadas. De este modo, los codificadores ACELP recurren a una solución similar a la codificación multipulso, en la que se introducen ciertas restricciones en la amplitud y ubicación de los pulsos. De ahí que la búsqueda por medio de diccionario de los codificadores CELP se mantenga. Éste fue el esquema de codificación más utilizado durante la década de los años 90, destacando los codificadores EFR (*Enhanced Full Rate*, 12.2 kbps) [77] y AMR (*Adpative Multi-Rate*, el cual dispone de 8 tasas comprendidas entre 4.75 y 12.2 kbps) [78] del sistema europeo GSM (posteriormente AMR fue también seleccionado por UMTS), así como el IS-641 (7.4 kbps) [79] del sistema celular norteamericano IS-136.

3.6.3. Codificadores Híbridos en Aplicaciones VoIP

Mientras que los codificadores anteriormente mencionados fueron principalmente desarrollados para su uso en redes de telefonía celular, también existen otros estándares de codificación, tales como G.723.1 y G.729, que son comúnmente utilizados para aplicaciones VoIP. En particular, el codificador G.723.1 [80] es un códec con dos posibles tasas, 5.3 kbps y 6.3 kbps, en el que la representación de la excitación depende de la tasa escogida. Así, el modo de 5.3 kbps lleva a cabo una codificación CELP, mientras que la tasa de 6.3 kbps establece una representación multipulso de la excitación. La longitud de la trama es de 30 ms, utilizando un look-ahead de 7.5 ms. El estándar G.729 [81] emplea un esquema de codificación CS-ACELP (*Conjugate Structure* ACELP) con una tasa de 8 kbps y una longitud de trama de 10 ms. En este caso el look-ahead es de 5 ms, consiguiendo una importante reducción del retardo algorítmico sobre el estándar G.723.1. Los estándares de G.723.1 y G.729 incluyen sus correspondientes algoritmos para la mitigación de pérdidas de tramas.

Aunque, como acabamos de ver, los codificadores CELP son utilizados en redes de paquetes, éstos fueron originalmente desarrollados para ser utilizados en comunicaciones

móviles, en las que es posible adaptar la codificación en función de las degradaciones presentes en el medio de transmisión. Así, por ejemplo, el estándar de codificación AMR permite la selección de distintas tasas de bits para el codificador de voz de modo que, si las características del canal empeoran, puede reducir la tasa del codificador para aumentar la tasa de protección frente a errores.

Sin embargo, este esquema no puede ser abordado en una red como Internet, en la que la información se encuentra fragmentada en paquetes individuales que pueden seguir caminos distintos. En este caso, la unidad de información codificada menor es la correspondiente a un paquete, el cual puede ser recibido o no en función del estado del canal. Cuando se produce la pérdida de un paquete implícitamente se pierde una porción de información que es igual para todos los codificadores. Pero aquí no acaban los efectos negativos, ya que tal y como se ha estudiado en los codificadores anteriormente vistos, algunos de ellos presentan fuertes dependencias temporales (véase el filtro de *pitch* o LTP) que provocan la expansión del error a las tramas consecutivas a la pérdida. Con el objetivo de eliminar las dependencias intertrama aparece el esquema de codificación iLBC (*internet Low Bit-Rate Codec*) [82], reduciendo así los efectos negativos de la pérdida de una trama. A pesar de su reciente creación en el año 2004, este códec ha tenido un éxito bastante relevante y se ha impuesto en algunas aplicaciones VoIP tan conocidas como *Skype* o *Google Talk*.

iLBC se encuadra dentro de los codificadores híbridos basados en el modelo LPC, aunque en este caso no responde a una estructura CELP de análisis por síntesis. Puesto que pretende evitar la dependencia intertrama, rechaza el uso de filtro de *pitch* para replicar esta periodicidad a largo plazo. Por el contrario, detecta la zona de mayor energía en la excitación de la trama actual, la cual es llamada *estado inicial* (donde normalmente se encuentra el pulso de *pitch*), para llevar a cabo sobre ella una codificación de forma de onda DPCM. En este punto, parecería razonable pensar que el codificador responde a un esquema de forma de onda. Sin embargo, el resto de la excitación no es codificada con DPCM, sino que se realiza a partir de las muestras del *estado inicial*, el cual es utilizado mediante un proceso de tres pasadas para aproximar el resto de la excitación. Este diccionario adaptativo se utiliza para codificar el resto de muestras de la excitación, empleando para ello predictores a largo plazo hacia delante y hacia atrás (pero siempre dentro de la trama actual). De este modo, la decodificación de las muestras correspondientes a una trama no depende de las muestras previas. No obstante, la robustez frente a pérdidas de paquetes presenta el inconveniente de una mayor tasa de bits. iLBC presenta dos modos de funcionamiento denominados en función de su longitud de bloque o trama, siendo éstas

de 20 o 30 ms. En el primero de los casos la tasa de bits obtenida es 15.2 kbps, mientras que en el segundo caso es 13.3 kbps.

El diseño de iLBC está directamente orientado a las características de las redes IP, teniendo como objetivo incrementar la robustez ante la pérdida de paquetes. Ésta característica lo distingue del resto de codificadores actualmente en uso y lo convierte en un caso de especial interés para el desarrollo de la presente tesis. Los detalles de implementación de este codificador se pueden encontrar en el documento RFC 3951 [83].

Capítulo 4

Evaluación de NSR sobre Redes IP

4.1. Introducción

El capítulo anterior presentaba los sistemas de reconocimiento remoto y sus ventajas en aplicaciones orientadas hacia terminales portables. Dentro de éstos, distinguíamos entre dos posibles arquitecturas: por un lado, los estándares para llevar a cabo la tarea de reconocimiento de forma distribuida (arquitectura DSR) y, por otro, aquellos sistemas donde el reconocedor se encuentra totalmente integrado en el servidor (arquitectura NSR).

Veíamos que la arquitectura DSR está diseñada desde un inicio con el objetivo principal de reconocimiento, integrando en el cliente el bloque de adquisición y extracción de características de reconocimiento (*Front-End*). Así pues, esta arquitectura presenta la ventaja de que las características son obtenidas directamente de la voz original, utilizando una codificación eficiente para la transmisión de éstas. Por contra, la arquitectura NSR envía la señal de voz utilizando un codificador de voz convencional. En este caso, el sistema de reconocimiento se encuentra integrado en su totalidad (a excepción del sistema de adquisición que es incorporado por el codificador) en el servidor. El hecho de utilizar un codificador para el envío de la señal de voz supone una gran ventaja, ya que supera las dificultades de implantación de los sistemas DSR. No obstante, ya que los codificadores tienen como objetivo la transmisión de la señal de voz con la máxima calidad perceptual, esta arquitectura presenta ciertas desventajas [84]:

1. Distorsión de codificación. El proceso de codificación y decodificación modifica las características espectrales de la voz, dando lugar a una posible reducción de la eficiencia del sistema.

2. Errores de transmisión. En el caso concreto de la transmisión en redes de paquetes, el rendimiento se puede ver severamente afectado debido a las pérdidas introducidas por la red.

3. Retardo. La codificación, transmisión y posibles procesos de recuperación de errores incrementan el tiempo de respuesta del sistema, pudiendo dar lugar a potenciales molestias para el usuario. Normalmente, el sistema de reconocimiento es sólo un subsistema dentro de una cierta aplicación remota. En tal caso, la efciencia del sistema completo vendrá determinada por la eficiencia de los subsistemas individuales.

Este capítulo tiene como objetivo la evaluación de los sistemas NSR, principalmente frente a las degradaciones introducidas por una red de paquetes. De este modo, evaluaremos la capacidad de reconocimiento de esta arquitectura utilizando los codificadores de voz más extendidos en la actualidad. En los capítulos posteriores, estos resultados servirán como referencia para evaluar las mejoras obtenidas por las propuestas realizadas.

4.2. Características del Canal IP

Ya que este capítulo tiene por objeto explorar el reconocimiento de voz sobre IP adoptando una arquitectura NSR, es necesario conocer las características de este medio de transmisión, con el fin de identificar las posibles fuentes de error en la transmisión y sus consecuencias. En primer lugar, llevaremos a cabo un breve repaso sobre los protocolos más utilizados en VoIP, para posteriormente introducir las fuentes de degradación de las aplicaciones en tiempo real sobre redes IP. Finalmente, concluiremos esta sección presentando los principales modelos para la simulación de estas degradaciones.

4.2.1. Protocolos VoIP

Cuando una aplicación precisa de una comunicación fiable entre dos extremos se emplea el protocolo TCP (*Transmission Control Protocol*), ya que es un protocolo orientado a conexión que permite la retransmisión de los paquetes perdidos. Sin embargo, las aplicaciones en tiempo real, dentro de las cuales se encuentra VoIP, tienen estrictas exigencias sobre los retardos sufridos por los paquetes, por lo que la retransmisión de éstos sería totalmente inútil. Ésta es la razón por la que en la capa de transporte de este tipo de aplicaciones se utilice el protocolo no orientado a conexión UDP (*User Datagram Protocol*), el cual

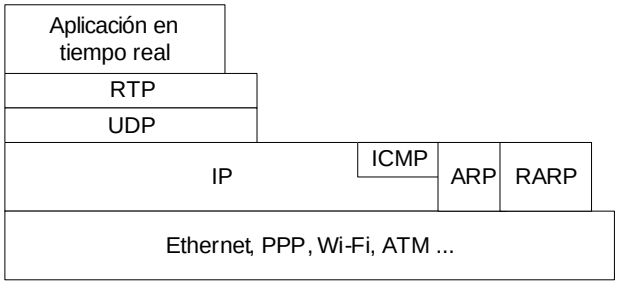


Figura 4.1: Torre de protocolos utilizada para la transmisión de datos en tiempo real.

presta un servicio de datagrama al nivel superior donde es empleado el protocolo RTP (*Real Time Protocol*).

La torre de protocolos empleada dentro de Internet para las aplicaciones en *tiempo real* es la mostrada en la figura 4.1. En ésta se observa como sobre UDP se sitúa RTP, protocolo que se encarga de llevar a cabo ciertas tareas para facilitar implementaciones de servicios con requerimientos de tiempo real. Concretamente, en el encapsulado propio de este protocolo se realizan las tareas de:

- Identificación del tipo de información transportada. Normalmente, las aplicaciones en tiempo real suelen incluir diferentes medios codificados independientemente. Así, con este identificador el receptor es capaz de seleccionar el decodificador adecuado para cada flujo de información.

- Numeración secuencial. En el proceso de transmisión no se garantiza el orden de recepción (característica de las redes basadas en datagrama), por lo que, para recuperar el orden de la secuencia original, es necesario incluir etiquetas de numeración.

- Inclusión de marcas de tiempo. Aunque RTP no se encarga de sincronizar las informaciones referentes a distintos medios, tales como el audio y el vídeo codificados de forma independiente, sí incluye marcas de tiempo de cuando las tramas fueron creadas, que pueden ser empleadas por la aplicación destino para tal efecto.

No obstante, el protocolo RTP no se especifica de forma completa, ya que, intencionadamente, ese grado de libertad permite que sea lo suficientemente flexible como para poder ser incorporado en las aplicaciones sin necesidad de implementarse en una capa separada. Para cada tipo de aplicación existirá un documento, denominado perfil, que define los atributos y/o modificaciones y extensiones RTP. Además, también existirán distintos documentos RFC (*Requests for Comments*) que definen la estructura de la carga útil para acomodar los datos provenientes de la capa de aplicación en la trama RTP.

Hasta ahora hemos descrito aquellos protocolos que son utilizados para la transmisión de los datos de voz, no obstante, las tecnologías VoIP hacen uso de otros protocolos para la señalización y establecimiento de llamadas. Los protocolos más extendidos para llevar a cabo estas tareas son H.323 y SIP (*Session Initiation Protocol*). H.323 es una recomendación [85] desarrollada por la ITU que originalmente se creó como un mecanismo para el transporte de aplicaciones multimedia en redes LANs, pero que, con el auge de las tecnologías VoIP, ha evolucionado rápidamente para dirigir las crecientes necesidades de las redes VoIP. H.323 provee de un grupo de estándares que no sólo están orientados al modelo básico de llamada, sino que además define servicios suplementarios necesarios para dirigir las expectativas de comunicaciones comerciales. Así, H.323 se convirtió en el primer estándar de VoIP en adoptar el estándar RTP, este último desarrollado por la IETF (*Internet Engineering Task Force*), para transportar audio y vídeo sobre redes IP, utilizando para ello una amplia variedad de protocolos desarrollados por la ITU. No obstante, esta diversidad de especificaciones incrementa considerablemente la complejidad de este protocolo. Ésta fue la razón que motivó el origen del protocolo SIP, que fue estandarizado por el IETF mediante el documento RFC 3261 [86]. Los objetivos de partida de este protocolo fueron que estuviera más integrado con las aplicaciones y servicios propios de Internet, y que fuera más simple y sencillo que la recomendación H.323 [87]. En este sentido, el protocolo SIP se vale de las funciones aportadas por otros protocolos, dando éstas por hechas y no volviendo a desarrollarlas, y se centra en el establecimiento, mantenimiento y terminación de sesiones multimedia entre dos o más usuarios. Su funcionamiento se basa en el intercambio de mensajes de texto, siendo un protocolo abierto que permite la generación de nuevos mensajes para dar cabida a nuevos servicios. Estas fueron algunas de las razones que marcaron que el consorcio 3GPP (*3rd Generation Partnership Project*) se decantara por el uso de este protocolo.

4.2.2. Degradaciones en Aplicaciones en Tiempo Real sobre Redes IP

Fundamentalmente, existen tres fuentes de degradación en el entorno IP impuestos por el medio de transmisión: la latencia, la dispersión en el retardo temporal y la pérdida de paquetes. En las aplicaciones de telefonía VoIP, los dos primeros parámetros, asociados con el retardo, imponen fuertes restricciones, ya que retardos superiores a 150 ms producen un diálogo molesto [88]. Por el contrario, en las aplicaciones de reconocimiento esas restricciones no son tan fuertes, ya que no se espera una respuesta inmediata del

servidor. En este caso se emplean *buffers* intermedios de mayor tamaño que los de VoIP, consiguiendo de este modo relajar las condiciones de latencia, siendo valores típicos en torno a 0.5 s [89], y corregir en cierta medida la dispersión del retardo. Desde un punto de vista general, los efectos relacionados con el retardo pueden ser vistos como pérdidas en las aplicaciones en tiempo real. La aplicación implementará un *buffer* que introducirá un cierto retardo sobre los paquetes recibidos, imponiendo un tiempo umbral dentro del cual se esperará la llegada de un cierto paquete. Si la desviación temporal o *jitter* instantáneo es superior a ese tiempo umbral el paquete recibido es descartado, dando por hecho que superado ese tiempo no podrá ser utilizado por la aplicación.

Uno de los posibles orígenes de las pérdidas de paquetes viene dado por las condiciones adversas de las redes de acceso. En principio cabe pensar que no existirán grandes tasas de pérdidas para aquellas redes de acceso soportadas por medios guiados, ya que éstos últimos suelen caracterizarse por altas relaciones señal a ruido (SNR) donde los errores son poco probables. Por contra, las redes inalámbricas pueden originar pérdidas de paquetes cuando presentan una baja SNR. Esto puede deberse a la distancia entre emisor y receptor [90], condicionantes del entorno del canal, como el *fading* debido a la propagación multicamino [91], o interferencias provenientes de otras fuentes emisoras [92].

Además, independientemente del tipo de acceso empleado, se producen pérdidas debidas a los dispositivos de encaminamiento ya que éstos pueden convertirse en cuellos de botella de la red. La figura 4.2 muestra la estructura interna de un dispositivo de encaminamiento (*router*). Podemos distinguir que se encuentra compuesto por un núcleo conmutador de paquetes y un conjunto de memorias tipo FIFO (*First Input First Output*) que almacenan los paquetes de entrada y salida. En la entrada es necesario este sistema de colas para permitir cierta flexibilidad entre la velocidad instantánea de llegada de paquetes y el ritmo de conmutación del *router*, mientras que la adaptación del flujo de paquetes de salida y la velocidad de los enlaces es llevada a cabo por el sistema de colas de salida. En este entorno existen tres puntos problemáticos:

- Desbordamiento de las colas de entrada. Esta situación se produce cuando la velocidad conjunta de los flujos de entrada es superior a la capacidad de proceso de la lógica de conmutación.

- Desbordamiento de las colas de salida. En este caso, la velocidad conjunta de las colas de salida es inferior a la capacidad de conmutación, por lo que a largo plazo se produce un desbordamiento.

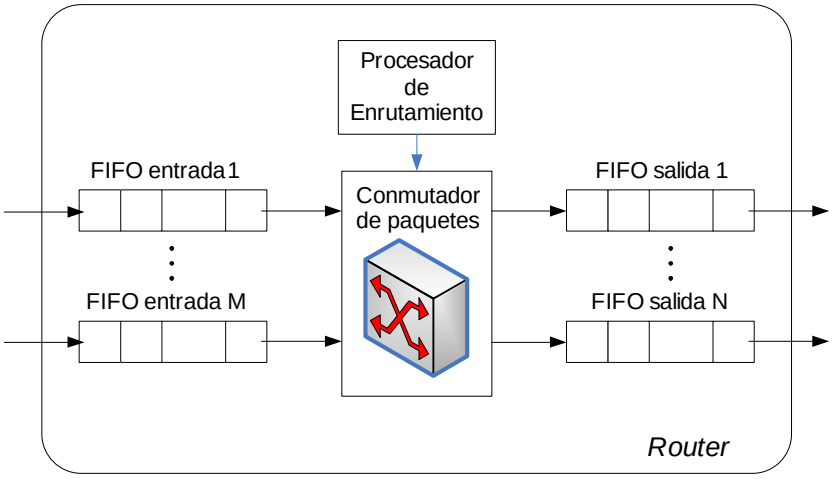


Figura 4.2: Estructura interna de un *router*.

- Conmutación de paquetes de entrada hacia una misma cola de salida. Normalmente, la estructura interna de los *routers* impide el encaminamiento paralelo de dos paquetes hacia una misma cola de salida. Esta situación introduce un cierto retardo en la operación de uno de esos paquetes y puede llevar al *router* hacia un desbordamiento de las FIFOs.

Ante estas adversidades, y siguiendo la simple filosofía de mejor esfuerzo (*best effort*), el paquete que llega a una cola saturada es descartado, siendo responsabilidad de los extremos afrontar esa pérdida. De hecho, la capa de aplicación puede hacer uso de técnicas para minimizar el impacto de los paquetes perdidos. Así pues, existirán técnicas orientadas a la anticipación de las pérdidas, tales como el entremezclado y los códigos FEC (*Forward Error Correction*). No obstante, las aplicaciones incluirán algoritmos de mitigación, tales como los algoritmos PLC (*Packet Loss Concealment*) implementados por los codificadores de voz, que permiten tratar las pérdidas remanentes. De cualquier modo, estás técnicas escapan de los objetivos del presente capítulo, siendo tratadas extensivamente en los siguientes capítulos para el caso de sistemas de reconocimiento remoto NSR.

4.2.3. Modelado de Pérdidas

Las aplicaciones multimedia en tiempo real despertaron un gran interés en la evaluación y modelado de las degradaciones presentes en las redes IP. En 1993 Bolot [93] desarrolló un estudio exhaustivo sobre el retardo y pérdida de paquetes. Para ello, examinó una conexión entre Francia y EE.UU. con el fin de analizar su comportamiento. En este sentido, llevó a

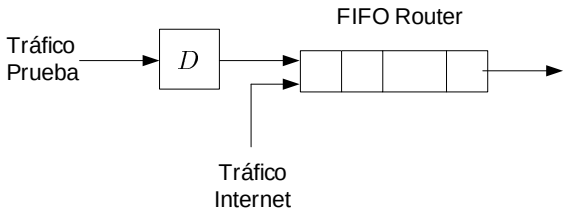


Figura 4.3: Modelo desarrollado por Bolot para la simulación de pérdidas y retardos de paquetes.

cabo medidas de tiempo de ida y vuelta (RTT, *Round Trip Time*) de paquetes UDP enviados a intervalos regulares. Como resultado Bolot concluyó que el tráfico generado podía ser caracterizado mediante un sencillo modelo como el presentado en la figura 4.3. Este modelo consta de un servidor encargado de procesar los paquetes suministrados por una cola, la cual presenta dos flujos diferentes de entrada. El primero, o flujo de prueba, representa el tráfico generado por la aplicación, el cual vendrá dado por paquetes de longitud fija recibidos a una tasa y retardo constante, D, el cual corresponde con el retardo RTT medio. El segundo flujo modela el tráfico de Internet, el cual surge por la superposición de muchos flujos que comparten los recursos con el flujo de prueba. En este caso, los tiempos de recepción y la longitud de los paquetes son variables y determinan el patrón de tráfico presente en la red.

Atendiendo al modelo presentado, la probabilidad de pérdidas o *ulp* (*Unconditional Loss Probability*) se define como,

$$ulp = P(rtt_n = \infty) \tag{4.1}$$

en donde rtt_n representa el tiempo de ida y vuelta (RTT) del paquete n, modelando la pérdida de éste último con un valor de RTT infinito. Cuando el ritmo de generación de paquetes de prueba es muy bajo (D tiene un valor alto), la contribución del flujo de prueba a la cola del servidor es despreciable. Por contra, el aumento de la tasa del tráfico de prueba (valores pequeños de D) produce un incremento del flujo total y, por tanto, adquiere importancia en los momentos de saturación. Otra conclusión importante es que los paquetes tienden a aparecer en ráfagas, ya que si se pierde un cierto paquete n probablemente se pierda el posterior $n+1$. Por consiguiente, es útil conocer la probabilidad de pérdida condicional o *clp* (*Conditional Loss Probability*) que expresa la probabilidad

de que un paquete se pierda dado que se perdió el anterior, o lo que es lo mismo,

$$clp = P(rtt_{n+1} = \infty | rtt_n = \infty) \tag{4.2}$$

Esta probabilidad tiene una dependencia directa con la tasa de generación de paquetes. Si el tráfico de prueba generado aumenta (D disminuye), se hace más probable que una hipotética congestión de la red en un momento t se mantenga en el momento $t+D$.Estudios posteriores [94, 95, 96] han ido verificando que las pérdidas producidas en Internet se dan mayoritariamente en ráfagas. La razón de este hecho puede encontrarse en que la mayoría de las pérdidas son ocasionadas por la congestión de los equipos de encaminamiento, tal y como se expuso en la sección anterior, por lo que si esta situación se da para un paquete probablemente también se de para los siguientes que, aunque no tienen por qué, tienen una alta probabilidad de haber sido enviados a ese mismo nodo. De igual forma, las transmisiones inalámbricas tienden a presentar errores distribuidos en ráfagas debido al *fading*originado por la propagación multicamino [43]. Consecuentemente, las redes de paquetes con soporte inalámbrico originarán también pérdidas distribuidas en ráfagas.

A continuación revisaremos los modelos de pérdidas de paquetes más extendidos atendiendo a su capacidad para adaptarse a trazas reales. Estos modelos nos suministrarán una potente herramienta para obtener un conjunto de condiciones de canal que nos permita explorar de una forma completa y controlada el rendimiento de los sistemas de reconocimiento remotos.

Modelo de Bernoulli

Este modelo considera que las pérdidas de paquetes se producen atendiendo a una secuencia de variables aleatorias $\{X_t\}_{t=1}^{\infty}$ estadísticamente independientes e idénticamente distribuidas. Cada variable aleatoria X_t es binaria y toma valores 0 (paquete recibido) o 1 (paquete perdido), de modo que el modelo queda definido por la probabilidad de pérdidas $r = P(X_t = 1)$, la cual corresponde con la probabilidad ulp. Determinada la tasa de pérdidas r, es posible determinar la probabilidades P_l y $\bar{P}_l$ de recibir o perder, respectivamente, un número l de paquetes consecutivos,

$$P_l = r(1-r)^{l-1} \tag{4.3}$$

$$\bar{P}_l = (1-r)r^{l-1} \tag{4.4}$$

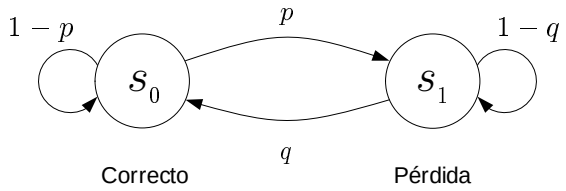


Figura 4.4: Modelo de Gilbert.

A partir de esta expresión se puede derivar fácilmente la duración media de una ráfaga L_{burst} del siguiente modo,

$$L_{burst} = \sum_{l=1}^{\infty} l\bar{P}_l = (1-r)\sum_{l=1}^{\infty} lr^{l-1} = \frac{1}{1-r} \tag{4.5}$$

Es necesario hacer notar que este modelo sólo nos permite ajustar la tasa media de pérdidas, mientras que la duración media de las ráfagas vendrá fijada por la expresión (4.5).

Puesto que este modelo no tiene en cuenta la correlación existente entre paquetes enviados consecutivamente, numerosos autores [94, 97, 98] coinciden en que esta aproximación no modela apropiadamente la aparición de pérdidas en ráfagas. Concretamente, Jiang y Schulzrinne [99] demostraron que el modelo de Bernoulli sobreestima la aparición de pérdidas aisladas, mientras que subestima las probabilidades de aparición de ráfagas más largas.

Modelo de Gilbert

El uso de un modelo de Markov se adapta bien al fenómeno de las pérdidas [100] ya que es capaz de capturar la dependencia temporal entre ellas. El modelo de Gilbert, que es el más simple de los modelos de Markov, se encuentra compuesto por dos únicos estados: recepción correcta (estado s_0) o pérdida de paquete (estado s_1), tal y como se muestra en la figura 4.4. Como hicimos con anterioridad, la pérdida del paquete correspondiente al instante t se modela mediante una variable aleatoria binaria X_t, la cual adoptará el valor 0 cuando se reciba el paquete (estado s_0) y el valor 1 cuando se produzca la pérdida (estado s_1). En este modelo, el estado actual X_t del proceso estocástico sólo depende del valor previo X_{t-1}, de modo que las probabilidades de transición entre estados son las siguientes:

$$p = P[X_t = 1 \,|\, X_{t-1} = 0]$$
$$q = P[X_t = 0 \,|\, X_{t-1} = 1] \tag{4.6}$$

En este caso, las probabilidades de recibir, P_l, y perder, $\bar{P}_l$, una ráfaga de l paquetes consecutivos vienen determinadas por,

$$P_l = p(1-p)^{l-1} \tag{4.7}$$
$$\bar{P}_l = q(1-q)^{l-1} \tag{4.8}$$

Es fácil derivar que la duración media de las ráfagas de pérdidas es,

$$L_{burst} = \sum_{l=1}^{\infty} l \cdot \bar{P}_l = \sum_{l=1}^{\infty} l \cdot q \cdot (1-q)^{l-1} = \frac{1}{q} \tag{4.9}$$

así como las probabilidades ulp y clp, que se obtienen del modelo mediante las siguientes expresiones,

$$ulp = \frac{p}{p+q} \tag{4.10}$$
$$clp = 1 - q \tag{4.11}$$

Concretamente, para adaptar las probabilidades de transición del modelo a las características de una cierta traza, se adaptan los valores L_{burst} y ulp (determinados por las expresiones anteriores) a los valores dados por la muestra de tráfico [99]. Atendiendo a las expresiones anteriores, podemos ver como el modelo de Gilbert dispone de un grado más de libertad que el modelo de Bernoulli, correspondiéndose con éste cuando se introduce la restricción $p + q = 1$.

A pesar de que el modelo de Gilbert mejora las prestaciones del de Bernoulli mediante el uso de dos parámetros, su simplicidad introduce una pérdida de precisión a la hora de ajustar los patrones de pérdidas a las trazas reales [99, 101]. Una posible solución es el modelo denominado como Gilbert-Elliot, donde se introduce un nuevo parámetro p_{el} que determina la probabilidad de perder un paquete en el estado de pérdidas, lo que permite un mejor modelado de los periodos cortos sin pérdidas. Esta modificación se traduce en

la siguiente tasa de pérdidas y duración media de ráfaga,

$$ulp = \frac{p}{p+q} p_{el} \tag{4.12}$$

$$L_{burst} = \frac{1}{1-(1-q)p_{el}} \tag{4.13}$$

Otros Modelos

Para conseguir un modelado más flexible que el obtenido por los modelos de Gilbert y Gilbert-Elliot, es posible considerar una cadena de Markov con más de dos estados. Así, Milner y James [101] proponen el empleo de un modelo de tres estados (véase figura 4.5) que permite el intercalado de paquetes correctamente recibidos dentro de una ráfaga de pérdidas. El modelo queda totalmente determinado por las probabilidades de transición entre estados p, q, r y s, a partir de las cuales se pueden derivar la tasa de pérdidas y la duración media de las ráfagas de pérdidas mediante las siguientes expresiones,

$$ulp = \frac{rp}{r(p+q)+ps} \tag{4.14}$$

$$L_{burst} = \frac{1}{q+s} \tag{4.15}$$

El ajuste de las probabilidades de transición a las características de una determinada traza de paquetes perdidos se lleva a cabo estimando las longitudes medias de los periodos sin pérdidas (N_1) y de los periodos sin pérdidas dentro de los segmentos de pérdidas (N_3), y la evaluación del siguiente conjunto de expresiones,

$$\begin{aligned} p &= \frac{1}{N_1} & r &= \frac{1}{N_3} \\ q &= \frac{p}{r-p}\left[\frac{r}{ulp} - \left(r + \frac{1}{L_{burst}}\right)\right] & s &= \frac{1}{L_{burst}} - q \end{aligned} \tag{4.16}$$

Una alternativa al modelo de tres estados que permite capturar tanto pérdidas consecutivas de corta duración, así como pérdidas con baja probabilidad de aparición, es el modelo de cuatro estados propuesto por la ETSI [102]. Este modelo interconecta a su vez dos modelos de dos estados denominados periodo de interrupción y periodo de ráfaga. El periodo de interrupción se define normalmente a través de la máxima tasa de pérdidas o el mínimo número de paquetes recibidos consecutivamente. A su vez, el periodo de ráfaga

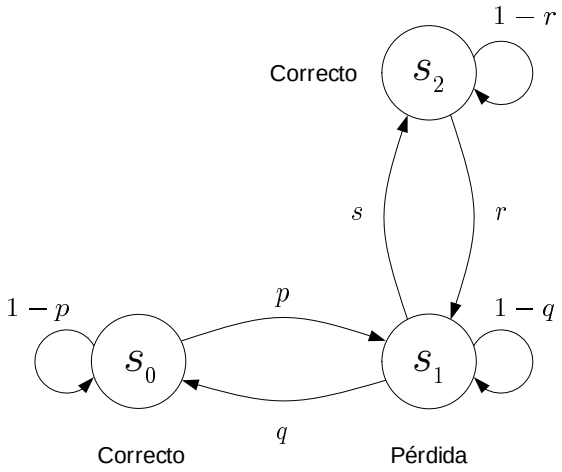


Figura 4.5: Modelo de Markov de tres estados.

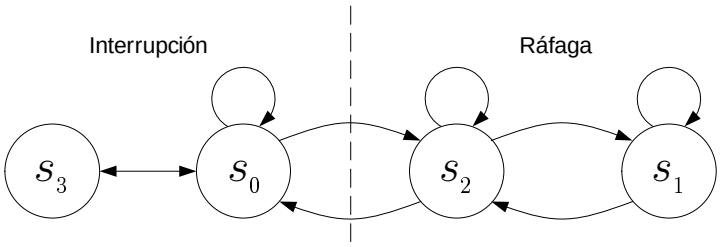


Figura 4.6: Modelo de Markov de 4 estados con periodos de ráfaga e interrupción.

debe comenzar y finalizar con un paquete perdido mientras que el número de paquetes recibidos consecutivamente debe ser menor que un cierto valor. En la figura 4.6 se representa un diagrama de este modelo, correspondiéndose sus estados con las siguientes situaciones: Estado 0, paquete recibido dentro de un periodo de interrupción; Estado 1, paquete recibido dentro de una ráfaga de paquetes perdidos; Estado 2, pérdida dentro de una ráfaga de paquetes perdidos; Estado 3, pérdida aislada en un intervalo de interrupción.

Otra alternativa es utilizar un modelo de Gilbert extendido [98] con $n+1$ estados como el que se muestra en la figura 4.7. Al igual que sucede en el modelo de Gilbert de dos estados, el modelo extendido presenta un estado de recepción correcta (s_0), mientras que el resto de estados s_k ($k = 1, \ldots, n$) representan la pérdida de k paquetes perdidos de forma consecutiva. Así pues, si el estado actual es s_k y se recibe un paquete, se producirá una transición al estado s_0, mientras que si, por contra, se produce una pérdida, se pasará al estado s_{k+1}. Por último, si la ráfaga es mayor de n paquetes, el modelo se mantendrá en el estado s_n. La ventaja de este modelo es que puede adaptarse perfectamente a una traza dada siempre que no tenga ráfagas mayores de n paquetes.

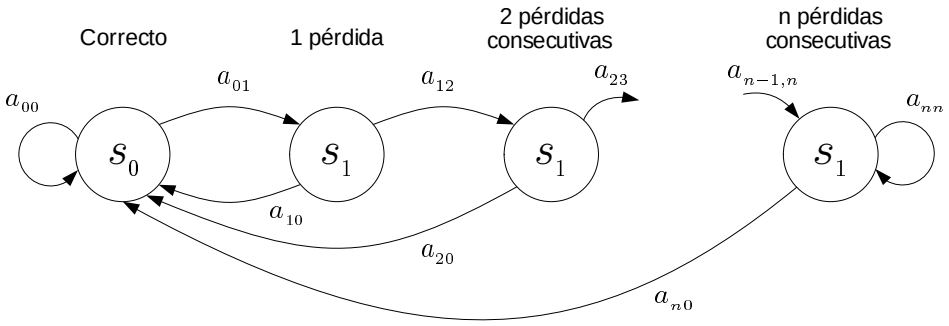


Figura 4.7: Modelo de Gilbert extendido.

Finalmente, sería posible obtener una representación más precisa de las correlaciones presentes en un proceso de pérdida de paquetes utilizando un modelo de Markov de orden superior. En este caso, consideraremos que la variable aleatoria X_t, la cual representa la recepción o pérdida de un paquete en el instante t, no sólo depende de la previa sino que lo hará de las n previas (modelo de Markov de orden n-ésimo), de modo que las probabilidades de transición entre estados responderán a la forma $P(X_t|X_{t-1}, X_{t-2}, \ldots, X_{t-n})$. Puesto que un estado se define a partir de las n previas variables, el modelo constará de un total de 2^n estados. A este respecto, Yajnik *et al.* [94] demostraron que un orden $n = 6$ es suficiente para modelar correctamente las trazas de tráfico reales que capturaron.

4.3. Marco Experimental

Una vez que se han visto las características del canal de transmisión y la forma de modelarlo, podemos pasar a explicar el marco experimental que utilizaremos a lo largo del presente trabajo para la evaluación de los sistemas de reconocimiento remoto NSR. Este marco experimental será el que utilizaremos posteriormente en este capítulo para la obtención de los resultados de referencia. La implementación más directa de un sistema de reconocimiento NSR viene dada por el reconocimiento de voz decodificada. Así pues, los resultados de referencia, o *baseline*, corresponderán con los obtenidos tras llevar a cabo el proceso de codificación y decodificación completo de la señal de voz. Esta primera aproximación nos permite evaluar el impacto de la distorsión de codificación que vendrá determinado por las características del códec empleado.

Para la evaluación de la robustez del sistema NSR frente a errores de canal es necesario introducir un modelo de pérdidas. En principio, la mejor solución sería la utilización de un canal real sobre el que se transmitirían los paquetes de voz codificada, de modo que

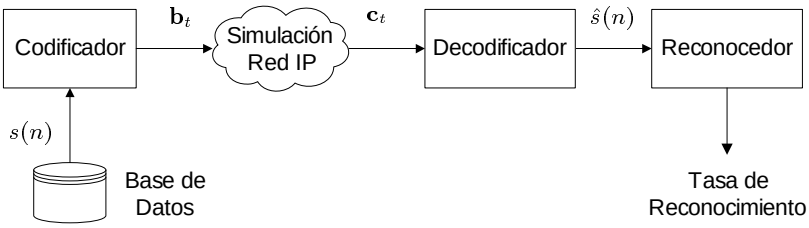


Figura 4.8: Esquema general del marco experimental.

el marco experimental nos permitiera evaluar el impacto de los paquetes perdidos en condiciones reales [103]. No obstante, es inviable obtener a través de este método un rango amplio de condiciones de canal. Una alternativa consiste en simular el comportamiento de pérdidas mediante un modelo, como los explicados con anterioridad, que nos permita obtener un conjunto lo suficientemente extenso de condiciones de canal controlables.

La figura 4.8 muestra el diagrama de bloques del sistema de evaluación utilizado. El rendimiento de un sistema NSR vendrá dado por el comportamiento del códec de voz que emplee. De este modo, para la evaluación de las prestaciones de un sistema NSR se lleva a cabo la codificación de una base de datos de voz. Así, la señal de voz original, $s(n)$, se codifica y segmenta en paquetes, formando el flujo de información a transmitir $\mathbf{b}_t$. Sobre este flujo de paquetes transmitidos se marcan aquellos paquetes perdidos correspondientes a una cierta condición de canal, transformando el flujo de paquetes transmitidos $\mathbf{b}_t$ en $\mathbf{c}_t$. En el lado servidor, el decodificador utiliza el flujo de información $\mathbf{c}_t$ para llevar a cabo la síntesis de la señal de voz $\hat{s}(n)$. Puesto que el flujo $\mathbf{c}_t$ omite la información de los paquetes perdidos, el decodificador utiliza su correspondiente algoritmo PLC (*Packet Loss Concealment*) para realizar la síntesis de voz correspondiente a las pérdidas. Obviamente, a medida que esta aproximación se desvíe de la información original, la voz sintetizada presentará mayor degradación. Finalmente, la voz sintetizada se emplea como entrada de un sistema de reconocedor de voz que se encarga de establecer una tasa de reconocimiento.

4.4. Selección de Codificadores

Antes de comenzar a detallar los distintos elementos del experimento que fueron introducidos en el apartado anterior es necesario seleccionar qué codificadores serán sometidos a prueba. En este punto se presentarán los codecs que fueron seleccionados, las razones que justifican esta selección y se enumerarán sus principios de funcionamiento (para una descripción más detallada de éstos últimos véase el capítulo 3). Además, en el caso de que

el codificador utilizado presente varios modos de funcionamiento se marcarán los modos de operación a evaluar:

- **AMR (Adaptive Multi-Rate).** Codificador basado en el paradigma CELP, típico de las redes de telefonía móvil que presenta un mecanismo de adaptación en función de las degradaciones del canal móvil. Realmente este codec es una agrupación de 8 codificadores con tasas distintas (4.75, 5.15, 5.90, 6.70, 7.40, 7.95, 10.2 y 12.2 kbps), de modo que a medida que la degradación del canal aumenta, se seleccionan tasas de transmisión inferiores incrementando paralelamente la información de redundancia y, por tanto, la protección frente a errores. Concretamente, hace uso de la tecnología ACELP (*Algebraic CELP*) [78] y es un codificador de interés ya que fue seleccionado por el consorcio 3GPP (*3rd Generation Partnership Project*) para la codificación de voz tanto en redes de conmutación de circuitos [104] como en redes de conmutación de paquetes [105]. Además, existe la especificación RFC 3267 [106] que define la estructura de la carga útil del protocolo RTP (*Real Time Protocol*) para este codificador. Los modos seleccionados fueron: 4.75, 7.95 y 12.2 kbps. Se consideró que éstos serían interesantes ya que el modo 4.75 kbps es el que presenta una tasa menor (similar al del estándar DSR introducido en el capítulo anterior), el modo 7.95 kbps corresponde a una tasa intermedia y el modo 12.2 kbps posee la tasa más alta y, por tanto, la mejor calidad perceptual. En este último caso hay que hacer resaltar que este modo se corresponde prácticamente con el antiguo estándar EFR (*Enhanced Full-Rate*) [77] utilizado por GSM. Todos los modos de este codificador utilizan un tamaño de trama de 20 ms utilizando un *look-ahead* (parte de la trama siguiente a la actual) de 5 ms, a excepción del modo 12.2 kbps que no utiliza *look-ahead*. Para que exista compatibilidad entre todos los modos se añade un retardo adicional de 5 ms al modo 12.2 kbps, estableciéndose así un retardo algorítmico de 25 ms.

- **G.729.** Códec estándar desarrollado por la ITU en 1996 [81] y que es de especial interés pues se encuentra, junto con otros códecs de la ITU (G.711, G.723, G.728...), amparado por la recomendación H.323 [85], la cual define los protocolos necesarios para establecer sesiones de comunicación audio-visual sobre redes de conmutación de paquetes. Concretamente, G.729 presenta una tasa de codificación de 8 kbps (aunque posteriormente se han añadido anexos que ofrecen bajo la misma arquitectura tasas de 6.4 y 11.8 kbps) y, según la recomendación H.323, se adecúa perfectamente a aquellos tipos de aplicaciones de audio con un bajo requerimiento de ancho de banda. El principio de funcionamiento de este codificador es CS-ACELP (*Conjugate*

Structure - Algebraic CELP) y, por tanto, se encuentra dentro de la familia de los codificadores CELP. El tamaño de trama utilizado por G.729 es de 10 ms realizando un *look-ahead* de 5 ms, por lo que presenta un retardo algorítmico de 15 ms. Además, el estándar de codificación de voz G.729.1 [107], desarrollado en 2006, utiliza G.729 como esquema de codificación base. G.729.1 es un codificador escalable tanto en tasa de transmisión como en ancho de banda acústico, operando con tasas de transmisión comprendidas entre 8 y 32 kbps. G.729.1 se estructura en capas jerárquicas que vienen definidas por una cierta tasa de codificación. Así, el primer nivel corresponde con una tasa de 8 kbps y viene dado por la especificación G.729. La segunda capa (12 kbps) realza el rendimiento en banda estrecha de la primera. La tercera capa (14 kbps) se trata de una extensión en ancho de banda acústico (8 khz), mientras que el resto de capas se destinan al realce de ésta. Los codificadores desarrollados por la ITU son adaptables a la carga útil de una trama RTP mediante el documento RFC 3551 [108], el cual define el número mínimo de tramas por paquete para cada codificador. En el caso de G.729, el empaquetado mínimo es de 2 tramas por paquete. Igualmente, el codificador G.729.1, a través de la definición de su carga útil de tramas RTP [109], establece que la unidad mínima de empaquetado será de 2 tramas G.729.

- **iLBC.** Este codificador fue diseñado específicamente para la transmisión sobre redes de paquetes IP. Atendiendo a su principio de funcionamiento, es un codificador híbrido basado en el modelo LPC pero que no corresponde al paradigma CELP. En la generación de la excitación no se utiliza un filtro de *pitch*, de modo que se evitan todas las dependencias intertrama con el objetivo de aumentar su robustez frente a la pérdida de paquetes. Como contrapartida iLBC requiere tasas de transmisión mayores que los codificadores CELP, ya que la codificación de la excitación se realiza de forma intratrama a partir de un cierto segmento de la excitación codificado mediante ADPCM. Este codificador fue estandarizado por el organismo IETF mediante el documento RFC 3951 [83], en el cual se puede encontrar una descripción detallada de su funcionamiento. La definición de la carga útil del protocolo RTP para este codificador se encuentra en el documento RFC 3952[110]. Su estandarización, así como la no necesidad de pagar derechos de autor (*royalty free*), han hecho que este códec se difunda rápidamente en aplicaciones VoIP. La actual predilección en su uso por sistemas tan extendidos como *Google Talk* y *Skype* [111], así como su diseño orientado directamente hacia VoIP [112], lo hacen un serio aspirante a convertirse en un códec predominante en este entorno. En este sentido, el organismo

Códec	*Tasa (kbps)*	*Trama (ms)*	*Look-ahead (ms)*	*Empaquetado (tramas/paq)*	*Retardo Alg. (ms)*
AMR 4.75	4.75	20	5	1	25
AMR 7.95	7.95	20	5	1	25
AMR 12.2	12.2	20	0	1	20
G.729	8	10	5	2	25
iLBC 20 ms	15.2	20	0	1	20

Tabla 4.1: Resumen de las características de los codificadores seleccionados.

CableLabs, consorcio formado por los operadores de televisión por cable, incluye este codificador dentro de los recomendados para la codificación de voz en este tipo de redes [113].

De este modo, el conjunto de códecs y modos seleccionado queda de la siguiente manera: AMR {modos 4.75, 7.95 y 12.2 kbps}, G.729 e iLBC {modo 15.2 kbps}. El empaquetado seleccionado para todos estos codificadores es de 20 ms de voz codificada por paquete, ya que es un tamaño permitido para todos los codificadores y que nos permitirá establecer una comparativa entre los resultados obtenidos. La tabla 4.1 resume las características principales de los codificadores utilizados, así como el empaquetado seleccionado y el correspondiente retardo algorítmico derivado. Puesto que cada uno de estos codificadores incluye su propio algoritmo de mitigación frente a pérdidas de paquetes, hay que hacer notar que éstos serán los utilizados durante el estudio objeto de este capítulo.

4.5. Simulación de Condiciones de Pérdidas

En el apartado 4.2.2 vimos como las principales degradaciones presentes en las redes IP vienen dadas por los retardos y la pérdida de paquetes. Posteriormente, concluíamos que, bajo la perspectiva de una aplicación en tiempo real con *buffer* de espera, cualquier tipo de degradación se traduce en una pérdida de información, es decir, en la prestación o no de servicio por las capas inferiores en un intervalo de tiempo determinado por las características de la aplicación.

Bajo esta perspectiva, el marco experimental nos debe permitir evaluar las prestaciones de los esquemas de codificación y sus respectivos algoritmos de mitigación de pérdidas. Las pérdidas de paquetes tienen un efecto degradante sobre la voz reconstruida, ya que llevan asociadas una ausencia de información. A esto hay que añadir el hecho de que éstas tienden a producirse en ráfagas, cuyos efectos suelen ser más destructivos que los provocados por

pérdidas aisladas. Entonces, es posible identificar dos factores que determinan la influencia de las pérdidas en la reconstrucción de la voz:

- El porcentaje de paquetes perdidos. Cuantos más paquetes son perdidos, mayor es la falta de parámetros para la correcta decodificación de la señal de voz y, por tanto, más veces se tendrá que hacer uso de los algoritmos de mitigación de pérdidas.
- La longitud media de las ráfagas. Cuanto mayores son los segmentos perdidos peor será la reconstrucción suministrada por los algoritmos de mitigación.

Por este motivo, la simulación de las pérdidas de paquetes se lleva a cabo a través de un modelo de Gilbert, como el descrito con anterioridad en la sección 4.2.3, ya que supone un buen compromiso entre simplicidad y capacidad para simular ráfagas de pérdidas. Gracias a las probabilidades de transición p y q (véase figura 4.4) es posible establecer un porcentaje global de pérdidas, P_{loss} ó ulp, y una longitud media de ráfaga, L_{burst}. Por tanto, a partir de las expresiones (4.9) y (4.10) se puede fácilmente derivar que las probabilidades de transición para unas determinadas P_{loss} y L_{burst} vienen dadas por,

$$q = \frac{1}{L_{burst}} \tag{4.17}$$

$$p = \frac{P_{loss}}{L_{burst}(1 - P_{loss})} \tag{4.18}$$

Existen numerosos estudios [93, 94, 95, 99, 114] que han intentado caracterizar las conexiones punto-a-punto llevando a cabo medidas sobre un diverso conjunto de pruebas sobre Internet. Los resultados de estos estudios muestran una substancial variabilidad, dependiendo mucho del tipo de camino considerado, con valores de pérdidas típicos comprendidos entre 0 y 20 %. Además, la incorporación de las nuevas tecnologías inalámbricas abre más el abanico de posibles condiciones de canal. Por este motivo, se establece un conjunto de 20 condiciones de canal a fin de obtener un muestreo amplio del rendimiento de los sistemas bajo diferentes condiciones. Estas condiciones implican un porcentaje global de pérdidas (P_{loss}) desde el 5 al 20 %, en incrementos del 5 %, con longitudes de ráfaga media (L_{burst}) de 1, 2, 3 y 4 paquetes. Por otro lado, el parámetro L_{burst} determinará la longitud media de las ráfagas de pérdidas. En este caso, el modelo de Gilbert sólo permite controlar el valor medio de la distribucion de probabilidad de la longitud de ráfaga. La figura 4.9 muestra las 4 posibles distribuciones de la longitud de ráfagas correspondientes a los valores de L_{burst} seleccionados. Estas distribuciones nos permiten explorar un amplio rango de condiciones de canal, que van desde distribuciones de pérdidas aisladas

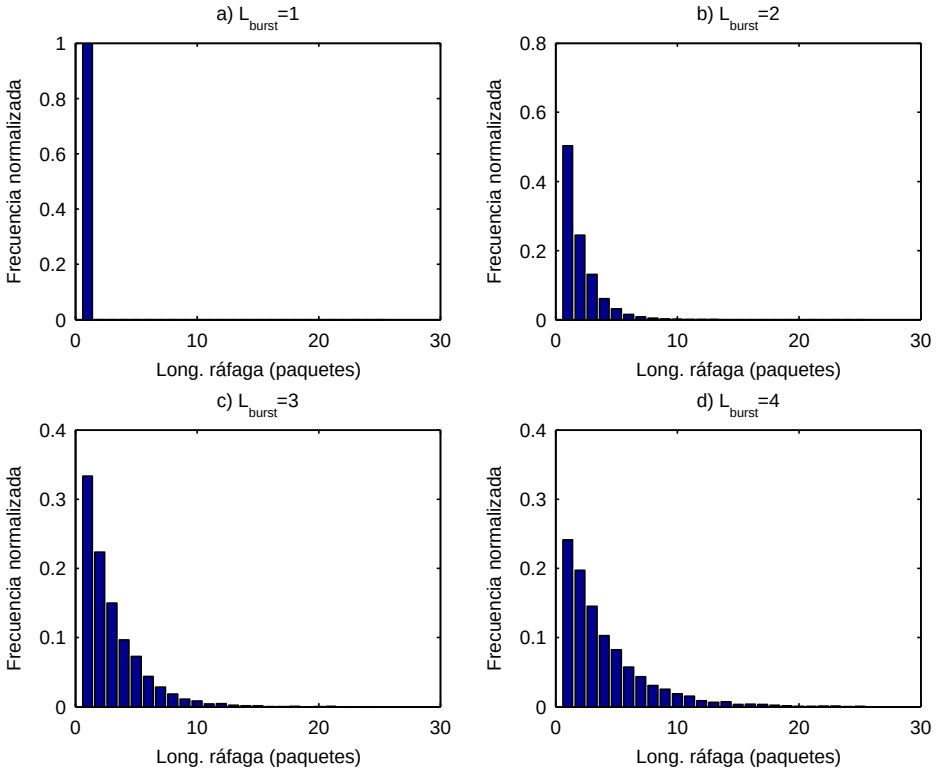


Figura 4.9: Distribuciones de las ráfagas de pérdidas en función de la longitud media de ráfaga en un modelo de Gilbert.

($L_{burst} = 1$) hasta condiciones de canal que presentan, aunque con muy baja probabilidad, pérdidas de 25 paquetes consecutivos ($L_{burst} = 4$).

4.6. Sistema RAH de Referencia

El sistema de reconocimiento utilizado está basado en el marco experimental propuesto por el grupo de trabajo de la ETSI STQ-Aurora DSR Working Group [36], válido para la evaluación de sistemas de reconocimiento remoto en general. A continuación pasamos a detallar las características del sistema RAH de referencia que utilizaremos, no sólo en este capítulo, sino en el resto de la presente tesis para llevar a cabo las pruebas de reconocimiento. A continuación describiremos el proceso de extracción de características, el reconocedor y la base de datos empleados en este trabajo.

4.6.1. Parametrización

El módulo de extracción de características es el utilizado por el estándar DSR FE (*Front-End*) [35], el cual está basado en la representación mediante banco de filtros que estudiamos en la sección 2.3.3. En este caso, al tratarse de una arquitectura NSR, la extracción de características se realiza en el lado servidor sobre la señal de voz decodificada que se encuentra muestreada a 8 kHz (los códecs empleados son de banda estrecha). Posteriormente, sobre la señal de síntesis se aplica una compensación de la componente continua y un filtrado de preénfasis con un factor $\mu = 0{,}97$ (véase sección 2.3.1). Posteriormente, la señal es segmentada en tramas de 25 ms (200 muestras) desplazadas cada 10 ms (dos tramas consecutivas se solapan 15 ms) usando una ventana de Hamming, la cual se adapta a las características espectrales de la señal de voz obteniendo un buen compromiso entre la anchura del lóbulo principal (resolución espectral) y la amplitud de los lóbulos secundarios (fenómeno de *leakage*) [12]. Tras aplicar la ventana se extiende cada trama con zeros (*zero-padding*) hasta 256 puntos sobre los que se lleva a cabo una FFT (*Fast Fourier Transform*) para el cómputo de la magnitud del espectro.

La extracción de características se realiza mediante un banco de 23 filtros triangulares aplicados sobre la magnitud de espectro. Los filtros triangulares se distribuyen uniformemente en escala *mel* desde 64 Hz hasta 4 kHz para introducir un pesado perceptual del espectro. El conjunto de coeficientes resultantes del filtrado se transforma para obtener una representación más apropiada para la tarea de reconocimiento. Así pues, se aplica un análisis cepstral [10] que consiste en realizar la DCT (*Discrete Cosine Transform*) de la salida del banco de filtros en dominio logarítmico, obteniendo un conjunto de 13 coeficientes cepstrales en escala mel (MFCC, *Mel Frequency Cepstral Coefficients*).

El vector de características correspondiente a una trama se forma por 12 coeficientes cepstrales (desde el de orden 1 al de orden 12) junto al logaritmo de la energía, obtenido este último a partir de la señal de voz con preénfasis. Finalmente, este vector se amplía con coeficientes dinámicos delta y delta-delta (véase sección 2.3.5) alcanzando una dimensión final de 39 componentes.

4.6.2. Modelado Acústico y del Lenguaje

La tarea de reconocimiento empleada en este marco experimental consiste en el reconocimiento de voz continua de cadenas de dígitos sin limitación en la longitud de éstas. En este caso, las cadenas de dígitos corresponden a la lengua inglesa con acento americano.

Concretamente, el vocabulario consta de 11 palabras, una por dígito excepto el cero, el cual admite dos pronunciaciones ("zero" y "o").

El reconocedor de voz está basado en el empleo de modelos ocultos de Markov a través de la herramienta HTK [115]. Este paquete software permite la construcción y manipulación de HMMs, estando principalmente orientado a la investigación sobre el reconocimiento de voz.

Atendiendo a que el vocabulario viene dado por 11 palabras que representan a los dígitos, cada una de ellas es modelada mediante un HMM continuo definido por los siguientes parámetros:

- Cada HMM consta de 16 estados (18, si consideramos los nodos nulos de inicio y final).
- Los HMM responden a una topología de izquierda a derecha con un salto máximo permitido de un estado.
- La probabilidad de observación de cada estado se modela mediante una mezcla de 3 gausianas multivariadas de 39 componentes, cuyas matrices de covarianza son diagonales.

Por contra, los silencios y pausas responden a una estructura diferente. En primer lugar, los silencios correspondientes al inicio y final de la frase son modelados mediante un HMM de 3 estados, cuyas probabilidades de observación se corresponden con una mezcla de 6 gausianas por estado. En segundo lugar, a las pausas entre palabras se les asigna el estado intermedio del modelo de silencio de inicio y fin de frase (comparte los mismos parámetros). El entrenamiento se realiza mediante la aplicación del algoritmo de Baum-Welch en varias iteraciones.

4.6.3. Base de Datos

El objetivo principal de este trabajo es la evaluación de la arquitectura NSR sobre redes IP, de ahí que se seleccione un subconjunto de la base de datos Aurora 2 (ETSI STQ-Aurora Proyect Database 2.0) [116]. Para la construcción de Aurora 2 se partió de la base de datos Noisy TIDigits, la cual está formada por frases de dígitos conectados en inglés con acento americano muestreadas originalmente a 20 kHz. Sobre estas frases de partida se aplicaron las siguientes operaciones para la obtención de Aurora 2 [117]:

- Submuestreo y filtrado. La base de datos es submuestreada a 8 kHz utilizando un filtrado "ideal" paso-baja para extraer el contenido espectral entre 0 y 4 kHz. Adicionalmente, se realiza un filtrado, aplicando el estándar G.712 [118], para considerar de forma realista las características en frecuencia de los terminales y equipos.

- Adición de ruido. Con el fin de evaluar las prestaciones en condiciones de ruido acústico una vez realizado el filtrado se añade ruido de forma artificial a diferentes SNRs. De este modo, la base de datos queda formada por 8 posibles escenarios de ruido, en cada uno de los cuales encontramos las condiciones de SNR: *clean* (sin ruido), 20, 15, 10, 5, 0 y -5 dB.

La base de datos se divide en conjuntos de entrenamiento y prueba. El conjunto de entrenamiento se divide a su vez en dos subconjuntos que permiten dos entrenamientos en diferentes condiciones acústicas. Así pues, se puede realizar un entrenamiento limpio (sin ruido) o, por contra, un entrenamiento multicondición. Ambos conjuntos se encuentran formados por 8440 frases conteniendo un total de 55 voces masculinas y 55 voces femeninas, todas de locutores adultos, sobre las que se ha aplicado el filtrado G.712. La diferencia entre ambos conjuntos reside en que el conjunto multicondición, en oposición al limpio, contamina estas frases con diferentes condiciones de ruido acústico (4 tipos de ruido con 5 SNRs).

El conjunto de prueba se subdivide a su vez en tres subconjuntos de prueba: *set A*, *set B* y *set C*. En este caso, los tres subconjuntos parten de 4004 frases, pronunciadas por 52 voces masculinas y 52 voces femeninas, subdivididos en 4 grupos de 1001 frases. El primer subconjunto, o *set A*, contiene los mismos 4 ruidos de ambiente que se utilizaron en el entrenamiento multicondición, de modo que se contamina cada grupo de 1001 frases con un tipo de ruido ambiente a 7 SNRs distintas (*clean*, 20, 15, 10, 5, 0 y -5 dB), resultando en un total de 28028 frases. El subconjunto *set B* contiene 4 ruidos que no fueron considerados durante el entrenamiento multicondición. Finalmente, el subconjunto *set C* parte de dos grupos de 1001 frases y de dos ruidos ambiente que, antes de ser sumados a 7 SNRs diferentes (las mismas que se utilizaron en el *set A* y *set B*), son filtradas utilizando un filtro MIRS (especificación en [118]). Este último subconjunto de prueba nos permite determinar la influencia en el reconocimiento de un sistema de adquisición con respuesta en frecuencia diferente del utilizado durante la fase de entrenamiento.

Puesto que inicialmente el tratamiento del ruido acústico queda fuera del interés de este trabajo, únicamente se han empleado los subconjuntos que incluyen voz limpia, es decir, sin ruido acústico. Así, el entrenamiento es realizado con el conjunto limpio, mientras

que la prueba de reconocimiento se lleva a cabo con el subconjunto de frases *clean* del conjunto *set A*, o lo que es lo mismo 4004 frases de voz sin ruido acústico pronunciadas por un número balanceado de locutores masculinos y femeninos.

4.6.4. Resultados de Reconocimiento

La precisión del reconocedor se evalúa contando el número de errores producidos durante la fase de prueba. Concretamente, 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.

No obstante, la tasa de error se obtiene de forma distinta dependiendo de si se reconocen palabras aisladas o voz continua. En el primer caso, se define la tasa de error de palabra (WER, *Word Error Rate*) como,

$$\text{WER} = \frac{n_e}{n_t} \tag{4.19}$$

donde n_e es el número de errores o palabras mal clasificadas, mientras que n_t hace referencia al número total de palabras de la prueba de reconocimiento.

En tareas de voz continua el reconocimiento se realiza frase a frase, en donde pueden aparecer errores de reconocimiento de tres tipos: inserciones, o palabras adicionales insertadas en la frase reconocida; sustituciones, o palabras sustituidas por otras palabras; borrados, o palabras que no aparecen en la frase reconocida. De este modo, la tasa WER se define como,

$$\text{WER} = \frac{n_i + n_s + n_d}{n_t} \tag{4.20}$$

donde n_i, n_s y n_d hacen referencia al número de errores de inserción, sustitución y borrado, respectivamente. Igualmente, el rendimiento de un reconocedor puede expresarse en términos de la tasa de acierto o precisión de palabra (WAcc, *Word Accuracy*) definida como,

$$\text{WAcc} = 1 - \text{WER} = \frac{n_t - (n_i + n_s + n_d)}{n_t} \tag{4.21}$$

Normalmente estas tasas son presentadas en tantos por ciento, pudiéndose dar el caso de tasas de acierto o WAcc negativas debido a los errores de inserción.

4.6.5. Medidas de Confianza

Como acabamos de ver, la precisión de un sistema reconocedor se evalúa mediante la estimación de una cierta probabilidad de acierto p o, alternativamente, la probabilidad de error $(1-p)$. Estas medidas permiten comparar diferentes sistemas de reconocimiento a fin de constatar cual es el mejor de ellos. Sin embargo, la prueba de reconocimiento no obtiene como resultado la probabilidad de acierto, sino una estima $\hat{p}$ de ésta. Esto implica que los resultados obtenidos han de ser cuidadosamente interpretados puesto que las mejoras observadas pueden no ser estadísticamente significativas.

Los intervalos de confianza nos permiten establecer cómo de fiables son las conclusiones que extraigamos de las pruebas de reconocimiento. Para entender el significado de un intervalo de confianza es necesario considerar que el número de elementos reconocidos correctamente 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 en el valor estimado de la probabilidad de acierto $\hat{p}$ que contendrá con probabilidad $(1-\alpha)$ la probabilidad de acierto. Mediante el teorema del límite central se demuestra que $\hat{p}$ tiende a la distribución normal $\mathcal{N}(0,1)$ y el intervalo de confianza para el valor $\hat{p}$ será,

$$\left[\hat{p} - z_{1-\frac{\alpha}{2}}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}, \hat{p} + z_{1-\frac{\alpha}{2}}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}\right] \tag{4.22}$$

donde $z_{1-\frac{\alpha}{2}}$ es el cuantil $(1-\frac{\alpha}{2})$ de la distribución normal estándar. En la expresión (4.22) se puede apreciar que cuanto mayor es el número de elementos reconocidos, más estrecho será el intervalo de confianza.

Teniendo en cuenta que los subconjuntos *clean* del *set A* de la base de datos Aurora 2 están formados por un conjunto de 13159 palabras, podemos precomputar los intervalos de confianza que obtendremos en las pruebas de reconocimiento llevadas a cabo. La figura 4.10 muestra la amplitud de los intervalos de confianza al 90, 95 y 99 % ofrecidos por el sistema de referencia para distintas tasas de reconocimiento. En este gráfico se muestran tasas de reconocimiento comprendidas entre 74 y 100 %, ya que los resultados obtenidos a lo largo de este trabajo se hayan comprendidos en dicho intervalo. Atendiendo a un intervalo de confianza del 95 %, podemos decir que para tasas de reconocimiento en torno al 95 %, se considerarán mejoras estadísticamente significativas aquellas superiores al cuarto de punto; en el caso de que las tasas de reconocimiento se encuentren en torno al 90 %, las mejoras deberán ser superiores a medio punto; finalmente, las mejoras en torno a un

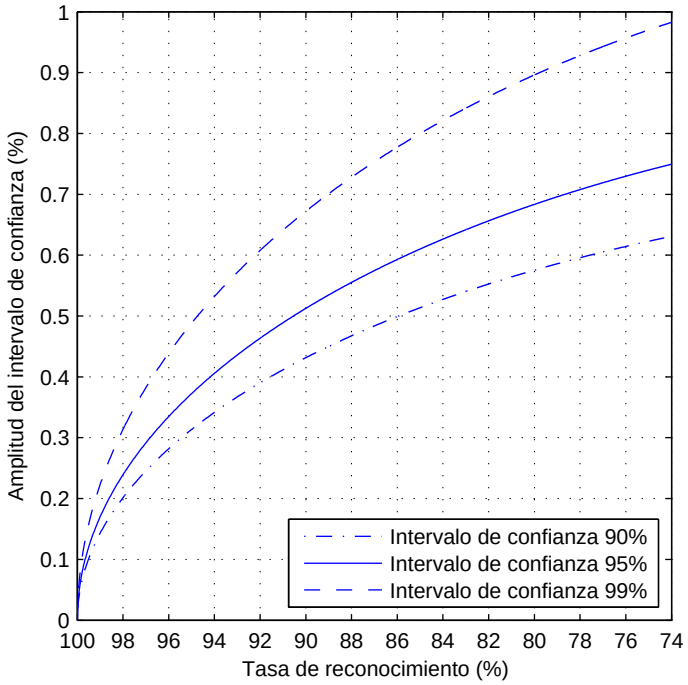


Figura 4.10: Amplitud de los intervalos de confianza al 90, 95 y 99 % en función de la estima de la tasa de acierto para la prueba de reconocimiento definida por los subconjuntos *clean* del test A de Aurora 2.

75 % precisarán de incrementos superiores a tres cuartos de punto. En el resto del presente trabajo, con el fin de no sobrecargar los gráficos y tablas de resultados, no se mostrarán los intervalos de confianza, recomendándose la consulta de esta gráfica para más detalles.

4.7. Resultados de Referencia

En este apartado presentamos los resultados obtenidos para la arquitectura NSR en su implementación más directa, es decir, a partir de voz decodificada. Estos resultados servirán como punto de partida, o *baseline*, para el desarrollo de los posteriores capítulos.

En primer lugar es necesario obtener los resultados obtenidos en el caso ideal, es decir, cuando no se producen pérdidas. Estos resultados se muestran en la tabla 4.2 para cada uno de los codificadores utilizados en este estudio. La primera línea de resultados muestra aquellos obtenidos al no utilizar ningún esquema de codificación, es decir, utilizando la voz original. Las dos últimas columnas de la tabla muestran los resultados de precisión de reconocimiento obtenidos llevando a cabo dos tipos de entrenamientos distintos. El denominado Entrenamiento con Voz Codificada (E.V.C.) se corresponde con realizar las

Códecs	*tasa (kbps)*	*WAcc E.V.C.*	*WAcc E.V.O.*
Ninguno	–	–	99.02
AMR	4.75	98.54	97.82
AMR	7.95	98.68	98.48
G.729	8	98.81	98.64
AMR	12.2	98.70	98.70
iLBC	15.2	98.96	98.92

Tabla 4.2: Resultados WAcc obtenidos para una condición de canal sin pérdidas de paquetes. Las siglas E.V.C. hacen referencia a Entrenamiento con Voz Codificada (mediante el codificador utilizado en la prueba), mientras que las siglas E.V.O. corresponden a Entrenamiento con Voz Original (sin utilizar codificación alguna).

fases de entrenamiento y prueba con las locuciones codificadas con el pertinente códec, mientras que el Entrenamiento con Voz Original (E.V.O.) no utiliza ningún tipo de codificación durante esta fase. Estos resultados de reconocimiento presentan una cierta pérdida de rendimiento debido a la distorsión de codificación. No obstante, la pérdida de rendimiento debida al proceso de codificación/decodificación es sólo considerable para el caso del codificador AMR 4.75 con E.V.O., viéndose notablemente reducida en el caso E.V.C. En el presente trabajo consideraremos que el servidor conoce el esquema de codificación empleado, disponiendo de un entrenamiento adecuado en cada caso (E.V.C.).

Las tablas 4.3, 4.4 y 4.5 muestran los resultados de reconocimiento en condiciones de canal con pérdidas para el codificador AMR con tasas de 4.75, 7.95 y 12.2 kbps, respectivamente. Los correspondientes resultados para los codificadores G.729 e iLBC (modo 20 ms) se muestran en las tablas 4.6 y 4.7. Como referencia superior se incluyen los resultados de reconocimiento obtenidos por la arquitectura DSR estandarizada mediante el documento [35] por el grupo Aurora de la ETSI. Este estándar define tanto la extracción de características (la misma que se detalla en el apartado 4.6.1), como el esquema de compresión y el algoritmo de mitigación de pérdidas, obteniéndose así un conjunto de resultados (tabla 4.8) que establecen una referencia superior para las distintas topologías NSR evaluadas. Para que los resultados DSR sean comparables con los esquemas NSR se empaquetaron 2 tramas DSR por paquete de forma que cada paquete corresponde con 20 ms de señal de voz (véase tabla 4.1). En líneas generales, la arquitectura DSR apenas reduce sus prestaciones ante condiciones de canal con pérdidas aisladas (longitud media de ráfaga unitaria), haciéndose más notable la degradación a medida que las pérdidas se concentran en ráfagas de mayor longitud. Los esquemas NSR evaluados presentan una

menor robustez ante las pérdidas de paquetes, degradando sus prestaciones considerablemente frente a la arquitectura DSR. La justificación de este hecho se halla en las técnicas de codificación y algoritmos de mitigación empleados en la arquitectura NSR.

Los algoritmos de mitigación de pérdidas de los codificadores utilizados se basan en los mismos principios, repetición hacia delante y progresivo apagado (*muting*). En el caso específico de que sólo se pierda un paquete, los codificadores mitigan la pérdida sintetizando una señal con las mismas características espectrales que la correspondiente al paquete anterior (no se realiza *muting*). Puesto que el tratamiento de pérdidas aisladas es similar para todos los codificadores considerados (en este caso repetición para todos), si comparamos la primera columna de todas las tablas de resultados podremos establecer cuál es la robustez intrínseca de cada códec, es decir, cómo se deteriora el rendimiento del codificador con independencia del algoritmo de mitigación empleado. En particular, observamos que el esquema de codificación iLBC decae sólo 2 puntos porcentuales de WAcc para una condición del 20 % de pérdidas aisladas, mientras que los codificadores CELP (AMR y G.729) pierden en torno a 8 puntos independientemente de la tasa de codificación que utilicen. El esquema de codificación CELP se basa en explotar las correlaciones temporales a corto y largo plazo, lo que se traduce en una fuerte dependencia intertrama. Por contra, iLBC evita explotar las correlaciones a largo plazo con el objetivo de eliminar las dependencias intertrama aumentando por tanto la robustez frente a pérdidas. Cuando observamos condiciones de canal con longitud media mayor que una pérdida (canales con ráfagas) vemos cómo las prestaciones de los sistemas NSR se degradan antes para aquellos codificadores que realizan el apagado progresivo con mayor celeridad. Así, como G.729 establece un procedimiento de apagado más rápido que AMR e iLBC, la pérdida de rendimiento es mayor. Independientemente del modo utilizado de AMR, el algoritmo de mitigación de pérdidas es el mismo, de ahí que los resultados de reconocimiento sigan la misma tendencia en todos los modos. Finalmente, iLBC presenta las mejores prestaciones frente a pérdidas pero a costa de una tasa de transmisión mayor. En este último caso, los resultados siguen siendo inferiores a los de DSR ya que el algoritmo de mitigación sigue llevando a cabo un progresivo apagado que da lugar a un mayor número de errores de inserción (silencios artificiales).

Tras analizar los resultados obtenidos a través del marco experimental presentado en este capítulo podemos conluir que:

- La pérdida de rendimiento ocasionada por la distorsión introducida por el proceso de codificación/decodificación se puede reducir notablemente mediante el uso de un

Tasa de Pérdidas	*Long. media ráfaga*			
	1	*2*	*3*	*4*
5 %	97.47	95.72	94.78	94.57
10 %	96.02	92.62	91.30	90.78
15 %	93.76	89.72	87.55	86.38
20 %	90.44	86.12	83.82	82.35

Tabla 4.3: Resultados de precisión de reconocimiento (WAcc) a partir de voz decodificada con AMR 4.75 kbps.

Tasa de Pérdidas	*Long. media ráfaga*			
	1	*2*	*3*	*4*
5 %	97.74	96.07	95.08	95.02
10 %	96.47	93.25	91.63	91.02
15 %	94.64	90.08	88.08	86.83
20 %	91.61	87.13	84.45	82.77

Tabla 4.4: Resultados de precisión de reconocimiento (WAcc) a partir de voz decodificada con AMR 7.95 kbps.

Tasa de Pérdidas	*Long. media ráfaga*			
	1	*2*	*3*	*4*
5 %	97.93	96.46	95.22	94.94
10 %	96.59	93.97	91.95	91.14
15 %	94.51	91.21	88.55	87.07
20 %	91.49	87.87	85.07	83.07

Tabla 4.5: Resultados de precisión de reconocimiento (WAcc) a partir de voz decodificada con AMR 12.2 kbps.

Tasa de Pérdidas	*Long. media ráfaga*			
	1	*2*	*3*	*4*
5 %	98.02	94.46	93.64	93.06
10 %	96.83	89.87	88.55	87.41
15 %	95.15	85.76	83.13	81.28
20 %	93.20	80.80	77.71	75.98

Tabla 4.6: Resultados de precisión de reconocimiento WAcc a partir de voz decodificada con G.729 8 kbps.

Tasa de Pérdidas	*Long. media ráfaga*			
	1	*2*	*3*	*4*
5 %	98.56	97.74	96.79	96.05
10 %	98.19	96.35	94.91	93.07
15 %	97.67	95.13	92.43	89.78
20 %	97.06	93.82	90.34	87.11

Tabla 4.7: Resultados de precisión de reconocimiento (WAcc) a partir de voz decodificada con iLBC 15.2 kbps.

Tasa de Pérdidas	*Long. media ráfaga*			
	1	*2*	*3*	*4*
5 %	99.04	98.86	98.35	97.84
10 %	98.99	98.65	97.74	96.64
15 %	98.96	98.43	97.10	95.21
20 %	98.96	98.08	96.45	94.10

Tabla 4.8: Resultados de precisión de reconocimiento WAcc obtenidos mediante el estándar DSR FE. La precisión de reconocimiento obtenida sin pérdidas es 99.04 %.

conjunto de modelos acústicos obtenidos a partir de voz decodificada.

- La principal fuente de degradación de la arquitectura NSR frente a DSR viene dada por la escasa robustez de los esquemas de codificación frente a pérdidas de paquetes.
- El comportamiento de los codificadores basados en el paradigma CELP introduce fuertes dependencias intertrama que degradan las prestaciones frente al esquema iLBC. No obstante, la codificación iLBC requiere de una tasa de transmisión mayor ya que no explota las correlaciones de la voz a largo plazo.
- Los algoritmos de mitigación de los codificadores no son apropiados para las tareas de reconocimiento, ya que el apagado progresivo deriva en un aumento de los errores de inserción (silencios artificiales).

Capítulo 5

Codificación Robusta frente a Pérdidas

5.1. Introducción

En el capítulo anterior analizamos las posibles degradaciones que puede sufrir la señal de voz en un esquema de reconocimiento NSR. En particular, observamos que las degradaciones en comparación con DSR vienen dadas por la distorsión del proceso de codificación y el nivel de robustez frente a pérdidas. En el primer caso, vimos cómo utilizando modelos de reconocimiento entrenados con voz codificada, se conseguía minimizar el impacto del proceso de codificación. De este modo, el principal problema de la arquitectura NSR viene dado por las pérdidas de paquetes, por lo que se hacen precisas técnicas que prevengan, corrijan y compensen sus efectos. Atendiendo a la clasificación propuesta por Perkins [119], estas técnicas de robustecimiento pueden clasificarse como:

1. Técnicas basadas en el emisor: Requieren la participación del emisor y pueden, a su vez, clasificarse como activas o pasivas. Las primeras se refieren a los esquemas de retransmisión, lo que las hace poco adaptables al entorno de reconocimiento remoto de la voz debido al gran aumento de retardo que originan. Las pasivas actúan como técnicas de prevención de errores, evitando así las pérdidas, o al menos, minimizando el impacto de éstas sobre el rendimiento del sistema. Dentro de estas técnicas podemos encontrar las siguientes:

   a) Técnicas de corrección de errores hacia delante (FEC, *Forward Error Correction*): El codificador introduce información redundante en el flujo de bits trans-

mitido para prevenir las degradaciones producidas por el canal. La redundancia introducida puede ser independiente de la información transmitida (FEC independientes del medio) o relacionada con la misma (FEC dependientes o específicos del medio).

b) Técnicas de Entremezclado: La información es reordenada antes de la transmisión de modo que las pérdidas introducidas por el canal, que normalmente aparecen en ráfagas, se dispersan haciendo más sencilla su mitigación en el receptor. Estas técnicas aumentan el retardo lo que las hace menos propicias para su empleo en sistemas VoIP.

2. Técnicas basadas en el receptor: En este caso, las técnicas de mitigación se aplican en el receptor y no requieren la intervención del emisor. Dentro de estas técnicas podemos distinguir entre tres posibles grupos: interpolación, estimación o técnicas de pérdidas en el reconocedor. La interpolación y la estimación son técnicas de reconstrucción, puesto que su objetivo es el reemplazo de los datos dañados o perdidos. De este modo, el reconocedor emplea los datos reconstruidos en el proceso de reconocimiento. Las técnicas de pérdidas en el reconocedor emplean el potente modelo estadístico utilizado en el proceso de reconocimiento para tratar las pérdidas. En este caso, se llevan a cabo modificaciones en el reconocedor para que éste pueda trabajar con datos perdidos o poco fiables.

Atendiendo a la clasificación anterior en este capítulo se presentan técnicas basadas en el emisor que persiguen el objetivo de conseguir esquemas robustos frente a pérdidas de paquetes. Ya que la principal ventaja de NSR es que no precisa de una aplicación en el terminal cliente específicamente orientada al reconocimiento, no tiene sentido introducir modificaciones en el emisor que estén únicamente orientadas a incrementar el rendimiento de los sistemas de reconocimiento remoto. En este sentido, las modificaciones realizadas sobre las técnicas de codificación deben perseguir un aumento de la calidad perceptual. Con respecto al reconocimiento, partiremos de la hipótesis de que si las técnicas propuestas logran robustecer los esquemas de codificación desde un punto de vista perceptual, entonces también incrementarán el rendimiento de los sistemas de reconocimiento NSR. Esta hipótesis es inicialmente verosímil ya que algunos autores sugieren esquemas basados en reconocimiento de voz como medidas de inteligibilidad [120, 121]. No obstante, verificaremos esta hipótesis llevando a cabo las pertinentes pruebas de reconocimiento.

5.2. Pérdidas de Paquetes y Propagación de Error

El problema de la pérdida de paquetes se ve agravado por el hecho de que la mayoría de los codificadores empleados en la actualidad se basan en el paradigma CELP. Como vimos en el capítulo 3, el paradigma de codificación CELP proporciona una síntesis de voz de alta calidad mediante una tasa de transmisión baja, lo que la hace óptima para su uso en canales con un ancho de banda limitado. Sin embargo, estos codificadores son más vulnerables a la pérdida de tramas debido al uso intensivo de filtros predictivos.

Estos codificadores se basan en el modelo de predicción lineal, en el que la señal de voz sintetizada se obtiene mediante el filtrado de una señal de excitación, $e(n)$, a través de un filtro de predicción lineal (LP, *Linear Prediction*), $H(z) = 1/A(z)$. En los codificadores CELP más empleados en la actualidad, esta excitación se obtiene como la suma de dos señales, a saber, el vector adaptativo, $e_a(n)$, y el vector de código, $e_c(n)$, ambas pesadas por sus correspondientes ganancias, g_a y g_c, es decir,

$$e(n) = g_a e_a(n) + g_c e_c(n) \tag{5.1}$$

Los vectores de código y adaptativo se escogen de un diccionario fijo y otro adaptativo, respectivamente. El diccionario adaptativo (ACB, *Adaptive Code-Book*) tiene como objetivo modelar las correlaciones a largo plazo de la señal de excitación, las cuales están intrínsecamente relacionadas con los pulsos glotales y el periodo de *pitch* de los segmentos sonoros de la voz. Así, las entradas de este diccionario se construyen dinámicamente a partir de las muestras de la excitación previa mediante un filtro LTP (*Long-Term Prediction*), de modo que,

$$e_a(n) = \sum_{k=-(q-1)/2}^{(q+1)/2} p_k e(n - (T_a + k)) \tag{5.2}$$

donde T_a es el retardo y p_k son un conjunto de coeficientes de predicción. Por otro lado, el diccionario fijo (FCB, *Fixed Code-Book*) se encarga de representar la señal residual remanente tras eliminar las correlaciones a largo plazo. Este diccionario también se conoce como diccionario de innovación ya que, en contraste con el adaptativo, no se obtiene a partir de la señal previa. En la figura 5.1 se muestra un diagrama del proceso de síntesis de voz en un decodificador CELP como el descrito. Tal y como explicamos en el capítulo 3, los parámetros de codificación CELP se obtienen mediante un proceso de análisis por síntesis, de modo que el índice del diccionario fijo, las ganancias g_c y g_a y el retardo T_a se seleccionan minimizando el error entre la señal de voz original y la obtenida por el esquema de síntesis

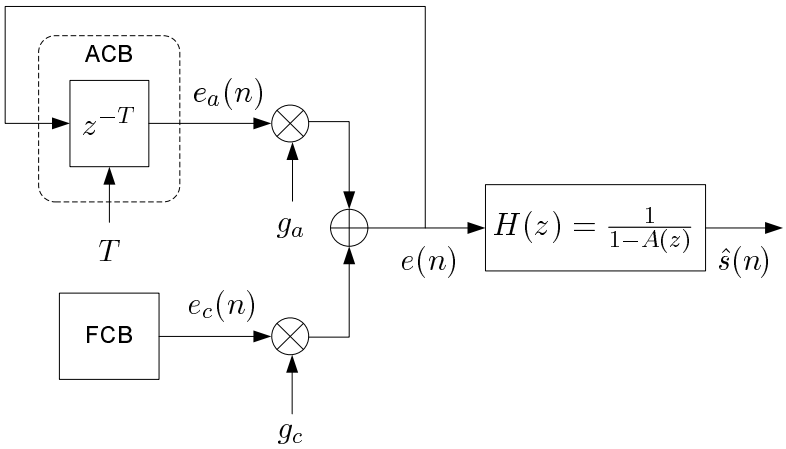


Figura 5.1: Diagrama de decodificación basada en el paradigma CELP.

de la figura 5.1. Además, este procedimiento nos permite integrar ciertas consideraciones perceptuales mediante el pesado de este error por un filtro perceptual $W(z)$.

Aunque la codificación CELP se lleva a cabo trabajando sobre tramas o subtramas, existen ciertas dependencias intertrama durante el proceso de decodificación que hacen vulnerable su rendimiento en redes de paquetes. En particular, el filtro de retardo largo o filtro LTP(véase sección 3.6.1), el cual se corresponde con el diccionario adaptativo explicado con anterioridad, introduce dependencias entre tramas consecutivas, generando una propagación de error en caso de pérdida de paquetes. Este tipo de distorsión aparece justo en las tramas posteriores a una pérdida, cuando el filtro LTP requiere muestras pertenecientes a una trama no recibida. Normalmente, durante la pérdida las correspondientes muestras se generan por algún tipo de algoritmo PLC (*Packet Loss Concealment*), de modo que se produce una desincronización entre el diccionario ACB del codificador y el del decodificador. Esto origina una degradación de la señal de voz sintetizada, la cual se puede propagar varias tramas aunque estas últimas hayan sido recibidas correctamente [122]. La figura 5.2 muestra un ejemplo del impacto producido por una pérdida en la síntesis de voz de un codificador CELP. En el ejemplo se observa que, una vez concluye la pérdida, la degradación se extiende a las tramas recibidas posteriormente.

Tradicionalmente, el interés científico se ha centrado en la pérdida de tramas en sí, dando como fruto algoritmos PLC para la mitigación del segmento perdido. Es en los últimos años cuando se ha empezado a considerar la propagación del error como una fuente importante de degradación dando lugar a numerosas técnicas que minimizan o incluso evitan dicha propagación.

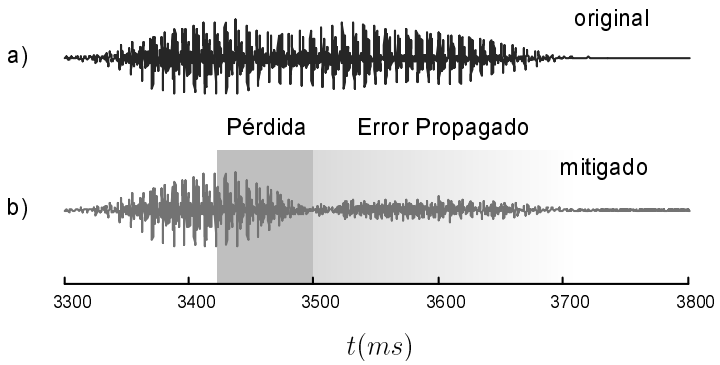


Figura 5.2: Ejemplo del impacto de una pérdida en la síntesis de voz de un codificador CELP (AMR 12.2 kbps): a) síntesis de voz sin pérdidas; b) síntesis de voz con pérdidas aplicando el algoritmo de mitigación integrado por el codificador.

Una de las aproximaciones más sencillas consiste en emplear mejor los paquetes recibidos, tal y como proponen Serizawa e Ito [123]. Estos autores proponen utilizar los paquetes recibidos con un retraso mayor del intervalo de retardo del buffer de reproducción. Aunque estos paquetes no pueden ser utilizados para la síntesis (introducirían un retardo excesivo y, por tanto, serían más molestos incluso que la pérdida), sí pueden ser utilizados para resincronizar la memoria del diccionario ACB reduciendo así la propagación de error. No obstante, esta técnica sólo ofrece solución para aquellos paquetes descartados por el efecto de *jitter* de la red.

Una técnica más sofisticada es la propuesta por Chibani *et al.* [124]. Estos autores proponen modificar la codificación de la excitación en los codificadores CELP restringiendo la ganancia del diccionario adaptativo. De este modo consiguen limitar la dependencia intertrama, forzando que parte del diccionario FCB modele la estructura del *pitch* de la voz. Además, la información relacionada con el *pitch* presente en el diccionario FCB puede ser utilizada en el decodificador para corregir la desincronización del diccionario ACB. El principal inconveniente de estas técnicas reside en que empobrecen ligeramente la calidad perceptual cuando no hay pérdida de paquetes, problema que puede ser resuelto incrementando moderadamente la tasa de bits del codificador.

Las soluciones hasta ahora comentadas tienen la ventaja de que son compatibles con los codificadores estándar basados en CELP. No obstante, algunos autores proponen esquemas de codificación que no atienden exactamente al paradigma CELP. En este sentido, Eksler y Jelinek [125] proponen la sustitución del diccionario ACB por un diccionario de formas de impulsos glotales. Este diccionario no requiere de las muestras previas y, por tanto, evita la dependencia intertrama. No obstante, la eficiencia de estos codificadores es menor que

la de los filtros LTP por lo que su uso se restringe exclusivamente a la primera subtrama en aquellas tramas donde se produce un impulso glotal significativo en la señal residuo LP.

Una solución más drástica es la utilizada en el codificador iLBC [112], donde se propone un esquema de codificación que elimina completamente las dependencias intertrama. En el capítulo anterior vimos cómo este esquema consigue una gran robustez frente a pérdidas de paquetes, aunque mediante un incremento considerable de la tasa de bits. Así, siempre que no se produzcan pérdidas de paquetes, para conseguir una calidad perceptual de voz similar a la de la de los codificadores AMR 12.2 kbps y G.729 8 kbps, iLBC requiere una tasa de 15.2 kbps.

Otra posible opción para reducir el error de propagación es utilizar códigos de corrección de errores hacia delante (códigos FEC, *Forward Error Correction*). Este tipo de técnicas introduce cierta información redundante para minimizar el impacto de las pérdidas. Particularmente, se pueden utilizar códigos FEC específicos del medio, los cuales hacen uso de un cierto conocimiento sobre los datos transmitidos para mejorar el proceso de recuperación o mitigación de pérdidas [126]. Así, la información redundante transmitida puede consistir en una codificación paralela mediante un codificador robusto. Esta es la solución propuesta por Xydeas y Zafeiropoulos en [127], donde el flujo de información redundante consiste en una codificación retardada de un codificador con baja tasa. Así, en caso de pérdidas, la codificación redundante permite resincronizar el diccionario adaptativo. Una aproximación similar es llevada a cabo por Ehara y Yoshida [128], en donde las tramas redundantes son generadas reinicializando el codificador con un estado conocido.

5.3. Evaluación de la Calidad Perceptual

Antes de presentar las diferentes propuestas para combatir la pérdida de rendimiento de los esquemas de codificación frente a la pérdidas de paquetes, es necesario introducir los métodos de evaluación de la calidad perceptual existentes. Como mencionamos en la introducción de este capítulo, la introducción de ciertas modificaciones en el codificador sólo se puede considerar bajo la premisa de que éstas repercutan en un incremento del rendimiento perceptual. Adicionalmente consideraremos que un incremento de la calidad perceptual ante la pérdida de paquetes conllevará una cierta mejora sobre el reconocimiento obtenido a partir de la voz sintetizada. Finalmente, esta hipótesis la verificaremos mediante la realización de las correspondientes pruebas de reconocimiento.

El principal criterio para medir la calidad perceptual de un servicio de comunicación de audio es la calidad subjetiva, es decir, la calidad de servicio que el usuario percibe. Ésta se puede medir a través de métodos de evaluación subjetiva, aunque normalmente estos métodos son costosos y el desarrollo de estas pruebas conlleva una gran cantidad de tiempo. Por estos motivos, surgen métodos de evaluación objetiva que intentan predecir la calidad subjetiva a partir de medidas cuantitativas obtenidas de las señales a evaluar.

5.3.1. Métodos Subjetivos

El método más fiable y extendido para la medición de la calidad perceptual es la puntuación MOS. En este caso, la señal decodificada es reproducida para un número de oyentes (las condiciones de estas pruebas vienen determinadas en [129, 130]) que califican ésta como: (1) mala, (2) pobre, (3) razonable, (4) buena, (5) excelente. Finalmente, la calificación MOS se obtiene como la media de las valoraciones dadas por los oyentes. En la práctica, este método presenta ciertas desventajas, ya que reunir a un número de oyentes expertos como el que especifica la prueba puede resultar difícil y, cuanto menos, costoso.

Otra posibilidad es utilizar la metodología MUSHRA (*MUltiple Stimuli with Hidden Reference and Anchor*) [131]. Este método se basa en una prueba doblemente ciega y multiestímulo con una referencia oculta (original antes de codificar) y uno o varios anclajes (*anchors*) que establecen una referencia inferior, también oculta, correspondiente a codificaciones con baja calidad. La prueba de audición se realiza en una o varias sesiones, que a su vez se subdividen en varios ítems. En cada uno de ellos se presenta la misma señal de audio procesada de varias formas diferentes (estímulos), entre los cuales se encuentran ocultas la señal de referencia (señal sin procesar) y los anclajes. Además, el oyente dispone de la señal de referencia etiquetada como tal. La idea de proporcionar la referencia oculta es poder asegurar la capacidad del oyente de detectar las degradaciones de los estímulos de prueba. El propósito de los anclajes es dar una comparación del material codificado con respecto a niveles de calidad de audio bien conocidos, así como asegurarnos de que las puntuaciones otorgadas a pequeñas degradaciones no se identifiquen con calidades muy bajas. De este modo se consigue que la escala de valoración se acerque lo máximo posible a una escala absoluta. La escala de medida usada es continua entre 0 y 100, aunque se divide en 5 intervalos similares a las calificaciones MOS que van desde mala (puntuaciones entre 0 y 20) a excelente (puntuaciones entre 80 y 100).

La principal ventaja que tiene la metodología MUSHRA sobre la MOS reside en el hecho de que se precisa de un número menor de oyentes expertos para la obtención de

resultados estadísticamente significativos. Esto se debe a que todas las comparaciones que se realizan se llevan a cabo de forma agrupada y a que la escala utilizada (0-100) permite establecer diferencias pequeñas entre calificaciones. No obstante, esta metodología es sólo adecuada para evaluar calidades de audio intermedias, existiendo otras metodologías, como la ABC/HR [132], para la evaluación de pequeñas distorsiones.

5.3.2. Métodos Objetivos

Aunque inicialmente puede parecer que la calidad de la señal de voz puede determinarse mediante la relación señal a ruido, la influencia de las características psicoacústicas, tales como el enmascarado y el agrupamiento de frecuencias no uniforme, hacen que este tipo de mediciones no reflejen bien la calidad percibida subjetivamente por los humanos. Por ejemplo, mientras que la relación señal a ruido de una codificación PCM es superior a la obtenida por los predictores en lazo abierto, la calidad subjetiva de estos últimos es mejor debido a la ponderación espectral que llevan a cabo sobre el ruido de cuantización [133].

En 1998 el organismo ITU desarrolló el algoritmo PSQM (*Perceptual Speech Quality Measure*), descrito en [134], el cual se recomendaba para la evaluación objetiva de códecs vocales. Sin embargo, esta recomendación fue suprimida ya que no tenía debidamente en cuenta los efectos producidos por el filtrado, el retardo variable y las distorsiones cortas localizadas. El algoritmo PESQ (*Perceptual Evaluation of Speech Quality*), desarrollado también por ITU en 2001 [135], fue el encargado de sustituirlo, ya que éste sí que trata dichos efectos mediante la ecualización, la alineación en el tiempo y un nuevo bloque para promediar distorsiones en función del tiempo. Además, en el documento [135] se expresa que el algoritmo PESQ presenta una exactitud aceptable en la estima de la calidad perceptual en un entorno de pérdida de paquetes y mitigación de pérdida de paquetes con códecs CELP. Por todo ello, este algoritmo fue seleccionado para llevar a cabo las medidas de calidad perceptual de la voz necesarias en este estudio.

Un aspecto interesante es que la nota PESQ mantiene una correspondencia con la nota MOS [136], tal y como muestra la figura 5.3. Teniendo en cuenta que la nota PESQ tiene un rango comprendido entre -0.5 y 4.5, la función de correspondencia presenta una característica cuasi-lineal entre ambos parámetros en el rango de notas PESQ entre 2.5 y 4 haciéndolos corresponder aproximadamente con el rango MOS comprendido entre 2 (calidad pobre) y 4 (calidad buena).

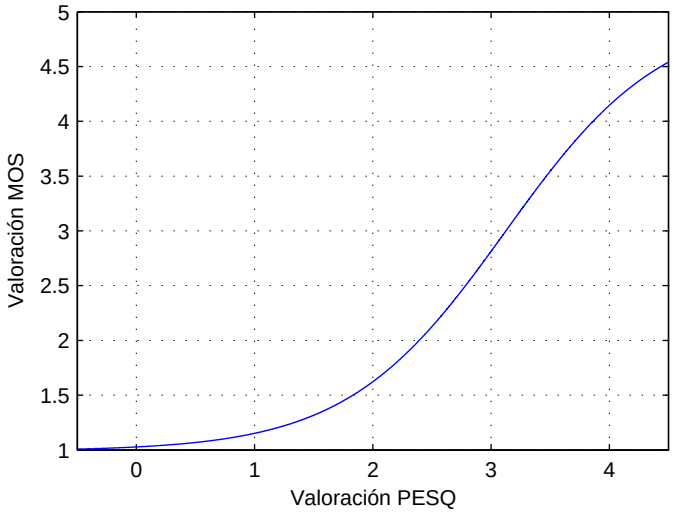


Figura 5.3: Función de correspondencia de las valoraciones realizadas por el algoritmo PESQ y las valoraciones subjetivas MOS.

5.4. Combinación de Tramas

El codificador iLBC fue especialmente concebido para combatir las pérdidas de paquetes. Para ello, iLBC reusa explotar las correlaciones entre tramas adyacentes durante la codificación de la excitación. De este modo elimina las dependencias intertrama, aunque en contrapartida la tasa de codificación es mayor que la de otras técnicas de codificación.

Por contra, los codificadores CELP sí explotan la correlación entre tramas consecutivas reduciendo considerablemente la tasa de transmisión. Como acabamos de ver en la sección anterior, los codificadores CELP se basan en el principio de análisis por síntesis, el cual consiste en escoger aquella señal de excitación que minimiza el error entre la señal sintetizada y la señal objetivo. La señal de excitación surge de la contribución de un diccionario adaptativo y un diccionario fijo. Éste último contiene un número de secuencias de innovación o códigos fijos, mientras que el diccionario adaptativo está compuesto por las muestras anteriores de la excitación. La principal ventaja del diccionario adaptativo es que permite codificar eficientemente los segmentos sonoros de la voz, ya que éstos responden a una estructura cuasi-periódica. No obstante, este tipo de filtros introducen fuertes dependencias intertrama que propagan errores en caso de pérdidas.

Nuestra propuesta consiste en combinar ambos esquemas de codificación, iLBC y CELP (en particular el esquema ACELP), con el fin de obtener un codificador que consiga obtener robustez frente a pérdidas pero haciendo uso de una tasa de bits moderada. La idea se basa en utilizar un esquema combinado de tramas independientes y dependientes

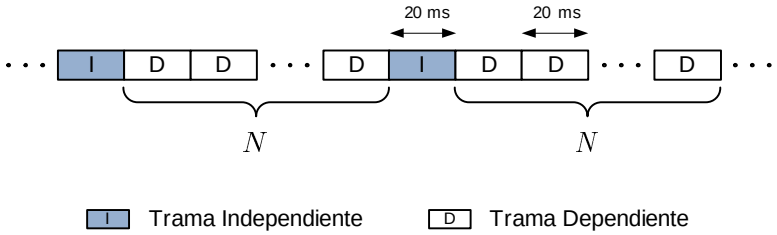


Figura 5.4: Codificación basada en la combinación de tramas iLBC y ACELP.

como el mostrado en la figura 5.4. Así, en caso de pérdidas de paquetes, la propagación de error de las tramas dependientes (tramas ACELP) está limitada por las tramas independientes (tramas iLBC), las cuales actúan como "cortafuegos". De este modo, por un lado conseguimos introducir cierta robustez al esquema de codificación, mientras que por otro conseguimos una tasa de codificación definida por la siguiente expresión,

$$B_N = \frac{B_i + N \cdot B_a}{N + 1} \tag{5.3}$$

donde B_i y B_a hacen referencia a las tasas de bits correspondientes a las tramas iLBC y ACELP, respectivamente, mientras que N se corresponde con el número de tramas ACELP insertadas entre dos tramas iLBC adyacentes (véase figura 5.4).

La reducción de la tasa de codificación viene dada por el número de tramas ACELP insertadas. Así, conseguimos un esquema de codificación variable que nos permite controlar a su vez la robustez frente a pérdidas. Si aumentamos la distancia entre tramas iLBC, el nivel de robustez del sistema decae ya que una separación mayor entre tramas clave supone propagaciones de error más largas. Por contra, si disminuimos la distancia entre tramas iLBC aumenta la robustez frente a pérdidas a costa de un incremento de la tasa de transmisión. Además de conseguir una robustez y tasa variable, este esquema tiene la ventaja de que no modifica el retardo ya que ambos esquemas de codificación utilizan el mismo retardo algorítmico establecido por la longitud de trama (no se emplea *lookahead* en el proceso de codificación).

La figura 5.5 muestra la estructura del decodificador correspondiente al esquema propuesto. En ésta podemos distinguir dos subestructuras que conmutan para la decodificación de la señal de excitación en función del tipo de trama recibida. Así, la única diferencia viene dada por la síntesis de la señal de excitación, mientras que el filtro de síntesis es independiente del tipo de trama recibido. Para evitar posibles sonidos molestos producidos por las discontinuidades entre tramas de diferentes tipos se lleva a cabo un

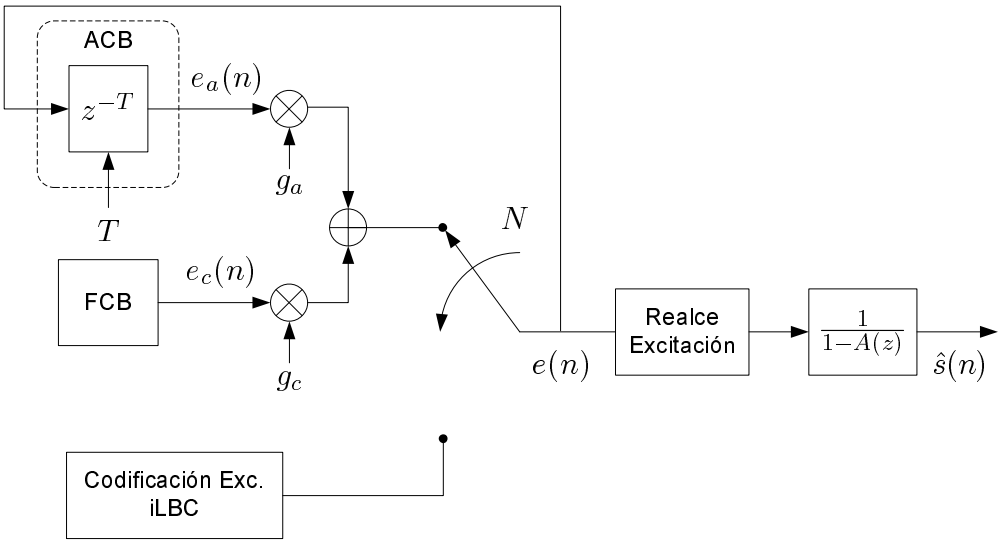


Figura 5.5: Estructura del decodificador para la propuesta basada en la combinación de tramas.

proceso de realce (definido en [83]) sobre la señal de excitación final, el cual introduce un retardo adicional de síntesis de 5 ms. En los siguientes apartados se detallan algunas de las características técnicas del esquema de codificación.

5.4.1. Filtro de Predicción Lineal

El filtro de análisis LP o filtro de síntesis es un módulo común para los dos tipos de tramas de nuestro esquema. Para facilitar la implementación de este sistema se optó por realizar el menor número de modificaciones sobre el esquema iLBC. Así, la propuesta lleva a cabo la codificación del mismo modo que se propone en [83]. Concretamente, se computa un conjunto de coeficientes LPC por trama utilizando una ventana asimétrica de 30 ms de duración centrada en la tercera subtrama de 5 ms. Los coeficientes LPC resultantes (a_k, $k = 1, \ldots, 10$) son transformados en parámetros LSF para llevar a cabo su codificación de forma eficiente (véase sección 3.5.1). No obstante, antes de realizar esta transformación los coeficientes LPC son modificados mediante la siguiente expresión,

$$\tilde{a}_k = \gamma_1^k \cdot a_k \qquad k = 1, \ldots, 10 \tag{5.4}$$

donde $\gamma_1 = 0{,}9025$. Esta operación, denominada expansión de ancho de banda (*bandwidth expansion*) introduce una cierta distorsión sobre el espectro LPC que desencadena dos efectos. En primer lugar, reduce el espacio de cuantificación de los coeficientes, así como

asegura la estabilidad del filtro. En segundo lugar, disminuye el margen dinámico del espectro LPC, acortando en el dominio temporal la duración de la respuesta impulsiva del filtro.

5.4.2. Codificación de la Excitación

Tal y como hemos visto en el apartado anterior, la codificación de los coeficientes LP se lleva a cabo de forma común independientemente del tipo de trama que se trate. Por tanto, sólo resta especificar la codificación de la excitación.

Puesto que el nivel de robustez del sistema recae en la codificación de las tramas independientes, es decir, en las tramas iLBC, se decidió conservar exáctamente la codificación de éstas, adaptando el proceso de codificación de la excitación ACELP para que fuese posible la combinación de tramas sin que se produjesen sonidos molestos al conmutar de un tipo a otro. A continuación pasamos a detallar las características del proceso de codificación ACELP empleado, así como la correspondiente tasa de codificación obtenida.

Filtro de Peso

Durante el proceso de análisis por síntesis se utiliza un filtro de peso como los explicados en la sección 3.6.1. En este caso, el filtro perceptual utilizado responde a la siguiente expresión,

$$W(z) = \frac{1}{1 - A(z/\gamma_2)} \tag{5.5}$$

donde $\gamma_2 = 0{,}422$. Este filtro es utilizado para obtener la señal objetivo utilizada durante el proceso de minimización de error en el cálculo de los parámetros de la excitación.

Diccionarios Adaptativo y Fijo

Como comentamos con anterioridad, los codificadores CELP construyen la señal de excitación del filtro de síntesis mediante dos diccionarios: el diccionario adaptativo y el diccionario fijo. Realmente, el diccionario adaptativo se construye mediante un filtrado LTP de las muestras anteriores de la excitación, dando lugar a la contribución adaptativa $e_a(n)$. Por contra, el diccionario fijo está constituido por una serie de secuencias de innovación (éstas no varían) sobre las que se selecciona la contribución fija $e_c(n)$. En este caso concreto la trama se subdivide en 4 subtramas, optimizando los parámetros de codificación para cada una de ellas. El diccionario fijo seleccionado es del tipo ACELP (véase sección 3.6.2), basándose en el utilizado por el codificador AMR 10.2 kbps [78],

Subtrama	*1*	*2*	*3*	*4*	*Total*
LSFs					20
Retardo LTP	8	5	8	5	26
Código fijo	31	31	31	31	124
Ganancias	8	8	8	8	32
Total					202

Tabla 5.1: Asignación de bits para la codificación de las tramas ACELP (20 ms) en función del número de subtrama.

mientras que el diccionario adaptativo se forma por las 143 muestras previas de la excitación. En particular, la contribución $e_a(n)$ se determina mediante un proceso de búsqueda del retardo LTP en dos fases, siguiendo el procedimiento descrito en [137]. En la primera fase, denominada búsqueda en lazo abierto, se determina una serie de posibles candidatos a partir de los máximos de la autocorrelación de la señal objetivo. En la segunda fase, también conocida como búsqueda en lazo cerrado, se obtiene aquel retardo LTP candidato que logra minimizar el error cuadrático obtenido entre la señal sintetizada y la señal objetivo. El retardo LTP empleado es de tipo fraccional y varía de una subtrama a otra. En particular, la primera y tercera subtrama utilizan una resolución de 1/3 de muestra en el rango $[19+1/3, 84+2/3]$ y una resolución de una muestra en el rango $[85, 143]$. El retardo LTP de las subtramas segunda y cuarta se determina de forma diferencial respecto a T_1 con una resolución de 1/3 de muestra, donde T_1 se corresponde con el entero más cercano al retardo fraccional de la trama previa perteneciente al rango $[20, 143]$.

Ganancias

Puesto que el objetivo de esta propuesta es reducir el impacto de las pérdidas de tramas ACELP, se evita utilizar técnicas predictivas para la codificación de los parámetros ACELP. Así, para la cuantización de las ganancias fija, g_c, y adaptativa, g_a, se diseñó un cuantizador vectorial de 8 bits, el cual obtiene el par $(g_a, \log g_c)$. La búsqueda sobre este diccionario se realiza minimizando el error entre la señal de voz sintetizada y la señal objetivo. Este proceso se realiza 4 veces por trama correspondiendo a 4 subtramas de 5 ms. La tabla 5.1 detalla el número de bits empleado en la codificación de las tramas ACELP, el cual nos conduce a una tasa $B_a = 10{,}2$ kbps.

5.4.3. Resultados Experimentales

Para la obtención de los resultados experimentales PESQ se consideró que no era recomendable utilizar la base de datos Aurora 2, ya que esta base de datos contiene un vocabulario de 11 palabras que introduce ciertas limitaciones fonéticas. En su lugar se utilizó la base de datos TIMIT [138, 139], ya que su corpus es rico fonéticamente y nos permite evaluar correctamente los diferentes esquemas de codificación. Esta base de datos fue desarrollada por Texas Instruments (TI) en colaboración con el Instituto Tecnológico de Massachusetts (MIT) (su nombre surge fusionando las siglas de ambas entidades), y fue ideada para el desarrollo de estudios fonéticos avanzados y sistemas de reconocimiento de voz. TIMIT está compuesta por grabaciones acústicas de banda ancha (la frecuencia de muestreo es de 16 kHz) de 630 locutores de 8 de los principales dialectos del inglés americano. Dentro de la base de datos existe un conjunto de prueba y otro de entrenamiento. Estos conjuntos no comparten ningún locutor y ambos se encuentran balanceados tanto fonéticamente como en el número de dialectos.

Para obtener nuestra base de datos de prueba llevamos a cabo un proceso de submuestreo a 8 kHz del conjunto de prueba original de la base TIMIT. Además, puesto que el algoritmo PESQ recomienda locuciones de duración comprendida entre 8 y 20 s, fue necesario concatenar diferentes locuciones efectuadas por el mismo locutor. Así, las locuciones que componen nuestro conjunto de prueba disponen de una duración media de 14 s, asegurando una duración mínima de 9 s. Además, las locuciones se encuentran balanceadas entre locutores masculinos y femeninos, obteniendo 225 locuciones por grupo (450 en total). Para cada una de las 450 locuciones que componen nuestra base de datos, el algoritmo PESQ establece una puntuación individual. Así, para obtener una valoración global para cada una de las pruebas efectuadas es necesario llevar a cabo un cierto promediado sobre la base de datos. Puesto que las pruebas a realizar consisten en la simulación de pérdidas de ciertos segmentos codificados, cuanto mayor sea el segmento, mayor es la probabilidad de que se produzcan pérdidas sobre él, de ahí que se decidiera ponderar cada una de las puntuaciones PESQ en función de la duración de la locución.

La simulación de las pérdidas de paquetes se realizó en este caso mediante un modelo de Bernouilli, el cual establece trazas de pérdidas aleatorias. Como vimos en la sección 4.2.3, este modelo de canal no refleja completamente la naturaleza de pérdidas en ráfagas de los canales IP. Sin embargo, escogimos este modelo de pérdidas ya que la herramienta de evaluación PESQ no evalúa correctamente la degradación que producen las ráfagas de

pérdidas consecutivas. Este fenómeno fue observado por primera vez por Pennock, estando ampliamente detallado en [140], motivo por el cual el modelo de pérdidas aleatorias es el más utilizado en la literatura [123, 124, 125, 128, 141, 142] cuando se llevan a cabo evaluaciones PESQ. Para obtener un muestreo del rendimiento frente a pérdidas de paquetes lo suficientemente amplio, se simularon 8 condiciones de canal diferentes, caracterizadas por porcentajes de pérdidas de 4 %, 7 %, 10 %, 13 %, 16 %, 18 %, 21 % y 23 %.

La figura 5.6 recoge los resultados obtenidos utilizando el esquema basado en la combinación de tramas. Como se puede comprobar, se ha explorado el rendimiento del sistema intercalando diferentes números de tramas ACELP (N). Tal y como era de esperar, el mejor rendimiento se obtiene con iLBC, esquema que se corresponde con $N = 0$ en nuestra propuesta, es decir, no utilizando tramas ACELP. Por contra, el caso $N = \infty$ obtiene el rendimiento inferior de nuestra propuesta, el cual se corresponde con el esquema de codificación ACELP trabajando a 10.1 kbps (no se emplearían tramas iLBC). Estos casos ($N = 0$ y $N = \infty$) establecen los límites de rendimiento para nuestra propuesta tanto en tasa de codificación como en robustez frente a pérdidas. Así lo demuestran los resultados obtenidos para los valores de N intermedios ($N = 1$, 2, y 3) con tasas de codificación comprendidas entre 12.65 y 11.375 kbps. Además, también se han incluido los resultados correspondientes a los modos 12.2 y 10.2 kbps del estándar de codificación AMR. Se seleccionaron estos modos ya que presentan tasas de bits y retardos similares a los de algunas configuraciones de nuestra propuesta, siendo idóneos para establecer una comparativa justa [142].

Particularmente, las configuraciones con $N = 1$ (12.65 kbps) y $N = 2$ (11.8 kbps) presentan tasas de bits próximas a la de AMR 12.2 kbps. En condiciones ideales de transmisión, es decir, sin pérdidas de paquetes, AMR 12.2 presenta un rendimiento mayor (valoración PESQ de 3.96) que nuestra propuesta (valoraciones PESQ de 3.90 y 3.89 para $N = 1$ y $N = 2$, respectivamente). Sin embargo, esta situación cambia cuando consideramos canales con pérdidas, siendo el rendimiento de AMR 12.2 inferior al de cualquiera de las configuraciones de la propuesta realizada.

Incluso para el caso $N = \infty$ el nivel de robustez frente a pérdidas es superior que la de los modos de AMR. Concretamente, aunque la configuración $N = \infty$ y AMR 10.2 kbps comparten la misma arquitectura ACELP, AMR utiliza técnicas predictivas para cuantizar de forma más eficiente los parámetros de codificación (por ejemplo, las ganancias adaptativa y fija se codifican aplicando filtros predictores). Esto justifica que los resultados para AMR sean ligeramente superiores (PESQ de 3.89) frente a la configuración $N = \infty$ (PESQ de 3.84) en condiciones limpias. No obstante, estas técnicas predictivas no son

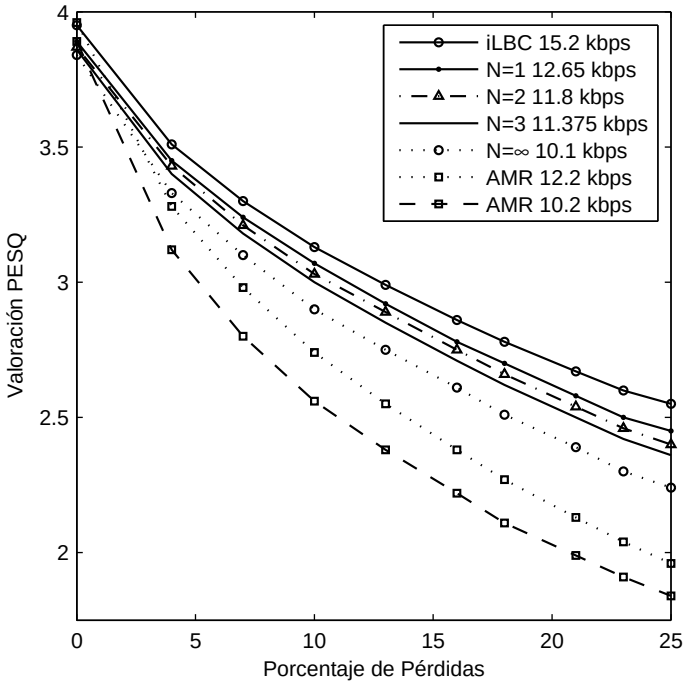


Figura 5.6: Resultados PESQ al aplicar la propuesta de combinación de tramas (N). Como referencias se muestran los resultados obtenidos con AMR 12.2 kbps, AMR 10.2 kbps e iLBC.

adecuadas en un entorno de pérdidas, lo que deriva en una pérdida de rendimiento del esquema AMR.

Hasta ahora, hemos trabajado bajo el supuesto de que un aumento de la calidad perceptual de la voz decodificada repercutiría en un aumento de la tasa de reconocimiento. Para verificar esta hipótesis se realizaron diferentes pruebas de reconocimiento. En este caso, el marco experimental utilizado es el definido en la sección 4.3. La tabla 5.2 recoge los resultados obtenidos aplicando diferentes configuraciones (distintos valores de N) para la combinación de tramas. También se recogen en esta tabla los resultados de las tablas 4.5 y 4.7 (AMR 12.2 kbps e iLBC 15.2 kbps, respectivamente) con el fin de evaluar las mejoras conseguidas. Estableciendo una comparativa directa con los resultados obtenidos por AMR 12.2, vemos cómo el rendimiento de cualquiera de las configuraciones de nuestra propuesta es superior. En particular, las configuraciones $N = 1$ y $N = 2$, las cuales presentan tasas de codificación en torno a 12.2 kbps, presentan un rendimiento superior que el de AMR. Las diferencias son más notables cuando las condiciones de pérdidas presentan una longitud media de ráfaga pequeña (1 ó 2 paquetes). Teniendo en cuenta que los algoritmos PLC empleados por ambos codificadores están basados en los principios

de repetición hacia delante y progresivo apagado, las ráfagas de larga duración derivan en errores de inserción (silencios artificiales), marcando el rendimiento del reconocedor. No obstante, cuando las ráfagas son cortas, la operación de apagado progresivo no tiene lugar y, por tanto, el rendimiento del reconocedor es más sensible a los errores de propagación. En este sentido, la propuesta realizada consigue limitar los efectos de esta propagación consiguiendo así incrementar el nivel de robustez frente a pérdidas. Al igual que sucedía en los resultados PESQ, el codificador ACELP utilizado en nuestra propuesta (se corresponde con la configuración $N = \infty$) es, de partida, más robusto que AMR 12.2 kbps ya que no hace uso de técnicas predictivas para la codificación de parámetros.

La configuraciones de nuestra propuesta con $N \geq 1$ presentan un comportamiento frente a pérdidas similar al de iLBC, pero con la ventaja de que emplean tasas de codificación inferiores. A medida que insertamos un mayor número de tramas ACELP entre dos tramas iLBC adyacentes reducimos el nivel de robustez del esquema. No obstante, este trabajo establece un método sencillo para convertir el esquema de codificación iLBC en un esquema de tasa variable introduciendo una pequeña pérdida de rendimiento en ausencia de pérdidas. Además, en comparación con un esquema de codificación ACELP de tasa variable como AMR, la propuesta presenta un rendimiento claramente superior en presencia de pérdidas de paquetes. En este sentido, se puede obtener un rango mayor de tasas si se utilizan menos bits en la codificación de las tramas ACELP.

5.5. Técnicas FEC basadas en Multipulso

Una posible solución para el problema de la propagación de error consiste en enviar una copia del diccionario adaptativo al receptor. En otras palabras, esta solución se corresponde con un código FEC que estaría formado por las muestras de la excitación anterior a cada trama. En principio, esta aproximación es un tanto descabellada puesto que rompe por completo el paradigma CELP y conllevaría un aumento de la tasa de bits desorbitado. No obstante, podemos partir de esta idea para posteriormente buscar un esquema más refinado. El histograma de la figura 5.7 representa la frecuencia de uso de las muestras de la trama anterior para la síntesis de la trama actual (a través del filtrado LTP). Este gráfico fue obtenido utilizando el codificador AMR 12.2 kbps sobre toda la base de datos TIMIT [138], de modo que la muestra con índice -160 se corresponde con la primera muestra de la trama anterior, mientras que la -1 se corresponde con la última. No obstante, el retardo LTP permitido por el codificador AMR debe estar comprendido en el intervalo -146 a -19, motivo que justifica que la frecuencia de uso del intervalo -160 a

	Tasa de Pérdidas	*Long. media ráfaga*			
		1	2	3	4
iLBC 15.2 kbps	*5 %*	98.56	97.74	96.79	96.05
	10 %	98.19	96.35	94.91	93.07
	15 %	97.67	95.13	92.43	89.78
	20 %	97.06	93.82	90.34	87.11
$N = 1$ *12.65 kbps*	*5 %*	98.18	97.33	96.35	95.76
	10 %	97.49	95.51	94.13	92.60
	15 %	96.51	94.00	91.28	89.04
	20 %	95.46	92.42	88.90	85.98
$N = 2$ *11.8 kbps*	*5 %*	98.12	97.21	96.23	95.56
	10 %	97.28	95.32	93.93	92.13
	15 %	96.13	93.52	90.79	88.36
	20 %	94.92	91.47	87.98	84.83
$N = 3$ *11.375 kbps*	*5 %*	98.05	97.09	96.13	95.43
	10 %	97.19	95.10	93.66	91.92
	15 %	96.01	93.22	90.35	87.98
	20 %	94.58	91.49	87.69	84.43
$N = \infty$ *10.1 kbps*	*5 %*	97.99	96.94	95.90	95.26
	10 %	96.71	94.72	93.11	91.58
	15 %	95.41	92.54	89.75	87.46
	20 %	93.71	90.29	87.03	83.97
AMR 12.2 kbps	*5 %*	97.93	96.46	95.22	94.94
	10 %	96.59	93.97	91.95	91.14
	15 %	94.51	91.21	88.55	87.07
	20 %	91.49	87.87	85.07	83.07

Tabla 5.2: Resultados WAcc a partir de voz decodificada empleando la propuesta basada en la combinación de tramas con N =1, 2, 3 e ∞. También se incluyen los resultados obtenidos con AMR 12.2 kbps e iLBC 15.2 kbps (N =0) como referencias inferior y superior, respectivamente.

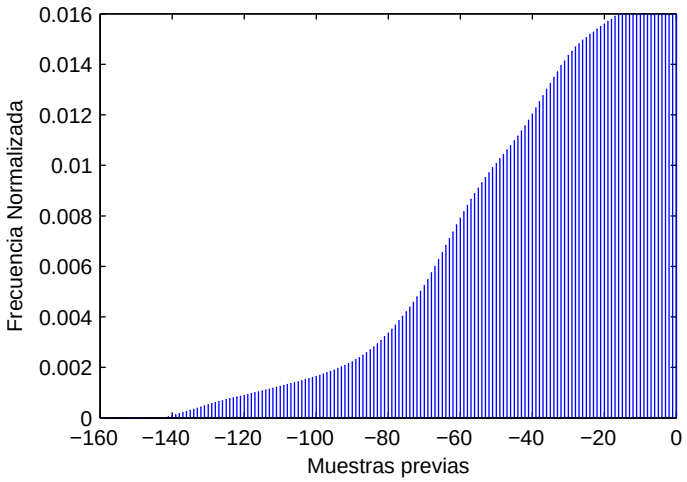


Figura 5.7: Histograma de la frecuencia de uso de las muestras de la trama anterior para la síntesis de la trama actual en el codificador AMR 12.2 kbps sobre toda la base de datos TIMIT.

-144 sea nula. Como podemos comprobar, a medida que nos distanciamos de la trama actual la frecuencia de uso disminuye. Por ejemplo, de este histograma podemos determinar que $P(-\infty < n < 32) \approx 0,07$, donde $P(n)$ es la probabilidad de que la muestra con índice n sea utilizada para la síntesis de la siguiente trama. Por tanto, un esquema FEC que enviara las 128 muestras previas permitiría, aproximadamente, la resincronización del 93 % del diccionario adaptativo. En la práctica este esquema sería inviable, ya que aunque todas estas muestras se sometieran a una fuerte cuantización, el incremento de la tasa de transmisión sería inasequible.

En lugar de llevar a cabo la codificación de todas las muestras de la trama anterior, nuestra propuesta consiste en codificar sólo las más representativas mediante un esquema multipulso como el introducido en la sección 3.6.1.Utilizaremos este esquema para la codificación de las muestras previas con un número reducido de bits, que será enviado junto al flujo de datos del codificador. En principio, partimos de la hipótesis de que esta representación multipulso nos permitirá la resincronización parcial del diccionario adaptativo en la trama posterior a una pérdida, hipótesis que verificaremos más adelante mediante resultados experimentales.

Como ya explicamos en el capítulo 3, el esquema de codificación multipulso construye la señal de excitación $\hat{e}(n)$ mediante una serie de L pulsos dados por unas ciertas amplitudes b_l y posiciones n_l $(l = 0, \ldots, L-1)$, obteniendo ésta tal y como indica la expresión (3.12).

Las posiciones y amplitudes de los pulsos se determinan mediante un procedimiento de mínimo error cuadrático medio (criterio LSE, *Least Square Error*) entre la señal objetivo

y la señal sintetizada. El error cuadrático se define como,

$$E = \sum_{n=0}^{N-1} (s(n) - \hat{s}(n))^2 = \sum_{n=0}^{N-1} (s(n) - h(k) * \hat{e}(n-k))^2 \tag{5.6}$$

donde $s(n)$ es la señal original, $\hat{s}(n)$ es la señal sintetizada, $h(n)$ es la respuesta impulsiva del filtro LP y $\hat{e}(n)$ la excitación codificada. Tal y como vimos en la sección 3.6.1, normalmente el error suele ponderarse mediante un filtro de peso con respuesta impulsiva $w(n)$, de modo que,

$$\begin{aligned} E_w &= \sum_{n=0}^{N-1} (w(n) * (s(n) - \hat{s}(n)))^2 \\ &= \sum_{n=0}^{N-1} (w(n) * s(n) - w(n) * h(k) * \hat{e}(n-k))^2 \\ &= \sum_{n=0}^{N-1} (s_w(n) - h_w(k) * \hat{e}(n-k))^2 \end{aligned} \tag{5.7}$$

Esta última expresión es idéntica a (5.6), excepto para la respuesta impulsiva del filtro LP $h_w(n)$, y la señal objetivo $s_w(n)$, las cuales ahora se encuentran convolucionadas con $w(n)$. Es necesario hacer notar que estas señales ya se encuentran disponibles durante la codificación CELP puesto que ésta se basa en el mismo principio LSE.

Si reemplazamos ahora la ecuación (3.12) en (5.7), obtenemos la siguiente expresión,

$$E_w = \sum_{n=0}^{N-1} (s_w(n) - \sum_{l=0}^{L-1} b_l h_w(n - n_l))^2 \tag{5.8}$$

Suponiendo que la posición de los pulsos es conocida, las amplitudes b_l óptimas, atendiendo al criterio LSE, se pueden determinar de la siguiente manera [143],

$$\mathbf{b}^* = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{s} \tag{5.9}$$

donde $\mathbf{b} = [b_0, b_1, \ldots, b_{L-1}]^T$, $\mathbf{s} = [s_w(0), s_w(1), \ldots, s_w(N-1)]^T$ y $\mathbf{H}$ es una matriz $N \times L$

que se corresponde con

$$\mathbf{H} = \begin{bmatrix} h_w(0-n_0) & h_w(0-n_1) & \cdots & h_w(0-n_{L-1}) \\ h_w(1-n_0) & h_w(1-n_1) & \cdots & h_w(1-n_{L-1}) \\ \vdots & \vdots & \ddots & \vdots \\ h_w(N-1-n_0) & h_w(N-1-n_1) & \cdots & h_w(N-1-n_{L-1}) \end{bmatrix} \tag{5.10}$$

Normalmente, la notación más común se obtiene llevando a cabo la siguiente reordenación e identificación de los términos de la ecuación (5.9),

$$\begin{aligned} (\mathbf{H}^T\mathbf{H})\mathbf{b}^* &= \mathbf{H}^T\mathbf{s} \\ \mathbf{\Phi}\mathbf{b}^* &= \mathbf{c} \end{aligned} \tag{5.11}$$

con $\mathbf{\Phi} = \mathbf{H}^T\mathbf{H}$ y $\mathbf{c} = \mathbf{H}^T\mathbf{s}$. A partir de la expresión (5.11) podemos derivar las siguientes ecuaciones,

$$\sum_{k=0}^{L-1} b_k \phi_{n_k,n_j} = c_{n_j} \qquad 0 \leq j \leq L-1 \tag{5.12}$$

donde,

$$\begin{aligned} \phi_{n_k,n_j} &= \mathbf{\Phi}[k,j] = \sum_{n=0}^{N-1} h_w(n-n_k)h_w(n-n_j) \\ c_{n_j} &= \mathbf{c}[j] = \sum_{n=0}^{N-1} s_w(n)h_w(n-n_j) \end{aligned} \tag{5.13}$$

Atendiendo a estas últimas expresiones, los términos ϕ_{n_k,n_j}son normalmente aproximados mediante la función de autocorrelación de la respuesta impulsiva LP, $R_{hh}(n_k - n_j)$, mientras que c_{n_j}se hace corresponder con la correlación cruzada entre la respuesta impulsiva LP y la señal objetivo.

Finalmente, el error LSE para una determinada combinación de posiciones de pulsos con sus amplitudes óptimas se obtiene como,

$$E_w^* = \mathbf{s}^T[\mathbf{I} - \mathbf{H}(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T]\mathbf{s} \tag{5.14}$$

En principio, el modo óptimo de obtener las posiciones y amplitudes de los pulsos consistiría en probar todas las combinaciones de L pulsos en N posiciones y seleccionar

Tec. Recup.	*Condiciones de Pérdidas*							
Error Prop.	*4 %*	*7 %*	*10 %*	*13 %*	*16 %*	*18 %*	*21 %*	*23 %*
AMR 12.2	3.28	2.98	2.74	2.55	2.38	2.28	2.13	2.05
Rec. Ideal	3.49	3.24	3.05	2.90	2.76	2.68	2.57	2.50
MP $L = 1$	3.45	3.19	3.00	2.84	2.70	2.62	2.51	2.44
MP $L = 5$	3.47	3.22	3.04	2.88	2.75	2.67	2.56	2.49

Tabla 5.3: Resultados PESQ en condiciones de pérdidas obtenidos para la voz decodificada utilizando AMR 12.2 kbps frente a diversas técnicas de recuperación del ACB: recuperación ideal (Rec. ideal) y recuperación multipulso (MP) con L =1 y 5 pulsos por subtrama.

aquellas que consigan minimizar la expresión (5.14). Evidentemente, este procedimiento es inviable desde un punto de vista computacional ya que existen $N!/L!(N-1)!$ posibles combinaciones. Por ello, en la práctica, este procedimiento se realiza mediante un algoritmo subóptimo como el propuesto por Singhal y Atal en [143]. Este algoritmo iterativo busca en cada paso la posición y amplitud óptimas de un único pulso, de modo que en el siguiente paso este pulso se considera fijo sustrayendo su contribución de la señal objetivo. Una posible mejora de este algoritmo consiste en recomputar todas las amplitudes óptimas en la última iteración. Puesto que durante este procesado ϕ_{n_k,n_j} y c_j se utilizan intensivamente, antes de iniciar el algoritmo de optimización normalmente se precomputan y almacenan para todas las posibles posiciones ($0 \leq k,\, j \leq N-1$).

Para verificar la hipótesis de partida de esta sección, es decir, que la resincronización del diccionario adaptativo se puede realizar utilizando una codificación multipulso de la excitación previa, se llevaron a cabo un conjunto de pruebas perceptuales bajo el mismo marco experimental definido en la sección 5.4.3. En este caso, utilizamos el codificador AMR 12.2 kbps y aplicamos la técnica multipulso a nivel de subtrama (4 subtramas por trama), tal y como AMR trabaja. Así, cada una de las subtramas de excitación obtiene una codificación paralela mediante L pulsos que se transmitiría junto a la siguiente trama AMR. De este modo, en caso de pérdida el decodificador contará con una codificación gruesa de la excitación anterior que utilizará como diccionario adaptativo.

La tabla 5.3 refleja los resultados PESQ obtenidos utilizando el esquema multipulso con $L = 1$ (MP $L = 1$) y $L = 5$ (MP $L = 5$) pulsos por subtrama. Además, también se muestra el resultado obtenido por el codificador (nota PESQ de 4.01 en condición libre de pérdidas), y el caso ideal de recuperación ACB, es decir, cuando se dispone de las muestras exactas de la excitación previa. De los resultados obtenidos vemos cómo la hipótesis de partida se verifica, ya que utilizando un único pulso por subtrama (4 pulsos por trama)

se consigue una notable mejoría, mientras que utilizando 5 pulsos por subtrama (20 por trama) conseguimos un rendimiento similar al caso ideal.

5.5.1. FEC basados en Multipulso-LTP

Hasta ahora hemos descartado el empleo de los parámetros CELP de la trama actual. No obstante, estos parámetros pueden ser relevantes en el proceso de resincronización ACB, así como podrían ser potencialmente útiles a la hora de reducir la información colateral a transmitir.

En este apartado proponemos una modificación sobre el esquema multipulso que nos permitirá integrar los parámetros CELP de la trama actual en el proceso de optimización de los pulsos. Al igual que antes, cuando se produzca una pérdida, la codificación multipulso se utilizará para actualizar el diccionario adaptativo evitando así la propagación de error. Sin embargo, a diferencia de la codificación multipulso explicada con anterioridad, ahora la descripción multipulso se optimizará minimizando el error perceptual con respecto a la trama actual (en el caso anterior se realizaba respecto a la trama previa). Para llevar esto a cabo, consideraremos las muestras de la trama previa como una memoria donde colocaremos un número P de pulsos. Posteriormente, en la síntesis de la excitación CELP, estos pulsos son transformados por el filtro LTP, escalados por su ganancia adaptativa y sumados al correspondiente código de innovación (ponderado este último por la ganancia de código). Finalmente, la señal de voz sintetizada se obtiene llevando a cabo el filtrado LP de la excitación, por lo que dependiendo de las amplitudes y posiciones de los pulsos insertados en la memoria obtendremos diferentes síntesis de voz. Así pues, nuestro objetivo será determinar los pulsos óptimos bajo un criterio LSE.

Eliminación de la Contribución Fija

Puesto que la contribución del diccionario fijo es conocida y no tiene dependencias de la trama anterior, podemos simplificar el problema eliminando esta contribución de la señal objetivo. Con este fin, aplicamos el principio de superposición y descomponemos la señal de excitación como la suma de dos contribuciones:

- Excitación sin memoria (ZSE, *zero state excitation*), $\hat{e}_{zs}(n)$. En este caso suponemos que las muestras antes de la trama actual son cero, es decir, que no existen pulsos en la memoria. Sólo se considera el código de innovación.

- Excitación sin entrada (ZIE, *zero input excitation*), $\hat{e}_{zi}(n)$. Esta señal se obtiene considerando que el vector de código de innovación es nulo para la trama actual. Sólo se usan las muestras de la memoria, las cuales se tomarán con el correspondiente T_a y pesadas a su vez por la ganancia g_a.

Como disponemos de los parámetros de la trama actual, podemos determinar la contribución de ZSE (tras llevar a cabo su filtrado LP) y eliminar esta parte de la señal objetivo. De este modo, el error cuadrático a minimizar se puede expresar en función de ZIE como,

$$\begin{aligned} E_w &= \sum_{n=0}^{N-1} \left(s_w(n) - h_w(k) * \hat{e}(n-k)\right)^2 \\ &= \sum_{n=0}^{N-1} \left(s_w(n) - h_w(k) * (\hat{e}_{zs}(n-k) + \hat{e}_{zi}(n-k)\right)^2 \\ &= \sum_{n=0}^{N-1} \left(s_w(n) - \hat{s}_{zs}(n) - h_w(k) * \hat{e}_{zi}(n-k)\right)^2 \\ &= \sum_{n=0}^{N-1} \left(s_{zi}(n) - h_w(k) * \hat{e}_{zi}(n-k)\right)^2 \end{aligned} \tag{5.15}$$

donde $\hat{e}_{zs}(n)$ es la ZSE, $\hat{s}_{zs}(n)$ es su respuesta LP, $\hat{e}_{zi}(n)$ es la ZIE y $s_{zi}(n) = s_w(n) - \hat{s}_{zs}(n)$ será la nueva señal objetivo. De este modo, simplificamos el problema ya que nuestro objetivo será determinar aquella $\hat{e}_{zi}(n)$, excitación que no depende del vector de código, que consigue minimizar el error respecto a la nueva señal objetivo $s_{zi}(n)$.

Integración de Filtro LTP no Fraccional

Tras eliminar la contribución ZSE y obtener la nueva señal objetivo $s_{zi}(n)$, la señal de excitación se simplifica a una recursión dada por las contribuciones adaptativas, las cuales se originan a partir de los pulsos que ubiquemos en la memoria. En esta sección estudiaremos la relación entre estos pulsos originales y la formación de la ZIE.

Para ilustrar este proceso, inicialmente partiremos de un filtro LTP no fraccional que consiste simplemente en aplicar un filtro de retardo. En la figura 5.8 se muestra un ejemplo en el que hemos ubicado $P = 3$ pulsos en la memoria de la excitación previa (índices de posición negativos) con posiciones n_p ={-55,-45,-10} y amplitudes b_p ={1, 0.5, 1.25}, los cuales establecen la entrada para el filtrado LTP. En el ejemplo consideramos 4 subtramas (40 muestras por subtrama), cuyos retardos LTP vienen dados por T_a ={60, 50, 25, 42} con ganancias adaptativas g_a ={1, 2, 0.5, 1.25}. Como podemos ver, los P pulsos ubicados

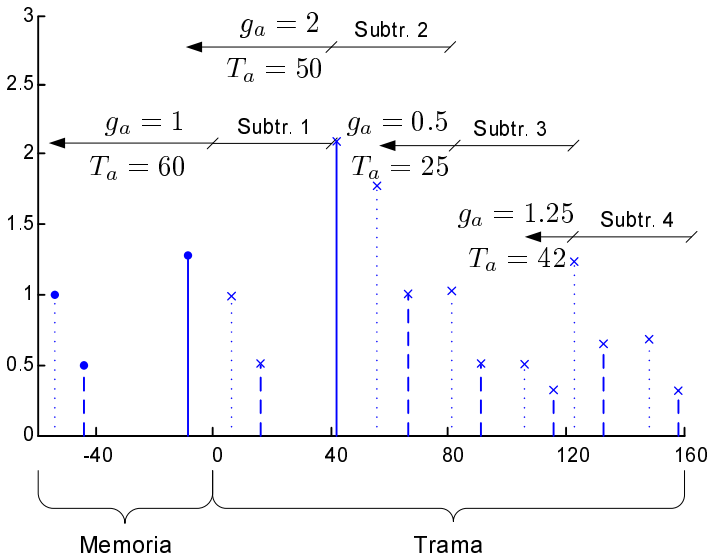


Figura 5.8: Generación de los pulsos ZIE (marcas **x**) a partir de un conjunto de pulsos iniciales (marcas **o**) obtenidos aplicando un filtrado LTP no fraccional para cada subtrama.

en la memoria se replican varias veces con diferentes amplitudes a lo largo de la trama. El proceso recursivo conlleva que las amplitudes finales de los pulsos de la excitación ZIE dependan de las amplitudes iniciales y las ganancias adaptativas acumuladas a lo largo de las subtramas, que desde ahora en adelante consideraremos como pesos. Aplicando el principio de superposición podemos representar la excitación ZIE como una combinación de réplicas de cada pulso inicial p ($p = 0, 1, \ldots, P-1$),

$$\hat{e}_{zi}(n) = \sum_{p=0}^{P-1} b_p \sum_{j=0}^{L-1} w(n_{p,j})\delta(n - n_{p,j}) \tag{5.16}$$

donde b_p son las amplitudes de los pulsos iniciales, mientras que $n_{p,j}$ y $w(n_{p,j})$ son, respectivamente, la posición y peso para la réplica j del pulso p. Aunque el número de réplicas, L_p, que aparecen a partir de un pulso inicial p es arbitrario, en la ecuación (5.16) hemos tomado su valor máximo, es decir $L = \max L_p$ para $0 \leq p < P$. Al considerar L como el número máximo de réplicas considerando los P pulsos ubicados en memoria, la ecuación (5.16) considerará réplicas en exceso. No obstante, este problema se puede resolver otorgando un peso nulo a dichas réplicas.

A partir de la expresión (5.16) podemos determinar la correspondiente señal de síntesis

como,

$$\begin{aligned}\hat{s}_{zi}(n) = h_w(n) * \hat{e}_{zi}(n) &= \sum_{k=0}^{N-1} h_w(k) \sum_{p=0}^{P-1} b_p \sum_{j=0}^{L-1} w(n_{p,j})\delta(n - n_{p,j} - k) \\ &= \sum_{p=0}^{P-1} b_p \sum_{j=0}^{L-1} w(n_{p,j}) \sum_{k=0}^{N-1} h_w(k)\delta(n - n_{p,j} - k) \\ &= \sum_{p=0}^{P-1} b_p \sum_{j=0}^{L-1} w(n_{p,j}) h_w(n - n_{p,j}) \end{aligned} \tag{5.17}$$

Por tanto, podemos expresar el error cuadrático de la ecuación (5.15) como,

$$E_w = \sum_{n=0}^{N-1} (s_{zi}(n) - \sum_{l=0}^{P-1} b_p g_p(n - n_p))^2 \tag{5.18}$$

donde,

$$g_p(n - n_p) = \sum_{j=0}^{L-1} w(n_{p,j}) h_w(n - n_{p,j}) \tag{5.19}$$

Puesto que la ecuación (5.18) responde a la misma estructura de la ecuación (5.8), las amplitudes óptimas b_p dadas las posiciones iniciales de los pulsos n_p (las posiciones de las réplicas $n_{p,j}$ se derivan automáticamente a partir de ellas) se pueden obtener aplicando las expresiones (5.9) a (5.12). Del mismo modo, las posiciones de los pulsos se pueden determinar aplicando el algoritmo cuasi-óptimo utilizado en el esquema multipulso.

Integración de Filtro LTP Fraccional

En la sección previa, hemos expuesto cómo integrar los parámetros del codificador en la optimización de la búsqueda de pulsos que constituyen el FEC, bajo el supuesto de que el filtro LTP para cada subtrama corresponda con un simple retardo. Además, la aproximación basada en filtros LTP no fraccionales considera un único conjunto de coeficientes LPC por trama. En este caso la minimimización no se puede realizar a nivel de subtrama ya que los parámetros de los pulsos iniciales afectan simultáneamente a todas las subtramas. No obstante, los codificadores CELP de última generación integran filtros predictivos fraccionales, que responden de forma general a la expresión (3.14), y utilizan distintos filtros LP para cada una de las subtramas.

Para resolver estos problemas, introducimos una nueva notación donde representare-

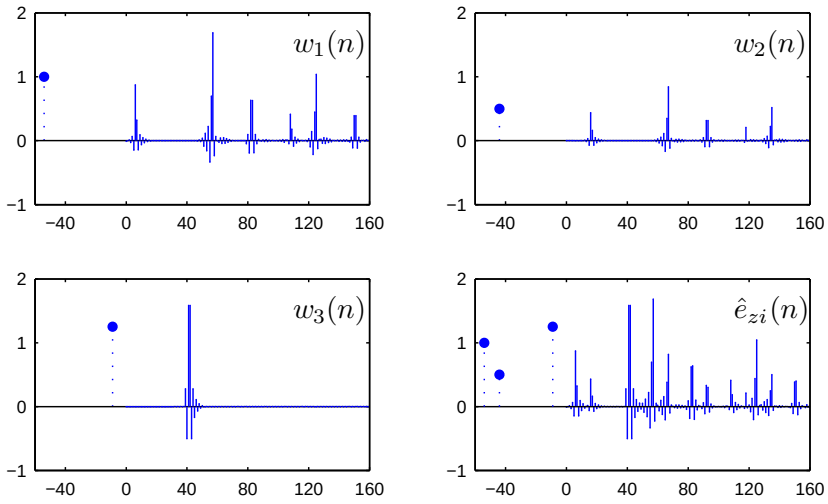


Figura 5.9: Generación de la señal ZIE $\hat{e}_{zi}(n)$ a partir de un conjunto de pulsos iniciales (marcas **o**) obtenidos como la suma de las señales $w_1(n)$, $w_2(n)$ y $w_3(n)$ (señales generadas aplicando un filtro LTP fraccional por subtrama y su correspondiente ganancia adaptativa).

mos la excitación ZIE como una señal completa y no como una serie de réplicas de pulsos. Así, ahora consideraremos que cada pulso p genera una cierta señal $w_p(n)$ definida para todas las muestras de la trama ($0 \leq n \leq N-1$). La integración del filtrado LTP no fraccional del apartado anterior nos lleva a un caso particular en el que la señal $w_p(n)$ se encuentra formada por una serie de pulsos (véase ecuación (5.16)) de modo que,

$$w_p(n) = \sum_{j=0}^{L-1} w(n_{p,j})\delta(n - n_{p,j}) \tag{5.20}$$

Sin embargo, $w_p(n)$ no tiene por qué restringirse a un conjunto de pulsos. Esta aproximación nos permite integrar filtros LTP fraccionales, de forma que cada señal $w_p(n)$ se obtiene llevando a cabo el filtrado LTP individual de cada pulso a lo largo de toda la trama. En la figura 5.9 se muestra un ejemplo de las señales $w_p(n)$ generadas para los pulsos iniciales considerados en la figura 5.8, pero considerando filtros LTP fraccionales. En contraste con la aproximación basada en filtros LTP no fraccionales (véase figura 5.8), la señal ZIE obtenida presenta una estructura fina más rica.

Finalmente, aplicando el principio de superposición, la ZIE final se obtiene como,

$$\hat{e}_{zi}(n) = \sum_{p=0}^{P-1} b_p w_p(n) \tag{5.21}$$

proporcionando para este caso la siguiente señal sintetizada,

$$\begin{aligned}\hat{s}_{zi}(n) = h_w(n) * \hat{e}_{zi}(n) &= \sum_{k=0}^{N-1} h_w(k) \sum_{p=0}^{P-1} b_p w_p(n-k) \\ &= \sum_{p=0}^{P-1} b_p \sum_{k=0}^{N-1} h_w(k) w_p(n-k) \\ &= \sum_{p=0}^{P-1} b_p g_p(n)\end{aligned} \tag{5.22}$$

donde $g_p(n)$ es la respuesta LP a la señal $w_p(n)$, la cual se obtiene como,

$$g_p(n) = \sum_{k=0}^{N-1} h_w(k) w_p(n-k) = h_w(n) * w_p(n) \tag{5.23}$$

Como podemos observar, la expresión (5.23) es una generalización de la expresión (5.19) que nos permite considerar cualquier tipo de señal que surja del filtrado LTP.

En este caso, el error cuadrático a minimizar se obtiene como,

$$E_w = \sum_{n=0}^{N-1} (s_{zi}(n) - \sum_{l=0}^{P-1} b_p g_p(n))^2 \tag{5.24}$$

Esta expresión es similar a (5.8) y, por tanto, podemos obtener las amplitudes óptimas b_p aplicando el criterio LSE mediante la ecuación (5.9), pero sustituyendo la matriz $\mathbf{H}$ por la matriz $\mathbf{G}$, definida como,

$$\mathbf{G} = \begin{bmatrix} g_0(0) & g_1(0) & \cdots & g_{P-1}(0) \\ g_0(1) & g_1(1) & \cdots & g_{P-1}(1) \\ \vdots & \vdots & \ddots & \vdots \\ g_0(N-1) & g_1(N-1) & \cdots & g_{P-1}(N-1) \end{bmatrix} \tag{5.25}$$

de modo que,

$$\mathbf{b}^* = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{s}$$

De igual modo, podemos reutilizar el algoritmo subóptimo para la búsqueda de amplitudes y posiciones de los pulsos iniciales (descrito en la sección 5.5) mediante la redefinición de $\mathbf{\Phi}$ y $\mathbf{c}$ como,

$$\mathbf{\Phi} = \mathbf{G}^T\mathbf{G} \qquad \mathbf{c} = \mathbf{G}^T\mathbf{s} \tag{5.26}$$

El esquema que acabamos de proponer, además de permitir la utilización de filtros LTP fraccionales, tiene la ventaja adicional de que nos permite trabajar subtrama a subtrama y, por tanto, admite el empleo de diferentes conjuntos LP para cada subtrama.

Resultados Experimentales

La tabla 5.4 muestra los resultados PESQ obtenidos mediante las aproximaciones multipulso LTP no fraccional (MP-LTP-nf) y LTP fraccional (MP-LTP-f) tomando P como 1, 2 y 4 pulsos. Los resultados fueron obtenidos utilizando el codificador AMR 12.2 kbps en un marco experimental como el definido en la sección 5.4.3. El codificador AMR 12.2 kbps emplea un filtro LTP no fraccional y un conjunto único de coeficientes LP por trama. Debido a las limitaciones de MP-LTP-nf, en este caso se aproximó el filtro LTP mediante un simple retardo y se obtuvo un único conjunto de coeficientes LP por trama promediando los coeficientes LP de las correspondientes subtramas (promedio realizado en el dominio LSF). Por contra, estas restricciones no son necesarias en MP-LTP-f ya que esta aproximación nos permite trabajar subtrama a subtrama. Como se puede observar los resultados no difieren mucho entre ambas aproximaciones, aunque MP-LTP-f obtiene un rendimiento ligeramente superior de forma consistente. Evidentemente, los resultados mejoran a medida que ubicamos un mayor número de pulsos iniciales, P, en la memoria previa a la trama. Así, la aproximación MP-LTP-f con $P = 4$ pulsos obtiene un rendimiento comparable a la recuperación ideal del diccionario adaptativo (todas sus muestras). Además, la propuesta MP-LTP-f con $P = 1$ pulso por trama obtiene aproximadamente los mismos resultados que la aproximación multipulso que utilizamos de partida con 4 pulsos por trama (1 pulso por subtrama) de la tabla 5.3 (experimento MP $L = 1$).

5.5.2. Aplicación Práctica del Esquema Multipulso-LTP Fraccional

En los apartados previos hemos verificado que la aproximación basada en pulsos de las muestras de la trama anterior produce una correcta resincronización del diccionario adap-

Tec. Recup. / *Error Prop.*	*Condiciones de Pérdidas*							
	4 %	*7 %*	*10 %*	*13 %*	*16 %*	*18 %*	*21 %*	*23 %*
AMR 12.2	3.28	2.98	2.74	2.55	2.38	2.28	2.13	2.05
Rec. Ideal	3.49	3.24	3.05	2.90	2.76	2.68	2.57	2.50
MP-LTP-nf $P = 1$	3.45	3.19	2.99	2.83	2.69	2.61	2.49	2.41
MP-LTP-nf $P = 2$	3.46	3.20	3.00	2.84	2.70	2.62	2.50	2.42
MP-LTP-nf $P = 4$	3.46	3.21	3.01	2.85	2.71	2.62	2.50	2.42
MP-LTP-f $P = 1$	3.45	3.19	3.00	2.84	2.71	2.62	2.51	2.44
MP-LTP-f $P = 2$	3.46	3.20	3.02	2.86	2.72	2.64	2.53	2.46
MP-LTP-f $P = 4$	3.47	3.22	3.03	2.87	2.74	2.66	2.55	2.47

Tabla 5.4: Resultados PESQ en condiciones de pérdidas obtenidos para AMR 12.2 kbps aplicando diversas técnicas de recuperación: recuperación ideal (Rec. ideal); recuperaciones multipulso integrando filtro LTP no fraccional (MP-LTP-nf) y fraccional (MP-LTP-f) con $P = 1, 2, 4$ pulsos por trama.

tativo. No obstante, para poder utilizar este esquema en la práctica debemos representar las amplitudes y posiciones de los pulsos con precisión finita, generando así un código FEC que será enviado junto a los parámetros de codificación. Evidentemente, el objetivo es transmitir la menor cantidad de información colateral posible, aunque hay que tener en cuenta que una cuantización gruesa podría dar lugar a una pérdida de rendimiento del esquema de resincronización.

En esta sección nos centraremos en la codificación necesaria para el esquema MP-LTP-f con $P = 1$ pulso por trama, ya que proporciona un incremento relativo del rendimiento sustancialmente mejor que utilizando 2 o más pulsos (véase tabla 5.4). Así, las propuestas de codificación que presentamos se encargarán de cuantizar la amplitud y posición de un único pulso. Aunque los esquemas que presentamos también podrían utilizarse para la cuantización de más de un pulso, en este caso el rendimiento no sería óptimo ya que no se explotarían las posibles relaciones entre los pulsos. Para el desarrollo de esta sección utilizaremos el codificador AMR 12.2 kbps, aunque los esquemas de codificación propuestos son fácilmente extensibles a otros codificadores CELP.

Cuantización de las Amplitudes de los Pulsos

En este caso, la cuantización de las amplitudes de los pulsos se llevará a cabo mediante un cuantizador no uniforme. El diseño de estos cuantizadores se puede llevar a cabo aplicando el algoritmo de Lloyd-Max a una base de datos de entrenamiento (en nuestro caso utilizamos el subconjunto de entrenamiento definido sobre la base de datos TIMIT).

Codificación Amplitud	*Condiciones de Pérdidas*							
	4 %	*7 %*	*10 %*	*13 %*	*16 %*	*18 %*	*21 %*	*23 %*
4 bits	3.44	3.18	2.98	2.83	2.69	2.61	2.50	2.42
5 bits	3.44	3.19	2.99	2.84	2.70	2.62	2.51	2.43
8 bits	3.45	3.19	3.00	2.84	2.71	2.62	2.51	2.44

Tabla 5.5: Resultados PESQ en condiciones de pérdidas obtenidos para voz decodificada AMR 12.2 kbps resincronización con un pulso (MP-LTP-nf $P = 1$) cuantizado con distintos números de bits (la posición de los pulsos no se cuantiza).

La tabla 5.5 muestra los resultados PESQ obtenidos aplicando diferentes cuantizadores (4, 5 y 8 bits) para el esquema de resincronización con un pulso MP-LTP-nf (por ahora, supondremos que las posiciones no son cuantizadas). Los resultados indican que el esquema propuesto se manifiesta bastante robusto al error de cuantización cometido en las amplitudes de los pulsos, obteniendo con 5 bits un rendimiento similar al esquema sin cuantización de amplitud.

Cuantización de las Posiciones de los Pulsos

En contraste con la codificación de los pulsos, en la que primero se computan las amplitudes óptimas y posteriormente se cuantizan, la codificación de las amplitudes exige conocer a priori (antes del proceso de minimización LSE) las posiciones permitidas. De este modo, durante el proceso de búsqueda de las posiciones cuasi-óptimas de los pulsos nos ceñiremos sólo a aquellas posiciones disponibles por el cuantizador. Para ello, simplemente tomaremos como nulas las señales $w_p(n)$ que surjan de ubicar el pulso p en una posición no permitida, descartando así esta posición en el proceso de minimización de error.

La opción más básica para la cuantización de las posiciones consiste en representar la posición absoluta del pulso (dentro de la trama previa) como un valor absoluto de N bits. De este modo, limitamos la búsqueda de posiciones óptimas a las $P_{pos} = 2^N$ posiciones previas a la trama actual. Teniendo en cuenta la figura 5.7, las muestras más allá de las 128 previas a la trama actual apenas son utilizadas y, por tanto, con $N = 7$ bits serían suficientes para codificar la posición del pulso.

No obstante, este esquema se puede refinar si, en lugar de utilizar un valor absoluto, consideramos un valor relativo referido al retardo LTP (considerando sólo su parte entera).

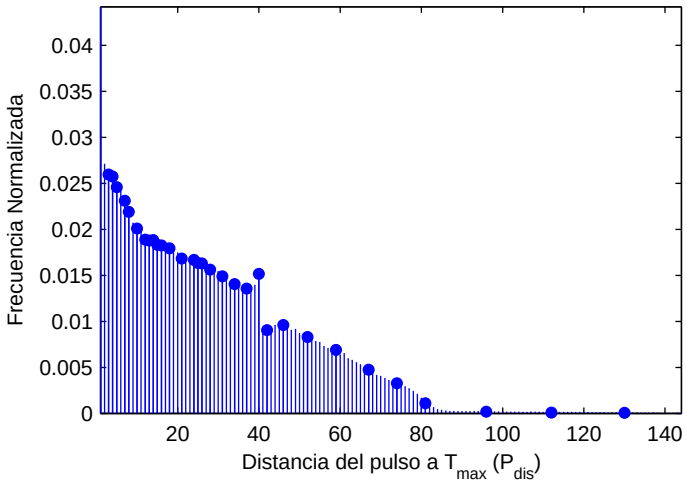


Figura 5.10: Histograma de las posiciones de los pulsos relativas a T_{max} utilizados para evitar la propagación de error del codificador AMR 12.2 kbps. Las posiciones relativas marcadas con puntos en negrita son las seleccionadas como disponibles por el cuantizador Lloyd-Max con 32 centroides.

Para cada trama podemos obtener el retardo máximo, T_{max}, de la siguiente manera,

$$T_{max} = \max[T_0, T_1 - N_s, T_2 - 2N_s, \ldots, T_{M-1} - (M-1)N_s] \tag{5.27}$$

donde T_i corresponde con la parte entera del retardo LTP de la subtrama i, M es el número de subtramas por trama y N_s es la longitud de la subtrama. Puesto que las muestras anteriores al retardo máximo no son utilizadas por el filtrado LTP, el pulso de sincronización debe de ser posterior a T_{max} y, por tanto, su posición n_p se puede codificar eficientemente como el número de muestras posteriores a T_{max}, es decir, $n_{dis} = n_p - T_{max}$.

En caso de que empleemos el posicionamiento relativo a T_{max}, una primera opción sería utilizar las 2^N primeras posiciones tras T_{max} (pertenecientes a la trama previa) durante el proceso de minimización. En este sentido, la figura 5.10 muestra la distribución de n_{dis} al llevar a cabo el proceso de minimización de error con un pulso sobre el codificador AMR 12.2 utilizando la base de datos TIMIT. Como se puede observar, un porcentaje significativo de posicionamientos del pulso coincide con T_{max}, mientras que el porcentaje de posiciones con $n_{dis} > 64$ ($N = 6$) es menos significativo.

El esquema de codificación previo presenta la desventaja de que descarta un grupo de posiciones consecutivas ($n_{dis} > 2^N$). En principio, ya que estas posiciones no son muy frecuentes y generalmente no pertenecen a segmentos sonoros, no parece una cuestión crítica. No obstante, este esquema se puede mejorar diseñando un cuantizador no uniforme

para la codificación de n_{dis}. La idea que subyace consiste en disponer de una alta densidad de posiciones disponibles para las distancias de los pulsos más probables, y sólo unas pocas para las menos frecuentes. Para ilustrar esta idea, en la figura 5.10 se han marcado las distancias n_{dis} seleccionadas como disponibles por el cuantizador no uniforme de 5 bits, diseñado mediante el algoritmo de Lloyd-Max.

Finalmente, en la figura 5.11 se muestra una comparativa del rendimiento de los diferentes esquemas de codificación expuestos utilizando 4, 5 y 6 bits para la codificación de las posiciones y otros tantos para las amplitudes. En este caso, se han promediado las puntuaciones PESQ para todas las condiciones de canal adversas (las representadas en las tablas 5.4 y 5.5) con el fin de reducir el número de resultados. Cuando no se utiliza esquema de codificación obtenemos un resultado de referencia superior de 2.84. Por contra, el resultado inferior viene dado por el codificador AMR 12.2 kbps (sin utilizar ningún tipo de información colateral) que obtiene una nota promedio PESQ de 2.55. Como era de esperar, en general, el rendimiento decae a medida que empleamos menor número de bits. Los mejores resultados se obtienen utilizando cuantización no uniforme, particularmente para las tasas de bits más bajas. Teniendo en cuenta estos resultados, podemos concluir que una cuantización de 6 bits para la posición y 5 para la amplitud podría ser una buena elección, ya que conlleva una reducción mínima del rendimiento respecto al esquema sin cuantización con una tasa colateral de 0.55 kbps (la tasa de codificación final es de 12.75 kbps).

5.5.3. Evaluación del Esquema Práctico

A lo largo de esta sección hemos utilizado la herramienta PESQ para llevar a cabo la evaluación de las diferentes alternativas propuestas para el desarrollo de la técnica FEC multipulso, ya que su carácter objetivo y automático nos permite establecer una rápida comparativa entre los métodos expuestos. No obstante, en este apartado se presenta una evaluación perceptual subjetiva, cuyo objetivo es corroborar la eficiencia del esquema práctico desarrollado.

A este efecto hemos empleado la metodología MUSHRA, introducida con anterioridad en la sección 5.3, para llevar a cabo una evaluación perceptual realizada por oyentes reales. En particular, la evaluación fue realizada utilizando un conjunto de 10 oyentes expertos, donde cada uno de los cuales evaluó 16 ítems diferentes obtenidos de la base de datos Albayzin [144]. Se seleccionó esta base de datos para la extracción del material de prueba ya que se encontraba en el idioma nativo de los oyentes (castellano). Las frases

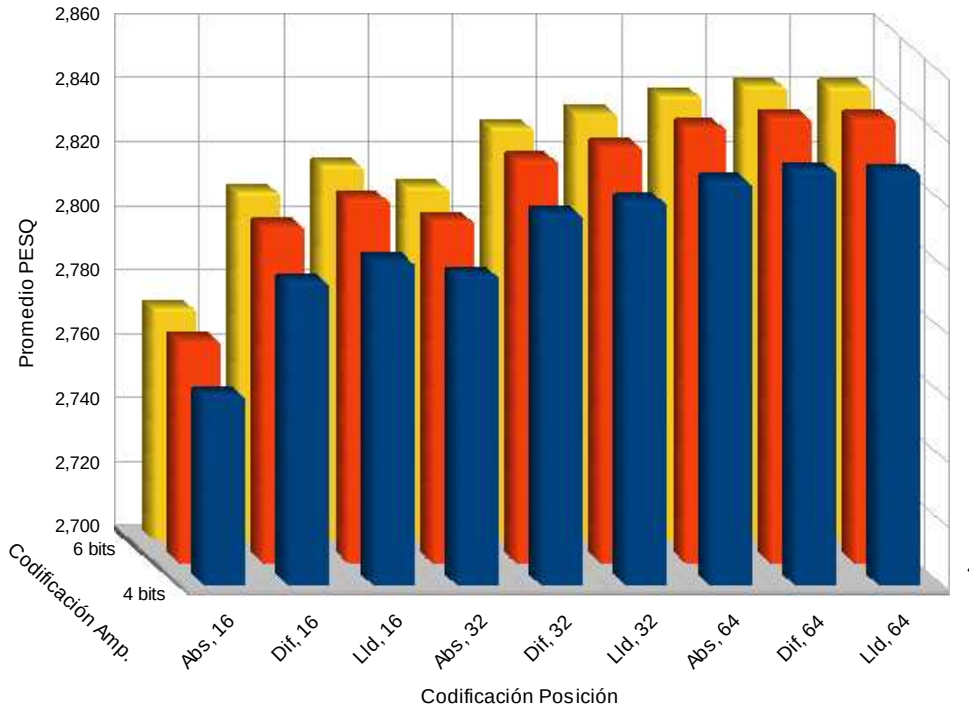


Figura 5.11: Efecto de la cuantización sobre el pulso de resincronismo en términos de promedio de resultados PESQ (sobre las condiciones de canal adversas) para el codificador AMR 12.2. La ubicación de los pulsos se restringe a 16, 32 y 64 posiciones (4, 5 y 6 bits) disponibles con esquemas de codificación de posicionamiento absoluto (Abs.), posicionamiento relativo a T_{max} (Dif.) y posicionamiento relativo utilizando cuantización no uniforme (Lld.), mientras que las amplitudes se codifican con cuantizadores no uniformes de 4, 5 y 6 bits.

seleccionadas están fonéticamente balanceadas y pronunciadas por locutores masculinos y femeninos en la misma proporción. Puesto que la base de datos se encuentra muestreada originalmente a 16 kHz fue necesario someter las frases a un proceso de diezmado a 8 kHz.

Durante la evaluación MUSHRA, los oyentes deben comparar las diferentes técnicas evaluando la calidad de la señal obtenida para cada ítem en comparación con una señal de referencia y un anclaje. La señal original muestreada a 8 kHz fue utilizada como señal de referencia, mientras que el anclaje se consiguió mediante un proceso de codificación AMR 12.2 kbps con un porcentaje de pérdidas del 30 % distribuidas de forma aleatoria (sólo se utilizó el algoritmo PLC incluido en el estándar de codificación). Ambas señales se incluyen entre los estímulos a evaluar, presentándose al oyente de forma ciega (ordenadas aleatoriamente). Además, el oyente dispone de la señal de referencia etiquetada como tal. Por tanto, el oyente debe de asignar a la referencia oculta una puntuación máxima,

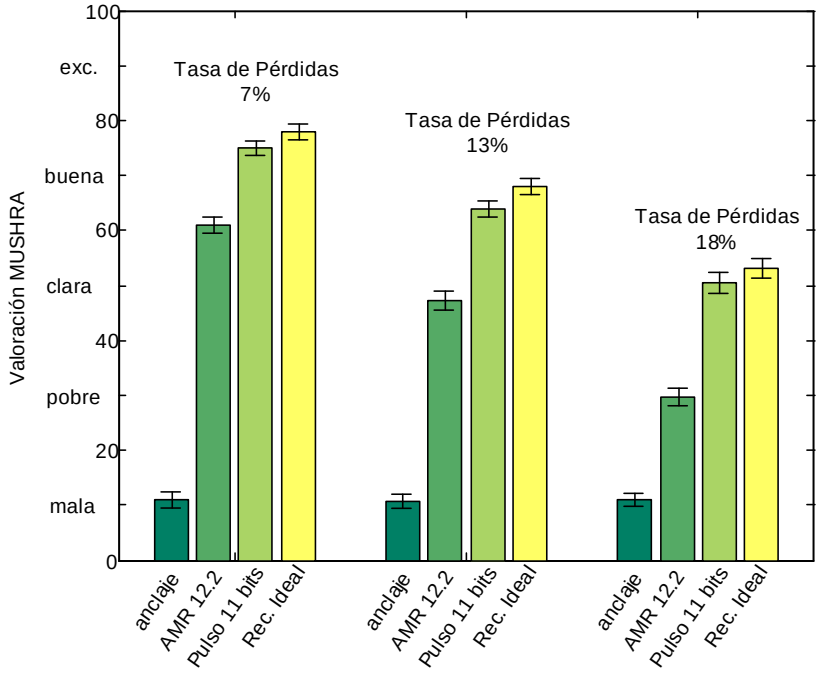


Figura 5.12: Evaluación MUSHRA del codificador AMR con la técnica de mitigación estándar (AMR 12.2 kbps), con recuperación ideal de las muestras previas (Rec. Ideal) y empleando un pulso de resincronización codificado mediante 11 bits (12.75 kbps).

sirviendo el anclaje para adecuar la escala de evaluación lo máximo posible a una escala absoluta.

Particularmente, hemos comparado tres algoritmos de mitigación: el incluido por AMR, un esquema ideal basado en la recuperación total del diccionario adaptativo donde todas las muestras previas están disponibles y el esquema multipulso descrito a lo largo de esta sección utilizando un único pulso cuantizado con 11 bits (6 bits destinados a la codificación de la posición y 5 a la amplitud). Además, la evaluación se llevó a cabo para tres condiciones de canal distintas obtenidas mediante un modelo de Bernoulli (pérdidas aleatorias) con probabilidades de pérdidas del 7 %, 13 % y 18 %.

La figura 5.12 muestra las valoraciones medias realizadas por los sujetos acompañadas por los intervalos de confianza del 95 %. Además, la tabla 5.6 recoge los resultados PESQ obtenidos por estos esquemas, donde hemos incorporado también el esquema de un pulso propuesto pero sin cuantizar. Este último resultado, junto a la recuperación ideal de la memoria del diccionario adaptativo, establece una referencia superior para la propuesta de un pulso cuantizado con 11 bits.

Como se puede observar, los resultados PESQ son consistentes con los obtenidos en

Tec. Recup. Error Prop.	*Condiciones de Pérdidas*							
	4 %	*7 %*	*10 %*	*13 %*	*16 %*	*18 %*	*21 %*	*23 %*
AMR 12.2	3.28	2.98	2.74	2.55	2.38	2.28	2.13	2.05
Rec. Ideal	3.49	3.24	3.05	2.90	2.76	2.68	2.57	2.50
Pulso no cuan.	3.45	3.19	3.00	2.84	2.71	2.62	2.51	2.44
Pulso 11 bits	3.44	3.18	2.99	2.84	2.70	2.62	2.50	2.43
Comb. $N = 1$	3.45	3.24	3.07	2.92	2.78	2.70	2.58	2.50
iLBC 15.2	3.51	3.30	3.13	2.99	2.86	2.78	2.67	2.60

Tabla 5.6: Resultados PESQ en condiciones de pérdidas obtenidos sin aplicar esquema de recuperación de error de propagación (AMR 12.2 kbps); con recuperación ideal (rec. ideal); FEC con 1 pulso sin cuantizar (pulso no cuant.); FEC con 1 pulso cuantizado 11 bits (con una tasa de transmisión 12.65 kbps); la propuesta de combinación de tramas (Comb. $N = 1$; con una tasa de transmisión de 12.75 kbps); y el codificador iLBC (15.2 kbps).

la evaluación subjetiva. No obstante, las diferencias de calidad perceptual se hacen más notorias a través de la evaluación MUSHRA. En particular, el esquema multipulso LTP propuesto logra una mejora ciertamente significativa sobre el algoritmo de mitigación estándar, mejora que es aproximadamente igual a la que ofrece la recuperación ideal del diccionario adaptativo. Sin embargo, mientras que esta última solución es inviable en términos de ancho de banda, nuestra propuesta sólo requiere de un incremento de 0.55 kbps en la tasa de codificación total.

En la tabla 5.6 también se muestran los resultados obtenidos con iLBC 15.2 kbps y la propuesta basada en la combinación de tramas con $N = 1$ (véase sección 5.4). En este último caso la tasa de transmisión obtenida es de 12.75 kbps, la cual es muy próxima al esquema FEC multipulso con 12.65 kbps. Ambas propuestas consiguen un rendimiento superior al del codificador AMR 12.2 kbps con un incremento moderado de la tasa de transmisión. En particular, la codificación basada en la combinación de tramas muestra un rendimiento ligeramente superior al esquema FEC. El hecho de que el esquema de combinación de tramas no utilice técnicas predictivas en la codificación de los parámetros (coeficientes LPC, ganancias, etc.) es una de las posibles razones que justifican este comportamiento.

Paralelamente, llevamos a cabo la evaluación de la robustez de la arquitectura NSR basada en el codificador AMR y la aplicación práctica del esquema FEC propuesto (basado en un único pulso codificado con 11 bits). El marco experimental utilizado es similar al propuesto en el capítulo 4, el cual hace uso de un modelo de Gilbert para la emulación de

	Tasa de Pérdidas	*Long. media ráfaga*			
		1	2	3	4
iLBC 15.2 kbps	*5 %*	98.56	97.74	96.79	96.05
	10 %	98.19	96.35	94.91	93.07
	15 %	97.67	95.13	92.43	89.78
	20 %	97.06	93.82	90.34	87.11
AMR-FEC 12.75 kbps	*5 %*	98.54	97.54	96.49	95.81
	10 %	98.15	96.04	94.34	92.82
	15 %	97.61	94.75	91.66	89.26
	20 %	97.13	93.00	88.81	85.90
$N = 1$ 12.65 kbps	*5 %*	98.18	97.33	96.35	95.76
	10 %	97.49	95.51	94.13	92.60
	15 %	96.51	94.00	91.28	89.04
	20 %	95.46	92.42	88.90	85.98
AMR 12.2 kbps	*5 %*	97.93	96.46	95.22	94.94
	10 %	96.59	93.97	91.95	91.14
	15 %	94.51	91.21	88.55	87.07
	20 %	91.49	87.87	85.07	83.07

Tabla 5.7: Comparativa de precisión de reconocimiento WAcc entre iLBC (15.2 kbps), AMR-FEC (12.2 + 0.55 kbps) y AMR (12.2 kbps) para condiciones de canal generadas mediante un modelo de Gilbert.

diferentes condiciones de canal. La tabla 5.7 muestra una comparativa de los resultados obtenidos por las dos propuestas realizadas en este capítulo (combinación de tramas con $N = 1$ y FEC de 1 pulso) frente a los resultados de AMR 12.2 kbps e iLBC 15.2 kbps. Como podemos observar, las propuestas realizadas introducen una notoria mejora frente a pérdidas sobre el esquema de codificación AMR, diferencias que se hacen más plausibles en el esquema FEC para longitudes de ráfaga cortas (L_{burst} =1 y 2). Por contra, cuando la longitud de las pérdidas es mayor, los algoritmos PLC integrados en los decodificadores producen un apagado progresivo (*muting*) de la señal decodificada. En este caso, los silencios artificiales insertados originan una pérdida de rendimiento en el reconocedor debida a errores de inserción. Por este motivo, la reducción del error de propagación es crítica para pérdidas de corta duración y no tanto para las ráfagas de pérdidas de larga duración. Así, para una condición de un 20 % de pérdidas aisladas ($L_{burst} = 1$), el esquema FEC logra incrementar el rendimiento de AMR 12.2 kbps en más de 6.5 puntos, consiguiendo un resultado similar al obtenido por iLBC 15.2 kbps, pero con una tasa de codificación final de 12.75 kbps.

5.6. Resumen de Resultados y Conclusiones

En este capítulo hemos propuesto diversas técnicas de recuperación basadas en el emisor cuyo objetivo es la mejora de la robustez de los esquemas de codificación frente a pérdidas de paquetes. Puesto que no tiene sentido introducir modificaciones en el esquema de codificación que estén únicamente orientadas a incrementar el rendimiento de los sistemas de reconocimiento remoto, el fin de las técnicas propuestas es incrementar la calidad perceptual de la voz sintetizada. En este sentido, hemos partido de la hipótesis de partida de que la mejora de la calidad perceptual repercutiría en un aumento de la tasa de reconocimiento. Hipótesis que hemos verificado mediante las correspondientes medidas de reconocimiento.

En particular, existen dos causas del bajo rendimiento frente a pérdidas de la arquitectura NSR. En primer lugar, ciertos codificadores introducen fuertes dependencias intertrama lo que ocasiona un bajo rendimiento en caso de pérdidas de paquetes. En segundo lugar los algoritmos de mitigación de pérdidas (PLC) que incorporan los codificadores están diseñados bajo premisas perceptuales, es decir, intentan minimizar el impacto perceptual de las pérdidas. Atendiendo al primer problema, veíamos que el codificador iLBC conseguía robustecer notablemente el esquema de codificación de voz frente a pérdidas. Este codificador conseguía mejorar su prestaciones eliminando las dependencias intertrama. De este modo, la síntesis de la señal de voz de una cierta trama no depende de la correcta síntesis de tramas anteriores. No obstante, el incremento de la tasa de reconocimiento frente a los codificadores basados en el paradigma CELP es notable.

Las propuestas realizadas en este capítulo se han centrado en combatir el error de propagación introducido por los codificadores CELP con un aumento moderado de la tasa de bits. Así, la primera de las propuestas realizada consiste en una nueva técnica de codificación basada en la combinación de los esquema de codificación iLBC y CELP. Concretamente, el esquema intercala tramas independientes (iLBC) y dependientes (CELP) de modo que consigue combinar la robustez frente a pérdidas de paquetes de iLBC con las bajas tasas de codificación CELP. La idea subyacente consiste en utilizar las tramas iLBC para limitar el error de propagación típico de los esquemas CELP. Así, modificando el número de tramas CELP insertado entre dos tramas iLBC adyacentes se consigue balancear el nivel de robustez frente a pérdidas y la tasa de codificación final. Atendiendo a este principio, se llevó a cabo una implementación de este esquema utilizando tramas iLBC (tasa de 15.2 kbps) y tramas ACELP (tasa de 10.2 kbps). Los resultados experimentales sin pérdidas muestran que el rendimiento perceptual de este esquema es ligeramente

inferior al del estándar de codificación AMR (utilizando modos con similares tasas de transmisión). No obstante, el rendimiento de la propuesta en condiciones con pérdidas es similar al de iLBC (tanto en calidad perceptual como en reconocimiento) y claramente superior al de AMR.

El esquema de combinación de tramas requiere aumentar considerablemente la complejidad tanto del codificador como del decodificador, ya que se precisa integrar dos esquemas de codificación distintos en ambos extremos. Una alternativa para aumentar la robustez de los esquemas CELP con una moderada tasa y complejidad consiste en emplear códigos FEC. En este sentido, la segunda propuesta que hemos realizado explora los códigos FEC dependientes del medio. En particular, los códigos FEC propuestos en este capítulo se basan en una representación multipulso de la señal de excitación de la trama previa. En caso de pérdida la representación multipulso es utilizada para resincronizar el diccionario adaptativo y, por tanto, evitar la propagación de error. Así, los parámetros de la representación multipulso son optimizados de modo que minimizan el error de la señal sintetizada respecto a la original. En la propuesta realizada este procedimiento de optimización multipulso se modifica para que tenga en cuenta los parámetros CELP ya enviados por el transmisor. De este modo, la representación del diccionario previo precisa de un número menor de pulsos y la búsqueda de éstos se puede llevar a cabo de un modo eficiente subtrama a subtrama. El esquema teórico se complementa con una aplicación práctica en la que se exploran diferentes esquemas de cuantización de las amplitudes y posiciones de los pulsos. Como resultado de esta última fase se propone una representación basada en un único pulso codificado mediante 11 bits, lo que supone un flujo de información colateral de 0.55 kbps. La validez de esta aplicación práctica se avaló mediante pruebas perceptuales objetivas (PESQ) y subjetivas (MUSHRA) en las que se obtuvo un notable incremento de la calidad perceptual, próximo al obtenido a través de la resincronización ideal del diccionario adaptativo.

En las pruebas de reconocimiento, al contrario de lo que sucedía en los resultados PESQ, el esquema FEC (12.75 kbps) consigue un rendimiento ligeramente superior al esquema de combinación de tramas (12.65 kbps). Este último permite, en el peor de los casos, la propagación de error durante N tramas, pasadas la cuales se produce una resincronización total del diccionario adaptativo. Por contra, el esquema FEC inicia la recuperación en la trama inmediatamente posterior a una pérdida, aunque esta recuperación del diccionario es sólo parcial (P pulsos). Esta diferencia entre los comportamientos de ambas soluciones puede justificar la controversia entre los resultados perceptuales y de reconocimiento. Así pues, la recuperación lenta pero total del diccionario adaptativo

produce mejores resultados perceptuales (esquema de combinación de tramas), mientras que una recuperación parcial del diccionario adaptativo (esquema FEC multipulso), pero más rápida, es mejor bajo la perspectiva del reconocimiento.

Finalmente, hay que resaltar que en ambas propuestas los resultados de reconocimiento confirmaron que los incrementos de la calidad perceptual en condiciones con pérdidas vienen acompañados de mejoras en las tasas de reconocimiento. Concretamente, los esquemas propuestos mejoran notablemente el rendimiento en condiciones de pérdidas con duración corta. Teniendo en cuenta que los algoritmos PLC empleados están basados en los principios de repetición hacia delante y apagado progresivo, las ráfagas de larga duración derivan en errores de inserción (silencios artificiales), marcando el rendimiento del reconocedor. No obstante, cuando las ráfagas son cortas, la operación de apagado progresivo no tiene lugar y, por tanto, el rendimiento del reconocedor es más sensible a los errores de propagación. En este sentido, las propuestas realizadas consiguen limitar los efectos de esta propagación consiguiendo así incrementar la robustez frente a pérdidas.

Capítulo 6

Mitigación de Pérdidas en el Receptor

6.1. Introducción

En el capítulo anterior establecimos una clasificación de las técnicas de robustecimiento frente a degradaciones de canal en la que aquéllas se dividen en dos grupos: técnicas basadas en el emisor y basadas en el receptor. Dentro de la arquitectura NSR apuntamos que carece de sentido considerar técnicas basadas en el emisor exclusivamente orientadas al reconocimiento. De esta forma, la modificación del emisor sólo es planteable bajo la premisa de mejorar la calidad perceptual de la voz decodificada. Así, propusimos ciertas técnicas basadas en el emisor que conseguían este objetivo. Además, verificamos que el empleo de dichas técnicas repercutía en una mejora del rendimiento de la arquitectura NSR ante pérdidas de corta duración. No obstante, veíamos que las medidas de mitigación de pérdidas adoptadas por el receptor limitaban el rendimiento del reconocedor remoto.

Los algoritmos de mitigación de pérdidas (PLC, *Packet Loss Concealment*) integrados en los decodificadores se encargan de generar los segmentos de señal correspondientes a los paquetes perdidos a partir de la información previamente recibida. Estos algoritmos están diseñados desde un punto de vista perceptual teniendo como objetivo que el segmento de voz artificial sea lo menos molesto posible. Bajo esta perspectiva, sólo utilizan la información recibida antes de la pérdida, ya que la espera para la obtención de las tramas posteriores causaría una latencia excesiva. Por este motivo, normalmente la generación del segmento perdido se lleva a cabo mediante la repetición de los últimos parámetros de codificación recibidos. Esta estrategia produce un segmento de voz artificial que replica

las características del último sintetizado correctamente. Sin embargo, la extensión de esta repetición durante un largo segmento (ráfaga de varios paquetes consecutivos) puede dar lugar a un sonido sintético molesto. Para evitar este efecto perjudicial se introduce un efecto de apagado progresivo (*muting*) sobre la señal de mitigación, de modo que tras un número determinado de tramas perdidas se anula completamente la señal.

El uso de este tipo de algoritmos por parte del decodificador introduce una pérdida de rendimiento en los sistemas de reconocimiento remoto ya que las consideraciones perceptuales resultan demasiado restrictivas. Así, la latencia del algoritmo PLC se puede prolongar hasta obtener la información correspondiente al segmento posterior a la pérdida. Igualmente, no tiene sentido el empleo de silencios artificiales generados por el *muting* durante la duración de la ráfaga. Además, en el capítulo anterior observamos que los codificadores de voz más extendidos (aquellos que hacen uso de la arquitectura CELP) presentan un problema adicional ante la pérdida de paquetes, ya que originan una cierta propagación de error debido a la desactualización de las memorias en el proceso de decodificación. Por tanto, se hace preciso el desarrollo de técnicas de mitigación de pérdidas específicas para sistemas de reconocimiento NSR.

Aunque se han presentado numerosas técnicas de mitigación para sistemas de reconocimiento DSR, pocas han sido las propuestas realizadas para sistemas NSR. En el presente capítulo llevamos a cabo una revisión, evaluación y propuesta de técnicas de migitación de pérdidas para sistemas de reconocimiento NSR basados en voz sintetizada por decodificadores CELP.

6.2. Esquema de Mitigación

Antes de iniciar el estudio de las diferentes alternativas existentes para el desarrollo de las técnicas de mitigación de pérdidas, es preciso introducir la arquitectura del sistema propuesto, así como ubicar el dominio donde estas tecnicas actúan.

En este sentido, el diagrama expuesto en la figura 6.1 muestra el esquema de mitigación de pérdidas propuesto. Como podemos observar, la señal de voz $s(n)$ es codificada en un flujo de bits $\mathbf{b}_t$ que es empaquetado y transmitido sobre una red de paquetes. Aunque sería posible emplear técnicas de codificación robustas, como las explicadas en el capítulo previo, trabajaremos bajo el supuesto de que el codificador empleado se basa en el paradigma CELP y que no hace uso de éstas. Así, las degradaciones características de la red transforman el flujo original $\mathbf{b}_t$ en el flujo recibido $\mathbf{c}_t$, donde las partes del flujo original correspondientes a los paquetes perdidos son omitidas. En el lado del servidor (receptor),

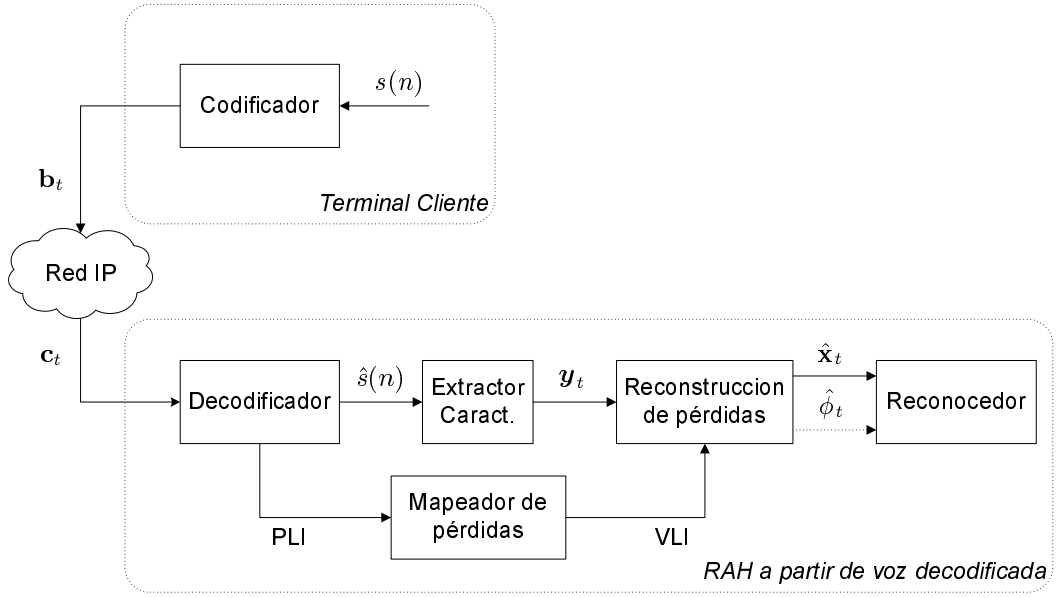


Figura 6.1: Esquema de mitigación de pérdidas propuesto para un sistema NSR basada en voz decodificada.

el decodificador sintetiza la señal $\hat{s}(n)$ que es utilizada como fuente para la extracción de una serie de vectores de características $\boldsymbol{y}_t$. A diferencia de los capítulos anteriores donde llevábamos a cabo el reconocimiento a partir de $\boldsymbol{y}_t$, en este punto insertamos un módulo reconstructor que obtiene una estima $\hat{\mathbf{x}}_t$ de la secuencia de vectores de características $\boldsymbol{x}_t$ que se obtendría en el caso de que no existieran pérdidas de paquetes. Este proceso exige a su vez introducir un bloque (*Mapeador de pérdidas* en la figura 6.1) que marque como perdidos aquellos vectores de características correspondientes a una pérdida, activando por consiguiente el módulo de reconstrucción.

Finalmente, es posible extraer cierta información $\hat{\phi}_t$ relativa a la confianza de las estimas que puede ser utilizada para mejorar el rendimiento del reconocedor. Estas técnicas, también conocidas como técnicas de mitigación basadas en el reconocedor, aprovechan el potente modelo estadístico utilizado en el proceso de reconocimiento para tratar las pérdidas.

6.3. Técnicas Sencillas de Mitigación

Esta sección se dedica a llevar a cabo una revisión sobre las técnicas de mitigación sencillas, desde un punto de vista computacional, basadas en el receptor. Éstas incluyen el borrado de tramas, las técnicas de inserción e interpolación.

6.3.1. Borrado de Tramas

Uno de los métodos más simples utilizado para reducir el efecto del canal de transmisión sobre el rendimiento del reconocedor consiste en unir los fragmentos recibidos antes y después de una ráfaga, de forma que se omite la información correspondiente a la pérdida. La idea de este esquema consiste en evitar aquellos segmentos distorsionados por el canal, ya que degradan rápidamente el reconocimiento. En principio, el reconocedor podría funcionar a pesar de la falta de ciertos segmentos en la secuencia de observaciones dada la redundancia de la señal de voz. Sin embargo, este esquema presenta un importante inconveniente tanto en decodificación como en reconocimiento de voz ya que la temporización de la voz se ve alterada drásticamente. Cuando esta técnica se utiliza para la decodificación de voz obtiene unos resultados pobres [145] puesto que introduce una alteración de la memoria previa a la reproducción, lo que degrada las prestaciones del sistema. Además, en el reconocimiento del habla, la unión o 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 las pérdidas se producen en ráfagas de una longitud relativamente alta que impide la evolución natural entre los estados del modelo correspondiente. Esto fuerza a que se produzca un salto brusco entre los estados anteriores y posteriores a la ráfaga que puede degradar significativamente la precisión de reconocimiento [146, 147].

6.3.2. Técnicas de Inserción

Las técnicas de inserción son aquellas que reconstruyen los paquetes perdidos sin tener en cuenta las características de la señal [119]. Las tramas perdidas son sustituidas insertando silencio, ruido o un valor estimado (como el valor medio de una base de entrenamiento).

En la sustitución por silencio se rellenan los segmentos perdidos por silencios artificiales, lo que consigue mantener la estructura temporal de la voz. Sin embargo, esta técnica no ofrece un gran rendimiento en reconocimiento de voz. Por contra, sí se utiliza extensamente en decodificación de voz debido a su facilidad de implementación [148], aunque la sustitución por ruido funciona mejor [149]. El estudio desarrollado por Warren [150], muestra que el cerebro humano subconscientemente recupera mejor segmentos con ruido que con silencio. En el caso del reconocimiento, Boulis *et al.* [89] demostraron que la inserción del valor medio obtiene unos resultados mejores que la técnica de unión de tramas o que la sustitución por silencio. Este resultado se puede justificar por el hecho de que la

sustitución por el valor medio no introduce los posibles errores de inserción que podrían provocar los silencios artificiales al ser colocados en medio de una palabra.

6.3.3. Técnicas de Interpolación

La interpolación es una técnica algo más avanzada que utiliza información recibida antes y después de una ráfaga para reconstruir la información perdida. De forma general, dada una ráfaga de vectores de características perdidos de longitud $T-1$, un vector interpolado para el instante t se obtiene como,

$$\hat{\mathbf{x}}_t = F(t; \boldsymbol{y}_{-\mathcal{M}+1}, \ldots, \boldsymbol{y}_0, \boldsymbol{y}_T, \ldots, \boldsymbol{y}_{T+\mathcal{N}-1}) \qquad 1 \leq t \leq T \tag{6.1}$$

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 realiza la interpolación. Por tanto, atendiendo a la notación de la expresión (6.1), los vectores $\boldsymbol{y}_0$ e $\boldsymbol{y}_T$ hacen referencia al vector previo y posterior a la ráfaga de pérdidas, respectivamente. De formal general, las técnicas basadas en la interpolación introducen una latencia en el proceso de decodificación de $T+\mathcal{N}-1$ vectores de características para $\mathcal{N} > 0$.

Las técnicas de repetición pueden considerarse un caso específico de las técnicas de interpolación. La repetición hacia delante surge de considerar $\mathcal{M} = 1$ y $\mathcal{N} = 0$ en la expresión (6.1), asignándose $\hat{\mathbf{x}}_t = \boldsymbol{y}_0$. De este modo, se mantiene la base de tiempos insertándose el vector de características previo a una pérdida. Igualmente, podemos considerar un esquema similar en el que se utiliza también el vector de características correspondiente al instante posterior a la ráfaga. Combinando ambos esquemas obtendríamos la siguiente expresión,

$$\hat{\mathbf{x}}_t = \begin{cases} \boldsymbol{y}_0 & t \leq T/2 \\ \boldsymbol{y}_T & t > T/2 \end{cases} \tag{6.2}$$

donde $\mathcal{M} = \mathcal{N} = 1$, siendo la latencia introducida de T vectores de características. Este es el algoritmo de mitigación utilizado por los estándares DSR [35, 37, 39, 40] desarrollados por el grupo de trabajo Aurora de la ETSI. El estándar asocia un número par de tramas por paquete [41], de modo que podemos resumir el algoritmo del siguiente modo: una vez que sucede una ráfaga de paquetes perdidos, conteniendo $2 \times B$ vectores de características, los primeros B vectores son reemplazados por el último vector recibido correctamente antes de la ráfaga ($\boldsymbol{y}_0$), mientras que los últimos B vectores se sustituyen por el primer

vector extraído tras la ráfaga ($\boldsymbol{y}_T$). Esta técnica también se conoce con el nombre de repetición del vector más cercano (NFR, *Nearest Frame Repetition*).

Dejando a un lado las técnicas basadas en repetición, normalmente la función F se toma como una función de interpolación polinomial, donde el grado mínimo requerido para el polinomio es $\mathcal{M} + \mathcal{N} - 1$. La interpolación lineal ($\mathcal{M} = \mathcal{N} = 1$) ha sido muy utilizada en numerosos contextos de reconocimiento remoto [122, 151, 152, 153] debido a su escasa complejidad en la reconstrucción que viene dada por la siguiente expresión,

$$\hat{\mathbf{x}}_t = \boldsymbol{y}_0 + \frac{t}{T}(\boldsymbol{y}_T - \boldsymbol{y}_0) \tag{6.3}$$

Al igual que la técnica de repetición NFR, la interpolación lineal requiere la introducción de una latencia igual a la longitud de la ráfaga.

James y Milner [154] proponen el uso de interpoladores no lineales donde se utilizan polinomios de Hermite cúbicos. La expresión general de estos interpoladores se expresa como,

$$\hat{\mathbf{x}}_t = \mathbf{a}_0 + \bar{t}\mathbf{a}_1 + \bar{t}^2\mathbf{a}_2 + \bar{t}^3\mathbf{a}_3 \tag{6.4}$$

donde $\bar{t} = t/T$ y los coeficientes $\mathbf{a}_i$ $(i = 0, 1, 2, 3)$ son los coeficientes multivariados del polinomio. Esta interpolación produce una trayectoria suave forzando la continuidad de la primera derivada al principio y al final de la ráfaga de pérdidas. Los coeficientes $\mathbf{a}_i$ son calculados a partir de los vectores anterior y posterior a la ráfaga, $\boldsymbol{y}_0$ e $\boldsymbol{y}_T$, y sus primeras derivadas, $\boldsymbol{y}_0^{'}$ e $\boldsymbol{y}_T^{'}$. Tras la obtención de los coeficientes $\mathbf{a}_i$, podemos expresar la ecuación (6.4) como,

$$\hat{\mathbf{x}} = \boldsymbol{y}_0(\bar{t} - 3\bar{t}^2 + 2\bar{t}^3) + \boldsymbol{y}_T(3\bar{t}^2 - 2\bar{t}^3) + \boldsymbol{y}_0^{'}(\bar{t} - 2\bar{t}^2 + \bar{t}^3) + \boldsymbol{y}_T^{'}(\bar{t}^3 - \bar{t}^2) \tag{6.5}$$

donde las primeras derivadas son aproximadas por,

$$\begin{aligned} \boldsymbol{y}_0^{'} &= T(\boldsymbol{y}_0 - \boldsymbol{y}_{-1}) \\ \boldsymbol{y}_T^{'} &= T(\boldsymbol{y}_{T+1} - \boldsymbol{y}_T) \end{aligned} \tag{6.6}$$

Aunque esta técnica emplea una interpolación más suave que la interpolación lineal, Milner y James [153] sólo obtienen mejoras significativas (sobre la interpolación lineal) para condiciones de canal caracterizadas por un 50 % de tasa de pérdidas y una longitud media de 4, 8 y 12 paquetes perdidos. Atendiendo a la revisión de los estudios de tráfico IP llevada a cabo en el capítulo 4, estas condiciones escapan al comportamiento real de una

red de paquetes, por lo que en el presente trabajo sólo consideraremos la interpolación lineal.

En la arquitectura NSR la pérdida de paquetes conlleva una degradación de los vectores de características extraídos de la voz sintetizada posterior a una ráfaga de pérdidas. El decodificador de voz precisa de una cierta memoria de decodificación para la correcta síntesis de la trama actual. Esto origina que, tras una pérdida de paquetes, estas memorias de decodificación se encuentren en un estado corrupto lo que introduce una propagación de error posterior a la pérdida. Gómez *et al.* [152] tratan esta degradación partiendo de la hipótesis de que 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 este supuesto, proponen la adaptación de la técnica *Fixed Codeword-Dependent Cepstral Normalization* (FCDCN) diseñada originalmente para la compensación de ruido acústico [155]. Esta técnica aplica un vector de corrección $\mathbf{r}$ sobre el vector de características ruidoso $\boldsymbol{y}_t$ que depende de la SNR instantánea y del propio vector $\boldsymbol{y}_t$. Concretamente, la dependencia de $\mathbf{r}$ sobre $\boldsymbol{y}_t$ se simplifica mediante la cuantización del espacio vectorial corrupto. De este modo, la estimación resultante $\hat{\mathbf{x}}_t$ se obtiene mediante la siguiente expresión,

$$\hat{\mathbf{x}}_t = \boldsymbol{y}_t + \mathbf{r}(SNR, \mathbf{y}_t) \tag{6.7}$$

donde $\mathbf{y}_t$ es una versión cuantizada de $\boldsymbol{y}_t$. Sin embargo, todavía es necesario modelar la relación señal a ruido (SNR, *Signal to Noise Ratio*) instantánea. A tal efecto, los autores observaron que esta degradación de los vectores de características posteriores a una ráfaga de pérdidas es dependiente de la longitud de la ráfaga previa, l, y de la distancia al final de la ráfaga, t_{ep}. Así, cuanto mayor es l, la SNR instantánea es menor; mientras que a medida que t_{ep} aumenta, la magnitud del error propagado disminuye y, por tanto, la SNR instantánea crece. Finalmente, la estimación propuesta se puede expresar como,

$$\hat{\mathbf{x}}_t = \boldsymbol{y}_t + \mathbf{r}(l, t_{ep}, \mathbf{y}_t) \tag{6.8}$$

Para la obtención de los vectores de corrección se recurre a un entrenamiento *estéreo* a partir de los vectores originales y corruptos. En el ámbito de la compensación de ruido acústico, este tipo de entrenamiento supone una limitación, puesto que el ruido acústico utilizado en esta fase no tiene por qué coincidir con la situación real del sistema. Sin embargo, en la compensación del error de propagación los vectores de características pueden ser precomputados simulando diferentes longitudes de ráfaga. En el mencionado trabajo los autores aplican la compensación del error de propagación mediante FCDCN

en conjunción con la interpolación lineal. Así, en primer lugar se realiza la compensación de los vectores de características posteriores a la ráfaga de pérdidas, para después realizar la interpolación lineal correspondiente a las tramas perdidas.

6.3.4. Resultados Experimentales

Las técnicas propuestas anteriormente han sido evaluadas bajo el marco experimental descrito en el capítulo 4, de modo que los resultados son directamente comparables. Atendiendo al esquema de mitigación propuesto en la sección 6.2, el terminal cliente transmite la voz haciendo uso de un codificador convencional a través de una red de paquetes. Como vimos en los capítulos anteriores, los codificadores de voz más extendidos responden al paradigma de codificación CELP. Estos codificadores consiguen una importante reducción de la tasa de bit a transmitir mediante el empleo de técnicas predictivas. Sin embargo, estas técnicas tienen la desventaja de que introducen dependencias intertrama, lo que hace que el rendimiento de estos codificadores decaiga significativamente cuando son utilizados en una red de paquetes con pérdidas.

Para evaluar el rendimiento de las técnicas de mitigación se han escogido dos de los codificadores CELP más extendidos en la actualidad en este tipo de entornos, AMR (modo 12.2 kbps) y G.729 (8 kbps), de los que podemos encontrar una breve descripción en la sección 4.4. El empaquetamiento seleccionado para los codificadores de voz corresponde a segmentos de 20 ms de voz codificada (2 tramas para G.729 y 1 trama para AMR), de modo que se pueda establecer una comparativa justa entre los resultados de ambos codificadores [142].

La extracción de parámetros relevantes para el reconocimiento (vectores de características) es llevada a cabo mediante el *front-end* estandarizado por ETSI [35]. Este bloque procesa tramas de voz de 25 ms desplazadas 10 ms, por lo que es preciso definir una correspondencia entre el indicador de paquetes perdidos (PLI, *Packet Loss Indicator*) y aquellos vectores que corresponden a él. De este modo, el bloque *mapeador de pérdidas* de la figura 6.1 asocia un indicador de vector perdido (VLI, *Vector Loss Indicator*) a cada vector de características mediante la siguiente expresión,

$$\mathrm{VLI}(t) = \begin{cases} 1 & \text{si PLI}(\lfloor \frac{t}{2} \rfloor) = 1 \text{ ó PLI}(\lfloor \frac{t}{2} + 1 \rfloor) = 1 \\ 0 & \text{en otro caso} \end{cases} \tag{6.9}$$

donde t es el índice temporal de un cierto vector de características. Así, $\mathrm{VLI}(t) = 1$ indica que el vector de características $\boldsymbol{y}_t$ debe ser descartado.

Dentro de las técnicas que hemos expuesto en esta sección, es de especial interés la técnica NFR ya que es la utilizada en los estándares DSR de la ETSI [35, 37, 39, 40]. Esta técnica sustituye los vectores marcados ($\mathrm{VLI}(t) = 1$) por aquellos no marcados ($\mathrm{VLI}(t) = 0$) más próximos, respondiendo a la expresión (6.2). Las tablas 6.1 y 6.2 recogen los resultados obtenidos aplicando NFR sobre una arquitectura NSR basada en AMR 12.2 kbps y G.729 8 kbps, respectivamente. A su vez, los resultados aplicando interpolación lineal son mostrados en las tablas 6.3 y 6.4. Aunque diferentes autores [156, 157] han verificado que la repetición NFR ofrece resultados superiores a la interpolación lineal en sistemas DSR, podemos ver que en el caso de sistemas NSR basados en codificadores CELP la interpolación lineal funciona mejor.

La razón de este hecho puede encontrarse en la propagación de error CELP que distorsiona el primer vector de características no marcado ($\mathrm{VLI}(t) = 0$) tras la pérdida. Así pues, en la repetición, este vector distorsionado se emplea para llevar a cabo la sustitución de la mitad de la ráfaga, mientras que en la interpolación lineal esta dependencia es menor, consiguiendo una mejor reconstrucción.

Gómez *et al.*, en sus trabajos [122, 152], identifica el problema de la propagación de error en el codificador CELP EFR (*Enhanced Full Rate*) [77]. Con el fin de combatir este error, los autores proponen considerarlo como un ruido acústico y corregir la distorsión tras la pérdida mediante una corrección aditiva del tipo FCDCN, para posteriormente utilizar los vectores de características reconstruidos como extremo final de la interpolación. Los resultados obtenidos para esta técnica son los correspondientes a las tablas 6.5 y 6.6, obteniéndose unas notables mejoras sobre la mitigación basada en interpolación y repetición. Estos resultados ponen de manifiesto la necesidad de considerar el error de propagación del códec como una distorsión sobre los parámetros de reconocimiento, de modo que si queremos aumentar el rendimiento del reconocimiento, tendremos que combatir no sólo la pérdida de información sino también la distorsión tras la pérdida.

6.4. Estimación de Mínimo Error Cuadrático Medio

En esta sección proponemos una técnica de reconstrucción basada en el modelado de la voz mediante un HMM y la estimación MMSE (*Minimum Mean Square Error*). Además, este esquema nos permite considerar las diversas degradaciones introducidas por las pérdidas de paquetes en una arquitectura NSR con codificación CELP.

Tasa de Pérdidas	*Long. media ráfaga*			
	1	*2*	*3*	*4*
5 %	97.90	96.90	95.90	95.74
10 %	96.66	94.75	93.33	92.51
15 %	94.65	92.26	90.59	89.31
20 %	91.43	89.77	87.73	86.26

Tabla 6.1: Resultados de precisión de reconocimiento (WAcc) para AMR 12.2 kbps al aplicar repetición NFR.

Tasa de Pérdidas	*Long. media ráfaga*			
	1	*2*	*3*	*4*
5 %	97.88	96.14	95.89	95.58
10 %	96.59	93.76	92.73	91.97
15 %	94.82	90.97	89.71	88.34
20 %	92.66	87.53	86.54	85.00

Tabla 6.2: Resultados de precisión de reconocimiento (WAcc) para G.729A (8 kbps) al aplicar repetición NFR.

Tasa de Pérdidas	*Long. media ráfaga*			
	1	*2*	*3*	*4*
5 %	98.03	97.19	96.36	95.93
10 %	96.99	95.42	94.18	93.06
15 %	95.20	93.54	91.76	89.71
20 %	92.64	91.54	89.55	87.08

Tabla 6.3: Resultados de precisión de reconocimiento (WAcc) para AMR 12.2 kbps al aplicar interpolación lineal.

Tasa de Pérdidas	*Long. media ráfaga*			
	1	*2*	*3*	*4*
5 %	98.17	97.24	96.56	95.99
10 %	97.11	95.62	94.14	93.14
15 %	95.89	93.64	91.44	89.72
20 %	94.30	90.93	89.01	86.74

Tabla 6.4: Resultados de precisión de reconocimiento (WAcc) para G.729 (8 kbps) al aplicar interpolación lineal.

Tasa de Pérdidas	*Long. media ráfaga*			
	1	*2*	*3*	*4*
5 %	98.21	97.64	96.86	96.27
10 %	97.64	96.35	95.03	93.64
15 %	96.46	95.14	92.94	90.49
20 %	94.88	93.59	91.21	87.99

Tabla 6.5: Resultados de precisión de reconocimiento (WAcc) para AMR 12.2 kbps al aplicar reconstrucción FCDN (tras pérdida de paquetes) e interpolación lineal.

Tasa de Pérdidas	*Long. media ráfaga*			
	1	*2*	*3*	*4*
5 %	98.25	97.51	96.70	96.22
10 %	97.51	96.22	95.04	94.12
15 %	96.48	94.90	92.42	90.23
20 %	95.92	93.48	90.69	87.78

Tabla 6.6: Resultados de precisión de reconocimiento (WAcc) para G.729 (8 kbps) al aplicar reconstrucción FCDN (tras pérdida de paquetes) e interpolación lineal.

6.4.1. Fundamentos de la Estimación MMSE

La estimación MMSE consiste en computar el valor esperado del vector de características que se extraería si no hubiera pérdida de paquetes. De forma general, esta estimación puede expresarse del siguiente modo,

$$\hat{\mathbf{x}}_t = E[\boldsymbol{x}_t|\Lambda] \tag{6.10}$$

donde $\boldsymbol{x}_t$ es el vector de características no afectado por la pérdida y Λ representa el conocimiento que se tiene a priori de $\boldsymbol{x}_t$, el cual vendrá dado por el conjunto de datos disponibles, así como por el modelo considerado para la evolución temporal de la voz.

Para llevar a cabo el cómputo del valor esperado, consideraremos un conjunto de vectores prototipo $\{\mathbf{x}^{(i)}; i = 0, \ldots, N-1\}$ que representa el espacio de características original. Así, $\mathbf{x}_t$ es el resultado de cuantizar el vector original $\boldsymbol{x}_t$ con este conjunto de vectores prototipo. El efecto de la transmisión de la voz codificada a través de la red de paquetes es que el vector de características $\boldsymbol{y}_t$ extraído de la voz decodificada puede diferir del original $\boldsymbol{x}_t$ debido a la pérdida de paquetes. Por tanto, la estima MMSE del vector

correspondiente al momento t puede ser obtenida como,

$$\bar{\mathbf{x}}_t = E[\mathbf{x}_t|Y] = \sum_{i=0}^{N-1} \mathbf{x}^{(i)} P(\mathbf{x}_t = \mathbf{x}^{(i)}|Y) \tag{6.11}$$

donde Y representa los datos disponibles y $P(\mathbf{x}_t = \mathbf{x}^{(i)}|Y)$ es la probabilidad condicional de que el vector original cuantizado $\mathbf{x}_t$ se corresponda con el vector prototipo $\mathbf{x}^{(i)}$ dados los datos disponibles Y.

Un ejemplo sencillo de estimación MMSE puede ser obtenido considerando $Y = \boldsymbol{y}_t$ ($\mathcal{M} = \mathcal{N} = 0$) [158],

$$\bar{\mathbf{x}} = \sum_{i=0}^{N-1} \mathbf{x}^{(i)} P(\mathbf{x}_t = \mathbf{x}^{(i)}|\boldsymbol{y}_t) \tag{6.12}$$

$$P(\mathbf{x}_t = \mathbf{x}^{(i)}|\boldsymbol{y}_t) = \frac{P(\boldsymbol{y}_t|\mathbf{x}_t = \mathbf{x}^{(i)})P_i}{P(\boldsymbol{y}_t)} \tag{6.13}$$

donde la fuente se caracteriza mediante un sencillo modelo a partir de las probabilidades *a priori* $P_i = P(\mathbf{x}_t = \mathbf{x}^{(i)})$ de los distintos prototipos.

Sin embargo, durante una ráfaga de pérdidas no existen datos disponibles del canal y, por tanto, no disponemos de observaciones. Para solventar este problema, Gómez *et al.* [159] proponen la siguiente estimación para una arquitectura DSR,

$$\bar{\mathbf{x}}_t = E[\mathbf{x}_t|Y^-, Y^+] = \sum_{i=0}^{N-1} \mathbf{x}^{(i)} P(\mathbf{x}_t = \mathbf{x}^{(i)}|Y^-, Y^+) \qquad 1 \leq t \leq T \tag{6.14}$$

donde Y^- e Y^+ representan la secuencia de datos disponibles con anterioridad y posterioridad a la ráfaga, respectivamente. Puesto que la expresión (6.14) no utiliza ninguna información instantánea, las estimas pueden ser precomputadas y almacenadas, de modo que el correspondiente algoritmo se reduce a la consulta de una tabla indexada por los vectores de características cuantificados Y^- e Y^+. Retomando la notación introducida en la sección 6.3.3, el principal problema de este tipo de estimación reside en que la tabla de estimas adopta tamaños enormes cuando $\mathcal{M} > 1$ o $\mathcal{N} > 1$. Adicionalmente, el problema de la propagación de error presente en los codificadores CELP (arquitectura NSR) deteriora las observaciones Y^+, lo que produciría un bajo rendimiento de esta técnica de reconstrucción.

6.4.2. Modelado HMM de la Voz

Para mejorar la estimación MMSE propuesta en (6.12) puede emplearse un modelo de voz de primer orden donde no sólo se tengan en cuenta las probabilidades a priori, sino también las redundancias temporales contenidas en la parametrización de la voz. Este objetivo se puede lograr mediante el uso de un modelo oculto de Markov (HMM), tal y como proponen Peinado *et al.* [156]. En este caso, cada prototipo $\mathbf{x}^{(i)}$ del espacio no corrupto se asocia a un estado, de forma que las probabilidades de transición entre estados a_{ij} vienen dadas por las probabilidades de transición entre prototipos. A su vez, la secuencia de vectores observados $(\boldsymbol{y}_0, \ldots, \boldsymbol{y}_T)$ es cuantizada obteniéndose $Y = (\mathbf{y}_0, \ldots, \mathbf{y}_T)$. Esta cuantización de las observaciones nos permite utilizar distribuciones de probabilidad discretas, estableciendo una implementación más sencilla que en el caso de utilizar un modelo HMM continuo. De este modo, la caracterización del modelo viene dada por las probabilidades de observación $b_i(\mathbf{y})$ y las probabilidades de transición a_{ij}, definidas como,

$$b_i(\mathbf{y}_t) \equiv P(\mathbf{y}_t|\mathbf{x}_t = \mathbf{x}^{(i)}) \tag{6.15}$$

$$a_{ij} \equiv P(\mathbf{x}_t = \mathbf{x}^{(j)}|\mathbf{x}_{t-1} = \mathbf{x}^{(i)}) \tag{6.16}$$

Aunque sería posible obtener un cuantizador específico para el espacio de vectores de características corrupto, en este trabajo se utilizará para este propósito el conjunto de vectores prototipo dado por $\{\mathbf{x}^{(i)}; i = 0, \ldots, N-1\}$. Esta consideración nos permite simplificar el desarrollo de la estimación MMSE sin degradar los resultados siempre que el conjunto de prototipos nos proporcione una representación precisa del espacio de características limpio. No obstante, en esta sección estudiaremos cómo afrontar las posibles degradaciones introducidas por esta cuantización.

Las probabilidades de transición pueden ser obtenidas a partir de una base de datos de entrenamiento, mientras que las probabilidades de observación dependen de las degradaciones introducidas por el canal. Partiendo del modelado HMM de las redundancias de la voz, podemos obtener la estima del parámetro extraído en el instante t dados los vectores de características observados $Y = (\mathbf{y}_0, \ldots, \mathbf{y}_T)$ como,

$$\bar{\mathbf{x}}_t = \sum_{i=0}^{N-1} \mathbf{x}^{(i)} \gamma_t(i) \qquad 1 \leq t \leq T \tag{6.17}$$

donde,

$$\gamma_t(i) = P(\mathbf{x}_t = \mathbf{x}^{(i)}|Y) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=0}^{N-1} \alpha_t(j)\beta_t(j)} \qquad 0 \leq i < N \tag{6.18}$$

que es computada a partir de las probabilidades condicionales hacia delante, $\alpha_t(i)$, y hacia atrás, $\beta_t(i)$, definidas mediante las siguientes expresiones,

$$\alpha_t(i) = P(\mathbf{x}_t = \mathbf{x}^{(i)}|\mathbf{y}_0, \ldots, \mathbf{y}_t) \tag{6.19}$$

$$\beta_t(i) = P(\mathbf{y}_{t+1}, \ldots, \mathbf{y}_T|\mathbf{x}_t = \mathbf{x}^{(i)}) \tag{6.20}$$

El cómputo de estas probabilidades puede ser realizado de forma recursiva [160] mediatne el algoritmo de *forward-backward* del siguiente modo,

$$\alpha_t(i) = \left[\sum_{j=0}^{N-1} a_{ji}\alpha_{t-1}(j)\right] b_i(\mathbf{y}_t) \tag{6.21}$$

$$\beta_t(i) = \sum_{j=0}^{N-1} a_{ij}\beta_{t+1}(j)b_j(\mathbf{y}_{t+1}) \tag{6.22}$$

Por tanto, ya sólo restan por determinar las condiciones iniciales, $\alpha_0(i)$ y $\beta_T(i)$, y las probabilidades de observación $b_i(\mathbf{y})$ para poder determinar la reconstrucción MMSE. Sin embargo, la inicialización del algoritmo, así como la determinación de las probabilidades de observación dependerá del tipo de degradación de canal considerada, tal y como se muestra en el siguiente apartado.

6.4.3. Distorsiones Producidas por las Pérdidas

Peinado *et al.*, en sus trabajos [156, 161], aplican la estimación MMSE basada en HMM sobre canales digitales inalámbricos para un sistema de reconocimiento remoto DSR. En este entorno, los vectores de características recibidos durante una ráfaga de errores no son totalmente fiables por lo que sus probabilidades de observación dependen del nivel de degradación. Así, la información de fiabilidad es suministrada por el decodificador de canal, a partir de la cual se obtienen las probabilidades de observación necesarias para la reconstrucción de los vectores de características.

En canales con pérdidas de paquetes la situación es completamente distinta, puesto que durante el intervalo de la pérdida no existen datos disponibles provenientes del canal. En el caso de una arquitectura DSR, donde la información empaquetada consiste directamente

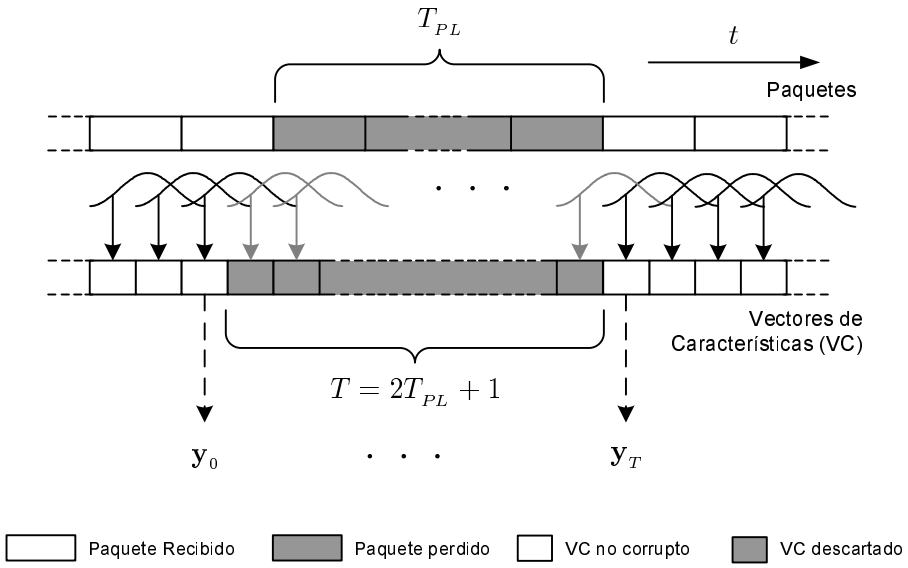


Figura 6.2: Diagrama resultante de sólo considerar los vectores de características (VC) descartados por la pérdida.

en los vectores de reconocimiento, las pérdidas se traducen en la ausencia de un cierto segmento de observaciones. La figura 6.2 muestra un esquema de esta situación aplicada a una arquitectura NSR.

En esta aproximación se considera que se dispone de información fiable antes y después de la ráfaga, de modo que la probabilidad de observación se determina como,

$$b_i(\mathbf{y}_t) = \begin{cases} 0 & \mathbf{y}_t \neq \mathbf{x}^{(i)} \\ 1 & \mathbf{y}_t = \mathbf{x}^{(i)} \end{cases} \qquad t = 0;\, t = T \tag{6.23}$$

mientras que durante la ráfaga, al no disponer de una observación válida, la probabilidad de observación adopta una función de distribución uniforme,

$$b_i(\mathbf{y}_t) = \frac{1}{N} \qquad 0 < t < T \tag{6.24}$$

donde N es el número de posibles centroides utilizados para obtener $\mathbf{y}_t$ a partir de $\boldsymbol{y}_t$. Consecuentemente, el algoritmo de estimación MMSE se guía únicamente por las probabilidades de transición en la reconstrucción de aquellos vectores correspondientes a la ráfaga.

Esta es la solución adoptada por Ion y Haeb-Umbach [162] para un sistema DSR en redes con pérdidas. Además, estos autores extienden esta solución para sistemas NSR en su trabajo [163], tanto para aquellos que utilizan codificadores de voz con independencia

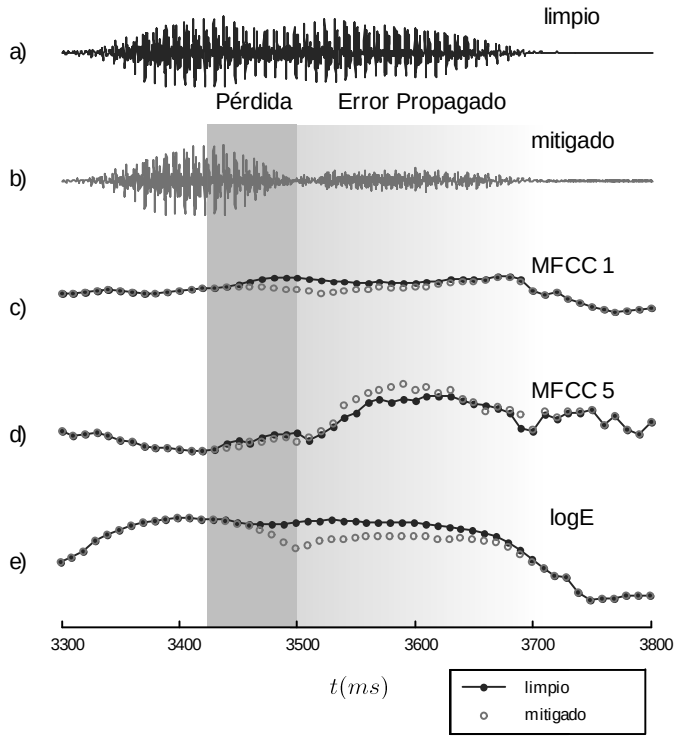


Figura 6.3: Impacto de una pérdida de paquetes sobre la síntesis de voz de un decodificador CELP (AMR 12.2 kbps) y la posterior extracción de características de reconocimiento. a) Voz decodificada sin pérdidas (*limpia*); b) Voz decodificada (*mitigada*) ante una pérdida de paquetes utilizando el algoritmo PLC incluido en el decodificador; c), d) y e) Características de reconocimiento extraídas a partir de las formas de onda correspondientes a la síntesis *limpia* y *mitigada*.

intertrama (G.711) como con dependencias (codificadores CELP G.723 y G.729.1).

En nuestro trabajo consideramos un modelo de degradación distinto. Como ya vimos en el capítulo anterior, los codificadores que introducen una cierta dependencia intertrama presentan el problema de la propagación de error. La correcta síntesis de voz está condicionada a la correcta decodificación de las tramas anteriores a la actual. Así, cuando se produce una pérdida de paquetes, la desadaptación existente entre la voz sintética generada por el algoritmo PLC y la voz correctamente decodificada origina un error que es propagado hacia delante por los esquemas predictivos del codificador de voz. En consecuencia, si esta señal es utilizada para llevar a cabo la extracción de los parámetros de reconocimiento, éstos estarán afectados por esta distorsión.

La figura 6.3 muestra un ejemplo de cómo afecta una ráfaga de tramas perdidas sobre la decodificación en un esquema CELP. Concretamente, esta figura se corresponde con la simulación de una pérdida de 4 tramas consecutivas en el intervalo 3420-3500 ms. En los

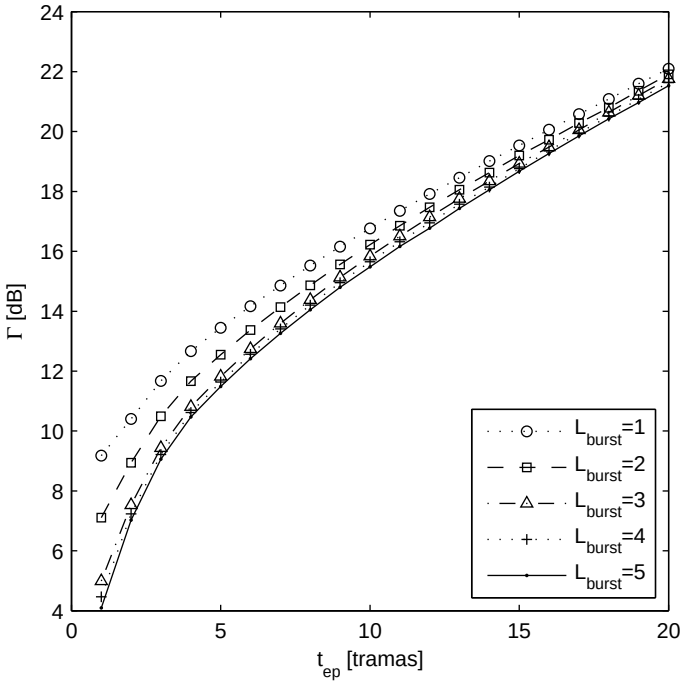


Figura 6.4: Relación en dB entre la varianza del parámetro log-Energía extraído de la síntesis *limpia* (sin pérdida de paquetes) y el error cuadrático medio originado por diferentes longitudes de ráfaga de pérdidas (L_{burst}) medido en las tramas t_{ep} posteriores a la pérdida.

dos primeros gráficos se muestran las formas de onda correspondientes a una transmisión *limpia* (sin paquetes perdidos) y la señal sintética *mitigada* generada por el algoritmo PLC integrado en el decodificador. Los siguientes trazos se corresponden con diferentes parámetros de reconocimiento extraídos a partir de la señal limpia y mitigada. Claramente podemos discernir entre dos efectos distintos. Por un lado, el algoritmo PLC del decodificador genera la señal mitigada correspondiente a la zona de pérdida, de modo que los parámetros de reconocimiento extraídos a partir de ésta presentan una distorsión. Tal y como comentamos con anterioridad, estos algoritmos PLC intentan mitigar los errores producidos en la síntesis teniendo en cuenta consideraciones perceptuales, aplicando técnicas como la repetición y apagado progresivo, que no son adecuadas para el reconocimiento de voz. Por último, las características extraídas a partir de los segmentos de voz sintetizada posteriores a la pérdida se ven afectadas por el error propagado, el cual puede llegar a tener una duración incluso superior a la del segmento perdido.

Con el fin de ilustrar el impacto del error de propagación sobre las características necesarias para el reconocimiento, la figura 6.4 muestra la relación (expresada en dB)

entre la varianza del parámetro log-Energía extraída a partir de la síntesis sin pérdidas, σ_x^2, y el error cuadrático medio que se origina por el error propagado tras una pérdida de paquetes. Esta relación puede ser expresada del siguiente modo,

$$\Gamma(t_{ep}) = 10\log_{10}\frac{\sigma_x^2}{E\left[(x_{t_{ep}} - y_{t_{ep}})^2\right]} \tag{6.25}$$

donde $x_{t_{ep}}$ e $y_{t_{ep}}$ representan los parámetros log-Energía *limpio* (sin pérdida) y *corrupto* (tras pérdida) correspondientes a la trama t_{ep} después de una pérdida de paquetes, respectivamente. Los resultados se obtuvieron utilizando el codificador AMR (modo 12.2 kbps, empaquetando 1 trama por paquete). El experimento consistió en simular sistemáticamente varias longitudes de ráfaga de paquetes perdidos (L_{burst}) sobre una base de datos de entrenamiento y promediar las distorsiones que se originaban en los parámetros correspondientes a la zona posterior a la ráfaga.

En la figura 6.4 se observa que existe una dependencia entre el número de paquetes perdidos consecutivos y el error de propagación originado por la pérdida. Este resultado es consistente, puesto que cuanto mayor es la longitud de la ráfaga, mayor es la desadaptación originada en la síntesis del algoritmo PLC y, por tanto, mayor es el impacto de la propagación de error. Otro efecto observado es que el error propagado se reduce a medida que nos distanciamos del final de la pérdida. De esta forma, la recepción de los parámetros correctos por parte del decodificador origina una resincronización que produce la progresiva desaparición del error propagado.

Podemos concluir que, al contrario de lo que ocurría en otro tipo de aplicaciones donde la degradación es aproximadamente constante durante una ráfaga de errores [164, 165], las pérdidas de paquetes en un sistema NSR originan diferentes tipos de datos corruptos. En primer lugar, tendremos que considerar que durante la ráfaga de paquetes perdidos no dispondremos de información proveniente del canal. Además, el error propagado por el codificador afectará a un cierto número de vectores de características tras la pérdida, de forma que no podremos considerarlos como parámetros totalmente correctos. La distorsión que aparecerá debida a la propagación de error varía (decrece) con el tiempo y su magnitud dependerá de la longitud de la ráfaga de paquetes perdidos. Estas consideraciones pueden ser introducidas en la reconstrucción MMSE formulada en el apartado previo, mediante probabilidades de observación que varían en función de los diferentes tipos y niveles de degradación.

La figura 6.5 muestra un diagrama de cómo una ráfaga de T_{PL} paquetes perdidos afecta a la extracción de vectores de características. Con carácter general, consideraremos que

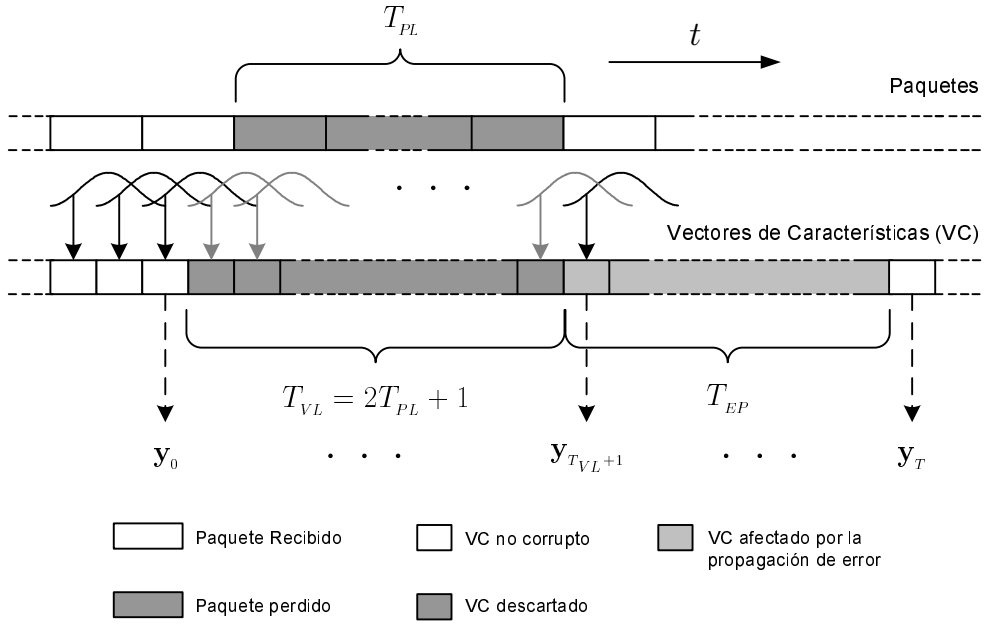


Figura 6.5: Modelado de la distorsión de canal introducida por la pérdida de información y la propagación de error de los codificadores CELP.

la secuencia de vectores afectados, de alguna manera, por las degradaciones introducidas por una pérdida de paquetes, es representada mediante $Y = (\mathbf{y}_0, \ldots, \mathbf{y}_T)$, donde $\mathbf{y}_0$ es el último vector extraído correctamente antes de la pérdida e $\mathbf{y}_T$ representa al primer vector no corrupto obtenido tras la ráfaga. A su vez, los vectores comprendidos entre $t = 1$ y $t = T_{VL}$ corresponden a la ráfaga de pérdidas. Puesto que estos vectores son extraídos a partir de la voz generada por el algoritmo PLC integrado en el decodificador, los consideraremos como información no fiable. Adicionalmente, consideraremos que la propagación de error intrínseca al proceso de decodificación origina una distorsión en los siguientes T_{EP} vectores posteriores a la ráfaga, de modo que $T = T_{VL} + T_{EP} + 1$. Con el objetivo de modelar correctamente esta degradación variable utilizaremos una distribución de probabilidad de observación variable, $b_i^{(t)}(\mathbf{y}_t)$, para cada instante t $(t = 0, \ldots, T)$ y definida por los siguientes casos:

- Suponiendo que los vectores en los instantes $t = 0$ y $t = T$ han sido obtenidos a partir de voz correctamente decodificada, las probabilidades de observación deben de establecerse del siguiente modo,

$$b_i^{(t)}(\mathbf{y}_t) = \begin{cases} 1 & \mathbf{y}_t = \mathbf{x}^{(i)} \\ 0 & \mathbf{y}_t \neq \mathbf{x}^{(i)} \end{cases} \qquad t = 0;\ t = T \tag{6.26}$$

Es decir, consideramos la información recibida como totalmente fiable. Además, debemos contemplar la posibilidad de que los vectores $\mathbf{y}_0$ e $\mathbf{y}_T$ no siempre se extraerán a partir de voz correctamente decodificada, ya que serían corruptos en el caso de ráfagas al principio o final de la frase a reconocer. Igualmente, podría suceder una pérdida de paquetes antes de que el error propagado de una pérdida previa se haya extinguido. En estas situaciones, tendremos que considerar alguno de los siguientes casos.

- En los instantes de tiempo comprendidos dentro del intervalo delimitado por $t = 1$ y $t = T_{VL}$, es decir, durante la ráfaga, las probabilidades de observación se establecen como,

$$b_i^{(t)}(\mathbf{y}_t) = \frac{1}{N} \qquad 1 \leq t \leq T_{VL} \tag{6.27}$$

  Puesto que no se obtienen observaciones válidas para estos instantes temporales, la probabilidad de observación se toma como una distribución uniforme para todos los posibles estados $\mathbf{x}^{(i)}$. En este caso, las probabilidades de observación no proporcionan ninguna información a la estimación MMSE, por lo que algoritmo *forward-backward* se guiará únicamente por las probabilidades de transición.

- Finalmente prestamos atención a aquellos vectores de características que son parcialmente fiables debido a que están afectados por el error de propagación. Como se mostró con anterioridad (véase figura 6.4), la propagación de error causa una distorsión mayor en los primeros vectores tras una ráfaga de paquetes perdidos que, posteriormente, disminuye a medida que el tiempo avanza. Además, el nivel de distorsión presenta una dependencia con la longitud del paquete perdido. Por estos motivos, modelamos la probabilidad de observación del siguiente modo,

$$b_i^{(t)}(\mathbf{y}_t = \mathbf{x}^{(j)}) = P(\mathbf{y}_t = \mathbf{x}^{(j)} | \mathbf{x}^{(i)}, l_{burst}, t_{ep}) \qquad T_{VL} < t < T \tag{6.28}$$

  donde l_{burst} es la longitud de la ráfaga de pérdidas y $t_{ep} = t - T_{VL}$ se corresponde con el instante de tiempo desde el final de la ráfaga. En la práctica, las probabilidades de observación se pueden calcular a partir de una base de datos de entrenamiento estéreo obtenida mediante la simulación sistemática de ráfagas de diferentes longitudes de paquetes perdidos. Así, dada una ráfaga de longitud l_{burst}, las probabilidades de observación para el instante $t = T_{VL} + t_{ep}$ son obtenidas mediante las siguientes

frecuencias de aparición,

$$b_i^{(t)}(\mathbf{y}_t = \mathbf{x}^{(j)}) = \frac{n_{j|i}}{\sum_i n_{j|i}} \qquad T_{VL} < t < T \tag{6.29}$$

donde $n_{j|i}$ es el número de veces que $\mathbf{y}_t = \mathbf{x}^{(j)}$ dado que $\mathbf{x}_t = \mathbf{x}^{(i)}$ para unos valores de l_{burst} y t_{ep} dados. Cuando la longitud de una pérdida es superior a un cierto número de L_{burst} vectores, el incremento en distorsión puede ser considerado despreciable. Adicionalmente, podemos considerar que la propagación de error desaparece una vez superado un cierto número T_{EP} de vectores de características tras la pérdida. De este modo, sólo precisamos almacenar $L_{burst} \times T_{EP}$distribuciones de probabilidad diferentes. En el caso de que una ráfaga verifique $l_{burst} > L_{burst}$, se aplicarán las distribuciones correspondientes a L_{burst}.

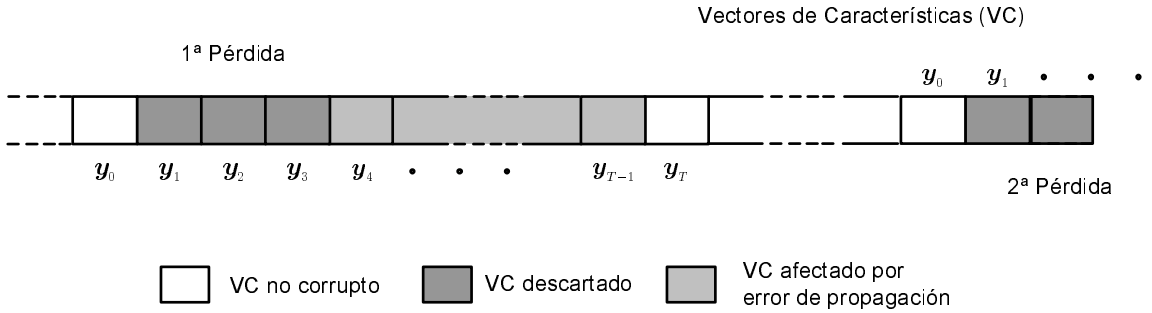
6.4.4. Inicialización de la Estimación MMSE

Para poder obtener las probabilidades condicionales hacia delante, $\alpha_t(i)$, y hacia atrás, $\beta_t(i)$, es necesario establecer un conjunto de condiciones iniciales para las recursiones establecidas en las ecuaciones (6.21) y (6.22). Inicialmente podemos considerar una pérdida de paquetes lo suficientemente distante de las pérdidas previa y posterior, de modo que no exista solape entre los errores de propagación. Este es el caso representado en la figura 6.6a, en el que se aplicarían las siguientes condiciones iniciales en los instantes temporales $t = 0$ y $t = T$,

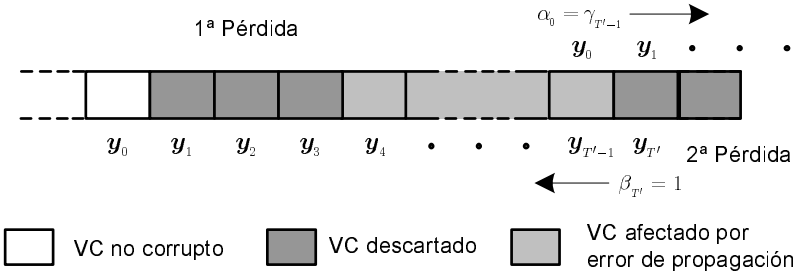
$$\begin{aligned} \alpha_0(i) &= P_i b_i^{(t)}(\mathbf{y}_0)/K_0 \\ \beta_T(i) &= 1 \end{aligned} \qquad (i = 0, 1, \ldots, N-1) \tag{6.30}$$

donde P_i es la probabilidad a priori del prototipo $\mathbf{x}^{(i)}$. Obsérvese que en este caso $b_i^{(t)}(\mathbf{y}_t)$ sigue la expresión (6.26).

No obstante, podría suceder una nueva pérdida de paquetes antes de que la propagación de error producida por la pérdida actualmente considerada haya desaparecido. Este caso se corresponde con el mostrado en la figura 6.6b donde tomamos las mismas condiciones iniciales consideradas en (6.30), pero tomando como punto de inicialización de las probabilidades condicionales $\beta_t(i)$ el instante T', correspondiente al primer vector descartado de la siguiente pérdida, en lugar de T. Consecuentemente, la reconstrucción MMSE finaliza en el instante $T' - 1$, es decir, justo en el vector que se considerará como



(a) Sin solape.



(b) Con solape.

Figura 6.6: Casos en la inicialización del algoritmo *Forward-Backward.*

$\mathbf{y}_0$ en la mitigación de la siguiente pérdida donde utilizaremos como valor de inicialización de la probabilidad condicional hacia delante $\alpha_0(i) = \gamma_{T'-1}(i)$.

6.4.5. Reconstrucción MMSE

La estimación MMSE propuesta en la ecuación (6.17) computa un vector de características reconstruido $\bar{\mathbf{x}}$ a partir de la secuencia de vectores cuantizados Y y el conjunto de prototipos $\{\mathbf{x}^{(i)}; i = 0, \ldots, N-1\}$. Sin embargo, esta reconstrucción puede originar una pérdida de rendimiento debido a la cuantización aplicada. Para solventar este problema, en lugar de utilizar directamente la estima MMSE de la ecuación (6.17), aplicaremos la siguiente reconstrucción,

$$\hat{\mathbf{x}}_t = \boldsymbol{y}_t + \mathbf{r}(\boldsymbol{y}_t) \tag{6.31}$$

donde $\mathbf{r}(\boldsymbol{y})$ es un factor de corrección aditivo que es función del vector de cuantización $\boldsymbol{y}$, que podemos obtener mediante la siguiente aproximación,

$$\mathbf{r}(\boldsymbol{y}_t) \simeq \mathbf{r}(\mathbf{y}_t) = \bar{\mathbf{x}}_t - \mathbf{y}_t \tag{6.32}$$

Es decir, suponemos que $\mathbf{r}(\boldsymbol{y}_t)$ es una función de variación suave con $\boldsymbol{y}_t$.

Aunque este método de reconstrucción podría ser aplicado tanto a las componentes estáticas como dinámicas del vector de características, sólo se aplica a las estáticas, obteniendo posteriormente las componentes delta reconstruidas como,

$$\Delta\hat{x}_t(k) = \sum_{l=-W_\Delta}^{W_\Delta} \omega_l \hat{x}_{t+l}(k) \tag{6.33}$$

donde $\hat{x}_t(k)$ es la secuencia reconstruida de la componente k-ésima del vector de características estático y ω_l ($l = -W_\Delta, \ldots, W_\Delta$) es el conjunto de pesos utilizado para el cómputo de la correspondiente característica dinámica $\Delta\hat{x}_t(k)$. Del mismo modo, se puede obtener una expresión similar para las componentes delta-delta del vector de características.

6.4.6. Resultados Experimentales

Los resultados presentados en esta sección son obtenidos bajo el marco experimental que se concretó en el apartado 6.3.4, de modo que se establecen las mismas bases para establecer una comparativa justa entre el rendimiento de las diferentes técnicas de mitigación presentadas.

Para poder aplicar las técnicas propuestas en esta sección, es necesario establecer un conjunto de prototipos que represente correctamente el espacio de características. A tal efecto hemos utilizado los centroides del diccionario de cuantización SVQ (*Split Vector Quantization*) definido en el estándar DSR [35]. Este conjunto de vectores proporciona una representación precisa del espacio de características [47] agrupando las mismas en pares y cuantizándolos por medio de siete diccionarios SVQ con 64 centroides, excepto el asociado a la energía y MFCC(0) que presenta 256 centroides. Esta cuantización por pares conlleva, sin pérdida de generalidad, que las estimaciones presentadas a lo largo de esta sección realmente se apliquen siete veces (una por cada cuantización SVQ) obteniéndose un conjunto de probabilidades $\gamma_t(i)$ para cada par de características.

Además del modelado de la voz mediante un HMM, en esta sección se han presentado diferentes modelos de distorsión introducidos por las pérdidas. El modelo más simple es el que considera únicamente la pérdida de información (véase figura 6.2), y que, por tanto, obtiene las probabilidades de observación mediante las expresiones (6.23) y (6.24). Las tablas 6.7 y 6.8 muestran los resultados obtenidos al realizar la estimación MMSE sólo teniendo en cuenta el descarte de vectores correspondiente a la pérdida. Esta propuesta fue originalmente ideada para la arquitectura DSR por Ion y Haeb-Umbach [162] que

posteriormente extendieron a NSR [163]. Aunque los resultados obtenidos consiguen una notable mejora frente a los resultados de referencia mostrados en el capítulo 4, son aproximadamente iguales a los obtenidos por la interpolación lineal presentados en la sección anterior. Al establecer confianza absoluta a los instantes anterior y posterior a una ráfaga de pérdidas y no disponer de observaciones durante la ráfaga, la estimación MMSE tiende a aproximarse a una interpolación lineal entre ambos extremos, lo que justifica la similitud entre ambos resultados.

No obstante, este modelo se puede mejorar si lo adaptamos al caso concreto de la arquitectura NSR. Como vimos con anterioridad, la pérdida de paquetes conlleva no sólo la pérdida de información, sino también una propagación de error sobre los vectores de características posteriores a una pérdida. Así, el cómputo de las probabilidades de observación descrito en el apartado 6.4.3, e ilustrado en la figura 6.5, nos permite considerar diferentes errores de propagación dependiendo de la longitud de la pérdida y la posición relativa tras ésta. Como se mostró en la figura 6.5, la distorsión producida por el error de propagación es cada vez menor a medida que nos distanciamos de la pérdida, de modo que, independientemente del tamaño de ésta, tras una distancia de $T_{EP} = 20$ vectores podemos considerar que la distorsión ha desaparecido. Además, se ha considerado una máxima longitud de ráfaga $L_{burst} = 5$ paquetes, ya que el incremento de distorsión provocado por ráfagas mayores puede ser considerado despreciable. Los resultados obtenidos al considerar las distorsiones originadas por el error de propagación se recogen en las tablas 6.9 y 6.10 para los codificadores AMR y G.729, respectivamente, donde podemos observar cómo la estimación MMSE mejora claramente sus prestaciones al considerar las distorsiones originadas por el error de propagación, frente a no considerarlas (tablas 6.7 y 6.8).

En los casos anteriores los vectores de características perdidos o corruptos eran reemplazados directametne por las estimas MMSE. Aunque la estimación utiliza una representación del espacio vectorial suficientemente precisa, es posible obtener una mejora de los resultados si utilizamos un factor de corrección aditivo, como el empleado por la técnica FCDCN, obtenido a partir de la estimación MMSE. Estos factores aditivos se pueden computar fácilmente tal y como se describe en el apartado 6.4.5. Este tipo de reconstrucción se traduce en un incremento de las tasas de reconocimiento mostrado en las tablas 6.11 y 6.12 para los codificadores AMR y G.729, respectivamente. Además, el hecho de que la estimación MMSE utilice información sobre las redundancias temporales de la voz (modelado HMM) consigue obtener unos resultados superiores a los conseguidos por Gómez *et al.* al aplicar la técnica de reconstrucción FCDCN [122].

Tasa de Pérdidas	*Long. media ráfaga*			
	1	*2*	*3*	*4*
5 %	97.99	97.19	96.40	95.91
10 %	96.87	95.62	94.41	93.09
15 %	95.11	93.72	91.93	89.99
20 %	92.42	91.64	89.58	87.33

Tabla 6.7: Resultados de precisión de reconocimiento (WAcc) para AMR 12.2 kbps al aplicar la reconstrucción basada en estimación MMSE sin considerar la propagación de error.

Tasa de Pérdidas	*Long. media ráfaga*			
	1	*2*	*3*	*4*
5 %	98.21	97.23	96.53	95.88
10 %	97.12	95.65	94.21	92.97
15 %	95.82	93.59	91.51	89.89
20 %	94.23	90.99	89.02	86.87

Tabla 6.8: Resultados de precisión de reconocimiento (WAcc) para G.729 8 kbps al aplicar la reconstrucción basada en estimación MMSE sin considerar la propagación de error.

Tasa de Pérdidas	*Long. media ráfaga*			
	1	*2*	*3*	*4*
5 %	98.28	97.63	96.73	96.27
10 %	97.36	96.50	95.11	93.56
15 %	96.16	94.98	92.86	90.88
20 %	94.36	93.17	90.74	87.94

Tabla 6.9: Resultados de precisión de reconocimiento (WAcc) para AMR 12.2 kbps al aplicar la reconstrucción basada en estimación MMSE considerando la propagación de error.

Tasa de Pérdidas	*Long. media ráfaga*			
	1	*2*	*3*	*4*
5 %	98.40	97.85	97.09	96.38
10 %	97.96	96.68	95.25	93.81
15 %	97.15	95.25	92.92	90.77
20 %	96.26	93.45	90.64	88.11

Tabla 6.10: Resultados de precisión de reconocimiento (WAcc) para G.729 8 kbps al aplicar la reconstrucción basada en estimación MMSE considerando la propagación de error.

Tasa de Pérdidas	*Long. media ráfaga*			
	1	*2*	*3*	*4*
5 %	98.33	97.77	97.04	96.60
10 %	97.62	96.73	95.62	94.12
15 %	96.54	95.49	93.46	91.67
20 %	94.63	93.71	91.76	89.13

Tabla 6.11: Resultados de precisión de reconocimiento (WAcc) para AMR 12.2 kbps al aplicar la reconstrucción aditiva basada en estimación MMSE considerando las distorsiones originadas por el error de propagación.

Tasa de Pérdidas	*Long. media ráfaga*			
	1	*2*	*3*	*4*
5 %	98.43	97.93	97.26	96.64
10 %	97.99	96.90	95.68	94.52
15 %	97.44	95.72	93.54	91.68
20 %	96.51	94.11	91.54	89.28

Tabla 6.12: Resultados de precisión de reconocimiento (WAcc) para G.729 8 kbps al aplicar la reconstrucción aditiva basada en estimación MMSE considerando las distorsiones originadas por el error de propagación.

6.5. Técnicas *Soft-Data*

Normalmente, se supone que todas las observaciones de una cierta secuencia de vectores de características son igualmente relevantes para el proceso de reconocimiento. Sin embargo, hay situaciones en las que algunas de estas observaciones no tienen certeza absoluta. Un posible ejemplo vendría dado por la transmisión de las observaciones por un canal digital ruidoso, en el que los errores de transmisión afectarían a ciertas características. Otra situación similar ocurre cuando se captura la voz en presencia de ruido acústico. De este modo, algunos vectores de características, extraídos de zonas de baja energía en la frase, estarán más afectados que otros, provenientes de zonas de alta energía.

En nuestro caso, las técnicas, hasta ahora propuestas, tienen como objetivo la sustitución de la secuencia de vectores de parámetros corruptos $\boldsymbol{y}_t$ por una cierta reconstrucción $\hat{\mathbf{x}}_t$. Sin embargo, las reconstrucciones resultantes no son totalmente fiables. Si podemos, de algún modo, obtener una medida del nivel de confianza de cada uno de los parámetros de cada reconstrucción, parece razonable modificar el algoritmo de reconocimiento de voz, de modo que aquellos parámetros reconstruidos cuya confianza sea mayor, tengan más peso que los menos fiables. En este sentido, la estimación MMSE propuesta en la

sección anterior no sólo se presenta como una eficaz herramienta para la reconstrucción de los vectores de características, sino que además nos ofrece información, a través de las distribuciones de probabilidad $\gamma_t(i)$ (véase ecuación (6.18)) utilizadas en la estimación, de la confianza de cada una de las reconstrucciones.

La idea principal de la técnica de mitigación tratada en esta sección consiste en aprovechar los potentes modelos estadísticos integrados en el reconocedor para mejorar el reconocimiento en caso de pérdida de paquetes. De este modo, en lugar de tratar la reconstrucción como un dato determinista (*hard-decision*), consideraremos que la reconstrucción presenta una cierta distribución de probabilidad (*pdf*) asociada (*soft-data*) [165].

6.5.1. Modificación de la Probabilidad de Observación

Tal y como se presentó en la sección 2, el reconocimiento de voz basado en HMM lleva a cabo su tarea aplicando el algoritmo de Viterbi a un único macromodelo λ. Este algoritmo establece el camino o secuencia de estados $Q = (q_1, \ldots, q_T)$ que maximiza la probabilidad $P(Q|X, \lambda)$ (véase la expresión (2.28)) dado un conjunto de observaciones $X = (\mathbf{x}_1, \ldots, \mathbf{x}_T)$.

Por conveniencia, atendiendo a la formulación y notación propuesta para los HMM en la sección 2.5.1, también podemos reescribir $P(Q|X, \lambda)$ como,

$$\begin{aligned} P(Q|X,\lambda) &= \frac{P(X,Q|\lambda)}{P(X|\lambda)} \\ &= \frac{1}{P(X|\lambda)}\pi_{q_1} b_{q_1}(\mathbf{x}_1) a_{q_1 q_2} b_{q_2}(\mathbf{x}_2) a_{q_2 q_3} \cdots a_{q_{T-1} q_T} b_{q_T}(\mathbf{x}_T) \\ &= C \prod_{t=1}^{T} a_{q_{t-1} q_t} b_{q_t}(\mathbf{x}_t) \end{aligned} \tag{6.34}$$

donde hemos considerado que el estado q_0 es una estado inicial nulo (π_{q_1} se corresponde con la probabilidad de transición $a_{q_0 q_1}$), que $P(X|\lambda) = 1/C$ es una constante (no depende de Q) y que $b_{q_t}(\mathbf{x}_t) = P(\mathbf{x}_t|q_t)$ es la probabilidad de observación de $\mathbf{x}_t$ en el estado q_t.

De este modo, si consideramos que los datos de entrada X presentan una cierta incertidumbre, entonces X y $P(Q|X, \lambda)$ se consideran variables aleatorias obteniéndose la siguiente regla de clasificación [166],

$$Q^* = \underset{Q}{\operatorname{argmax}}\, E[P(Q|X,\lambda)|\mathcal{K}] \tag{6.35}$$

donde $\mathcal{K}$ representa un cierto conocimiento *a priori* que podemos aplicar para llevar a cabo el cómputo del valor esperado.

De forma general, consideraremos que los datos observados X no son deterministas, sino que presentan una cierta incertidumbre considerada mediante una cierta *pdf* $f^{'}(X)$. En este caso el valor esperado a maximizar [167] en la ecuación (6.35) es,

$$E[P(Q|X,\lambda)|X \sim f^{'}(X)] = P(Q|\lambda)\int_X \frac{P(X|Q,\lambda)}{P(X|\lambda)} f^{'}(X)dX \tag{6.36}$$

A su vez, podemos considerar que $f^{'}(X)$ se deriva a partir de tres tipos de conocimiento: los datos de entrenamiento limpio X^{tr} (modelados por $P(X|X^{tr}) = P(X|\lambda)$), los datos observados no fiables Y y cualquier otro tipo de conocimiento $\mathcal{L}$. Si suponemos independencia entre X^{tr} e Y se puede obtener la siguiente expresión,

$$f^{'}(X) = P(X|X^{tr},Y,\mathcal{L}) = P(X|\lambda)P(X|Y,\mathcal{L})/P(X) \tag{6.37}$$

Si además suponemos que $P(X)$ es constante (igual a $1/C$), esta última ecuación se puede reescribir como,

$$f^{'}(X) = CP(X|\lambda)f(X) \tag{6.38}$$

donde $f(X) = P(X|Y,\mathcal{L})$ se conoce como la distribución *a posteriori* de X. Siempre que C y $P(X|\lambda)$ sean distintos de cero podemos escribir,

$$E[P(Q|X,\lambda)] = CP(Q|\lambda)\int_X P(X|Q,\lambda)f(X)dX \tag{6.39}$$

Bajo el supuesto de independencia temporal entre las probabilidades *a posteriori* de las observaciones,

$$f(X) = \prod_{t=1}^{T} f(\mathbf{x}_t) \tag{6.40}$$

finalmente obtenemos que el valor esperado a maximizar en (6.35) es,

$$E[P(X|Q,\lambda)] = C\prod_{t=1}^{T} a_{q_{t-1}q_t}\int_{\mathbf{x}_t} b_{q_t}(\mathbf{x}_t)f(\mathbf{x}_t)d\mathbf{x}_t \tag{6.41}$$

donde $f(\mathbf{x}_t)$ es la distribución *a posteriori* de $\mathbf{x}_t$. La expresión (6.41) es similar a (6.34), con la única diferencia de que la probabilidad de observación $b_{q_t}(\mathbf{x}_t) = p(\mathbf{x}_t|q_t)$ se reemplaza

por,

$$\int_{\mathbf{x}_t} p(\mathbf{x}_t|q_t) f(\mathbf{x}_t) d\mathbf{x}_t \tag{6.42}$$

Mediante esta modificación de las probabilidades de observación conseguimos que el reconocedor tenga en cuenta la probabilidad a posteriori $f(\mathbf{x}_t)$ asociada a las observaciones.

6.5.2. Aproximación Gausiana

La estimación MMSE que se presentó en la sección 6.4 nos permite aproximar la probabilidad *a posteriori* $f(\mathbf{x}_t)$ por la distribución $\gamma_t(i) = P(\mathbf{x}_t = \mathbf{x}^{(i)}|Y)$ (véase ecuación (6.18)), en la que se considera la evolución temporal de la voz (mediante el modelado HMM de la voz) y las distorsiones producidas por el error de propagación (mediante las probabilidades de observación variantes).

No obstante, aunque se disponga de la distribución *a posteriori* $f(\mathbf{x}_t)$, es necesario llevar a cabo una integración numérica para el cómputo de las probabilidades de observación modificadas, lo que puede traducirse en un incremento del coste computacional más allá de los límites del interés práctico. Afortunadamente, la integral establecida en la ecuación (6.42) se puede resolver de forma analítica bajo ciertas suposiciones:

1. La probabilidad de observación para un cierto estado s viene dada por una mezcla de gausianas,

$$p(\mathbf{x}_t|s) = \sum_{m=1}^{M} c_{s,m} \mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_{s,m}, \Sigma_{s,m}) \tag{6.43}$$

   donde $c_{s,m}$ es el peso de cada *pdf* de la mezcla, $\boldsymbol{\mu}_{s,m}$ es el vector media y $\Sigma_{s,m}$ es la matriz de covarianza correspondientes a la m-ésima gausiana multivariada del estado s.

2. La probabilidad *a posteriori* del vector no corrupto se puede aproximar mediante una densidad de probabilidad gausiana,

$$f(\mathbf{x}_t) \approx \mathcal{N}(\mathbf{x}_t; \hat{\mathbf{x}}_t, \hat{\Sigma}_t) \tag{6.44}$$

   donde $\hat{\mathbf{x}}_t$ es el vector media y $\hat{\Sigma}_t$ es la matriz de covarianza de la distribución a posteriori f.

En este caso, para cualquier estado s obtenemos la siguiente probabilidad de observación modificada,

$$\begin{aligned}\int_{\mathbf{x}_t} p(\mathbf{x}_t|s) f(\mathbf{x}_t) d\mathbf{x}_t &= \sum_{m=1}^{M} c_{s,m} \int_{\mathbf{x}_t} \mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_{s,m}, \Sigma_{s,m}) \mathcal{N}(\mathbf{x}_t; \hat{\mathbf{x}}_t, \hat{\Sigma}_t) d\mathbf{x}_t \\ &= \sum_{m=1}^{M} c_{s,m} \mathcal{N}(\hat{\mathbf{x}}_t; \boldsymbol{\mu}_{s,m}, \Sigma_{s,m} + \hat{\Sigma}_t)\end{aligned}$$

Además si suponemos que las gausianas multivariadas utilizadas en las ecuaciones (6.43) y (6.44) presentan matrices de covarianza diagonales, las distribuciones se pueden factorizar sobre los elementos de los vectores de características obteniendo la siguiente solución para cada dimensión del vector de características [168],

$$\sum_{m=1}^{M} c_{s,m} \prod_{k=1}^{K} \mathcal{N}(\hat{x}_t(k); \mu_{s,m}(k), \sigma^2_{s,m}(k) + \sigma^2_{\hat{x}_t}(k)) \tag{6.45}$$

donde $\hat{x}_t(k)$ y $\mu_{s,m}(k)$ son las componentes k-ésimas de los vectores $\hat{\mathbf{x}}_t$ y $\boldsymbol{\mu}_{s,m}$, respectivamente, mientras que $\sigma^2_{\hat{x}_t}(k)$ y $\sigma^2_{s,m}(k)$ hacen referencia a las correspondientes varianzas. En nuestro caso, cuando la distribución *a posteriori* se deriva de la distribución $\gamma_t(i) = P(\mathbf{x}_t = \mathbf{x}^{(i)}|Y)$, el valor medio $\hat{x}_t(k)$ se corresponde directamente con la componente k de la estimación MMSE, mientras que la varianza se puede determinar como,

$$\sigma^2_{\hat{x}_t}(k) = E[(x_t(k) - \hat{x}_t(k))^2 |Y]$$

De este modo, la incertidumbre de la estimación MMSE (establecida por la varianza $\sigma^2_{\hat{x}_t}(k)$) incrementa la varianza de las gausianas de la mezcla, restando peso a la contribución de estas componentes en la evaluación final de probabilidad llevada a cabo durante la fase de reconocimiento.

6.5.3. Incertidumbre de las Componentes Dinámicas

Tal y como vimos en el apartado anterior, la solución *soft-data* propuesta es directamente aplicable a las características estáticas. Sin embargo, puesto que la reconstrucción MMSE no es realizada para las componentes dinámicas, no se dispone de una estimación de las distribuciones condicionales de probabilidad para estas componentes. En el apartado 6.4.5 ya vimos que las medias de las componentes dinámicas pueden deducirse de sus

correspondientes estáticas. Las varianzas de las distribuciones de las componentes dinámicas deberán hallarse también a partir de las correspondientes distribuciones estáticas. Por tanto, si la característica delta o de velocidad $\Delta x_t(k)$ de la componente k-ésima se obtiene como la siguiente suma pesada,

$$\Delta x_t(k) = \sum_{l=-W^\Delta}^{W^\Delta} \omega_l x_{t+l}(k) \tag{6.46}$$

entonces, podemos computar su correspondiente varianza como,

$$\begin{aligned} \sigma^2_{\Delta x_t}(k) &= \mathrm{VAR}\left[\sum_{l=-W^\Delta}^{W^\Delta} \omega_l x_{t+l}(k)\right] \\ &= \sum_{l=-W^\Delta}^{W^\Delta}\sum_{k=-W^\Delta}^{W^\Delta} \omega_l\omega_k \mathrm{COV}\left[x_{t+l}(k), x_{t+k}(k)\right] \end{aligned} \tag{6.47}$$

Esta última expresión implica que el cómputo de la varianza de las componentes dinámicas exige el precómputo de las covarianzas cruzadas contenidas en la expresión anterior. Sin embargo, la complejidad de esta expresión se puede reducir notablemente si se supone que las variables aleatorias x_{t+l} $(-W^\Delta \leq l \leq W^\Delta)$ son independientes, obteniéndose entonces la siguiente aproximación,

$$\sigma^2_{\Delta x_t}(k) \approx \sum_{l=-W^\Delta}^{W^\Delta} \omega_l^2 \mathrm{VAR}\left[x_{t+l}(k)\right] = \sum_{l=-W^\Delta}^{W^\Delta} \omega_l^2 \sigma^2_{x_{t+l}}(k) \tag{6.48}$$

que sólo requiere el cómputo de las varianzas de las componentes estáticas. Del mismo modo, puede derivarse una expresión similar a (6.48) para las componentes delta-delta o de aceleración.

Aunque la suposición de independencia estadística es obviamente falsa, Peinado *et al.* demostraron en su trabajo [169] que esta expresión es útil y que introduce mejoras significativas respecto a no introducir las varianzas de las componentes dinámicas.

6.5.4. Resultados Experimentales

A lo largo de esta sección hemos descrito como la reconstrucción MMSE puede ser complementada mediante la consideración de la incertidumbre de las estimas en el proceso

de reconocimiento. Al igual que vimos durante la descripción de la reconstrucción MMSE, el éxito de este tipo de técnicas estará condicionado por el modelo de distorsiones considerado.

Así, Ion y Haeb-Umbach, en su propuesta [163], utilizan la estimación MMSE conjuntamente con la técnica *soft-data* en sistemas DSR, de modo que las distorsiones consideradas vienen dadas únicamente por la pérdida de un conjunto de observaciones. Estos autores extienden este mismo esquema, representado en la figura 6.2, a una arquitectura NSR. En este caso, las técnicas *soft-data* consiguen mejorar los resultados obtenidos por la reconstrucción MMSE (tablas 6.7 y 6.8).

Por otro lado, tal y como vimos en el apartado 6.4.6, el hecho de considerar un modelo de degradaciones y una reconstrucción adaptada a la arquitectura NSR nos permitió llevar a cabo una estimación MMSE más precisa que la propuesta por Ion y Haeb-Umbach. En este caso, las tablas 6.15 y 6.16 corresponden a los resultados *soft-data* obtenidos al aplicar reconstrucción MMSE considerando las distorsiones producidas por el error de propagación. En particular, estos resultados se determinaron utilizando el mismo conjunto de prototipos y probabilidades de observación descritas en el apartado 6.4.6.

Como acabamos de ver, las técnicas soft-data permiten incrementar el rendimiento de la reconstrucción MMSE mediante la consideración de la incertidumbre (varianza) de las estimas. En particular, el rendimiento de la técnica soft-data se hace más notable cuando las estimas MMSE son poco precisas. Así, las mejoras sobre la mera estimación MMSE se hacen más significativas cuando las pérdidas se producen en ráfagas largas (condiciones de canal con longitud media de ráfaga de tres y cuatro paquetes). En este caso, las estimas son poco fiables para los valores centrales de la ráfaga, lo que se traduce en valores de varianza altos para esas reconstrucciones. Así, al sumar estas varianzas a las de las probabilidades de observación de los estados del reconocedor, las probabilidades de éstos tienden a igualarse. De este modo, en esos puntos la observación errónea provoca una probabilidad de observación más uniforme entre estados, teniendo como efecto que el reconocimiento final venga dado por otras observaciones más fiables donde si se pueda discernir claramente entre estados.

6.6. Técnicas *Weighted Viterbi*

Una alternativa para tener en cuenta los valores de confianza o certeza de las observaciones es el reconocimiento de voz mediante el algoritmo *Weighted Viterbi* (WVA, *Weighted Viterbi Algorithm*). Tanto las técnicas *soft-data* como *weighted Viterbi* se engloban dentro

Tasa de Pérdidas	*Long. media ráfaga*			
	1	*2*	*3*	*4*
5 %	98.14	97.70	97.13	96.73
10 %	97.36	96.55	95.78	94.63
15 %	96.08	95.18	93.75	92.27
20 %	94.27	93.52	91.92	89.91

Tabla 6.13: Resultados de precisión de reconocimiento (WAcc) para AMR 12.2 kbps al aplicar la reconstrucción basada en estimación MMSE sin considerar la propagación de error e incluyendo la incertidumbre de las estimas mediante la técnica *soft-data*.

Tasa de Pérdidas	*Long. media ráfaga*			
	1	*2*	*3*	*4*
5 %	98.31	97.76	97.15	96.58
10 %	97.67	96.67	95.59	94.34
15 %	96.73	94.95	93.07	93.03
20 %	95.46	93.27	91.07	89.60

Tabla 6.14: Resultados de precisión de reconocimiento (WAcc) para G.729 8 kbps al aplicar la reconstrucción basada en estimación MMSE sin considerar la propagación de error e incluyendo la incertidumbre de las estimas mediante la técnica *soft-data*.

Tasa de Pérdidas	*Long. media ráfaga*			
	1	*2*	*3*	*4*
5 %	98.25	97.93	97.34	97.02
10 %	97.75	97.02	96.27	95.16
15 %	96.77	95.95	94.64	93.18
20 %	94.96	94.49	92.87	91.08

Tabla 6.15: Resultados de precisión de reconocimiento (WAcc) para AMR 12.2 kbps al aplicar la reconstrucción aditiva basada en estimación MMSE considerando la propagación de error e incluyendo la incertidumbre de las estimas mediante la técnica *soft-data*.

Tasa de Pérdidas	*Long. media ráfaga*			
	1	*2*	*3*	*4*
5 %	98.31	98.17	97.61	97.02
10 %	97.81	97.10	96.36	95.19
15 %	97.27	96.07	94.52	92.93
20 %	96.38	94.45	92.73	90.75

Tabla 6.16: Resultados de precisión de reconocimiento (WAcc) para G.729 8 kbps al aplicar la reconstrucción aditiva basada en estimación MMSE considerando la propagación de error e incluyendo la incertidumbre de las estimas mediante la técnica *soft-data*.

del paradigma del reconocimiento con datos no fiables e incompletos (MTD, *Missing Data Techniques*), recibiendo el nombre de técnicas de mitigación basadas en el reconocedor. La diferencia entre ambas técnicas reside en la aproximación que hacen al problema. Así, mientras que las técnicas soft-data obtienen una posible solución mediante un planteamiento puramente probabilístico, las técnicas WVA introducen un peso específico en el reconocimiento, calculado de forma heurística, para cada característica observada.

A diferencia de las técnicas de interpolación o estimación que tienen como objetivo la reconstrucción de los vectores perdidos o corruptos, las técnicas WVA adaptan la tarea de reconocimiento a las condiciones provocadas por las pérdidas. En este sentido, el reconocedor pasa a formar parte del algoritmo de mitigación de pérdidas, cuyo objetivo será la decodificación del mensaje de texto original a partir, no de la voz original, sino de la reconstrucción realizada.

La principal ventaja de este tipo de técnicas radica en el uso implícito del marco estadístico dado por el propio reconocedor. El modelado de la voz realizado por el reconocedor es muy superior al presentado en la estimación MMSE puesto que tiene en cuenta unidades acústicas superiores al vector de características. Además, el hecho de que las técnicas WVA no tengan como objetivo la reconstrucción de las observaciones originales, no las contrapone con las técnicas de interpolación y estimación descritas en secciones anteriores. Sin embargo, quedan pendientes dos cuestiones a resolver para la aplicación de las técnicas WVA:

- La modificación pertinente sobre el algoritmo de decodificación para aplicar dichos valores de confianza.
- Un criterio de asignación de confianza o fiabilidad a cada vector observado.

En las secciones siguientes se realizará una revisión sobre estos problemas tratando las soluciones presentes en la literatura y adaptándolas al marco del reconocimiento remoto basado en sistemas NSR.

6.6.1. Modificaciones al Algoritmo de Viterbi

La solución más extendida para los sistemas de reconocimiento de habla continua está basada en modelos ocultos de Markov. Tal y como vimos en la sección 2.5.2, los problemas derivados del reconocimiento basado en HMMs suelen agruparse en tres clases: la evaluación del modelo, la búsqueda del camino óptimo y el entrenamiento del modelo. Concretamente, la resolución del segundo problema, es decir, desvelar el conjunto de estados

ocultos dentro del macromodelo dado un conjunto de observaciones, es la base para el reconocimiento de voz continua. Este problema tiene una solución computacionalmente eficiente gracias al algoritmo de Viterbi. Este algoritmo es una solución de programación dinámica que opera de forma similar al algoritmo adelante-atrás introducido en la sección 6.4.2. El algoritmo de Viterbi se basa en el cálculo de la siguiente función [27],

$$\delta_t(i) = \max_{q_1,q_2,\ldots,q_{t-1}} P(q_1, q_2, \ldots, q_t = s_i, \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t|\lambda) \tag{6.49}$$

es decir, determinar la máxima probabilidad de que un único camino $q_1, q_2, \ldots, q_t$ en el instante t haya generado las observaciones $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t$ siendo 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 $b_j(\mathbf{x}_t)$ 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:

$$\delta_i(i) = \pi_i b_i(\mathbf{x}_1) \qquad 1 \leq i \leq N \tag{6.50}$$

$$\phi_1(i) = 0 \tag{6.51}$$

2. Recursión:

$$\delta_t(j) = \max_{1\leq i\leq N}[\delta_{t-1}(i)a_{ij}]b_j(\mathbf{x}_t) \qquad 2 \leq t \leq T \qquad 1 \leq j \leq N \tag{6.52}$$

$$\phi_t(j) = \operatorname*{argmax}_{1\leq i\leq N}[\delta_{t-1}(i)a_{ij}] \qquad 2 \leq t \leq T \qquad 1 \leq j \leq N \tag{6.53}$$

3. Finalización:

$$P^* = P[Q^*, X|\lambda] = \max_{1\leq i\leq N}[\delta_T(i)] \tag{6.54}$$

$$q_T^* = \operatorname*{argmax}_{1\leq i\leq N}[\delta_T(i)] \tag{6.55}$$

4. Recuperación del camino óptimo (secuencia de estados óptima $q_1^*, q_2^*, \ldots, q_T^*$):

$$q_t^* = \phi_{t+1}(q_{t+1}^*) \qquad t = T-1, T-2, \ldots, 1 \tag{6.56}$$

La idea central del algoritmo consiste en recorrer el diagrama de transiciones de estados a través del tiempo, almacenando para cada estado la probabilidad máxima acumulada y el estado anterior desde el que se llega con esta probabilidad. La complejidad 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.

En el siguiente apartado realizamos una breve revisión de las diversas modificaciones de este algoritmo que se han propuesto en la literatura para introducir valores de fiabilidad junto con las observaciones, de forma que sea posible expresar también la confianza en ellas.

6.6.2. Algoritmo *Weighted Viterbi*

El uso de este algoritmo fue propuesto inicialmente por Potamianos y Weerackody [170] como técnica de mitigación de errores para DSR sobre canales digitales. En este caso, el vector de características observado $\mathbf{x}$ está formado, debido a las degradaciones del canal, por un subconjunto de características confiables $\mathbf{x}_{rel}$ y por otro subconjunto de características desconfiables $\mathbf{x}_{unrel}$, de modo que $\mathbf{x} = [\mathbf{x}_{rel}, \mathbf{x}_{unrel}]$. Inicialmente, los autores proponen la marginalización de aquellas características no fiables, de modo que la probabilidad de observación para el estado s_i se puede obtener como,

$$P(\mathbf{x}_{rel}|s_i) = \int_{x_{unrel}} P(\mathbf{x}|s_i) d\mathbf{x}_{unrel} \tag{6.57}$$

Si modelamos la función de probabilidad como una mezcla de gausianas con matrices diagonales de covarianza, la probabilidad determinada en (6.57) puede ser fácilmente computada marginalizando aquellas componentes no fiables de la probabilidad total. La arquitectura DSR sobre canales inalámbricos digitales está sujeta a perturbaciones a nivel de bit. De este modo, es posible la existencia de vectores de características compuestos por componentes fiables y no fiables. En una red de paquetes esta situación no es posible,

sin embargo, este esquema es extensible introduciendo la siguiente modificación sobre la fase de recursión (previamente explicada) del algoritmo de Viterbi,

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}][b_j(\mathbf{x}_t)]^{\rho_t} \tag{6.58}$$

donde el peso ρ_t toma los valores 1 o 0, dependiendo de si el vector de características es confiable o no (en el caso de un sistema DSR sobre una red de paquetes, si se recibió o no). Nótese que cuando se asigna $\rho_t = 1$ la probabilidad de observación no se ve modificada, obteniéndose equivalencia entre las ecuaciones (6.52) y la ecuación (6.58). El caso contrario, cuando $\rho_t = 0$, hace que el factor dependiente de la probabilidad de observación adopte el valor de 1 independientemente de cual sea la observación. Esta medida hace que el proceso de reconocimiento no tenga ninguna dependencia de ese valor observado no confiable, guiándose únicamente por las probabilidades de transición. Por ejemplo, si se produce una pérdida en el instante de tiempo $t = l$, asignaríamos al vector de observación correspondiente un peso $\rho_l = 0$. Así, al computar la probabilidad en el reconocedor del camino $Q^* = (q_1, \ldots, q_{l-1}, q_l, q_{l+1}, \ldots q_T)$, expresada en la ecuación (6.54), para una frase de duración T vectores de características obtendríamos,

$$\begin{aligned} P^* = & a_{q_0 q_1} b_{q_1}(\mathbf{x}_1) a_{q_1 q_2} b_{q_2}(\mathbf{x}_2) \cdots \\ & b_{q_{l-1}}(\mathbf{x}_{l-1}) a_{q_{l-1} q_l} a_{q_l q_{l+1}} b_{l+1}(\mathbf{x}_{l+1}) \cdots \\ & a_{q_{T-1} q_T} b_{q_T}(\mathbf{x}_T) \end{aligned} \tag{6.59}$$

Como vemos, en este último desarrollo no interviene la probabilidad de observación $b_l(\mathbf{x}_l)$. Si tenemos en cuenta que las probabilidades de transición en el algoritmo de Viterbi son menos importantes que las probabilidades de observación, la ecuación (6.59) se comportaría como la técnica de borrado de tramas descrita en la sección 6.3.1. Esta aproximación puede ser interesante gracias a su simplicidad, pero sólo es correcta cuando el número de tramas perdidas de forma consecutiva es reducido. Cuando se producen ráfagas de pérdidas largas, esta técnica tiene un comportamiento mejor que el del borrado de tramas, ya que mantiene la estructura temporal de la frase original.

En oposición a las técnicas *soft-data*, presentadas en la sección 6.5, esta modificación del algoritmo de Viterbi tiene la ventaja de que no es necesario realizar ninguna suposición sobre la forma de la distribución de las probabilidades de observación, gracias a lo cual esta técnica puede aplicarse tanto a modelos HMM discretos como continuos. Además esta versión *hard* en la que sólo se discierne entre vectores confiables y no confiables,

puede ser fácilmente extensible a una versión *soft* en la que la confianza de los vectores de características adopta valores comprendidos en el intervalo $[0, 1]$.

Bernard y Alwan [147] establecen la siguiente aproximación,

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] \prod_{k=1}^{K} [b_j(x(k))]^{\rho_{t,k}} \tag{6.60}$$

donde los factores de confianza $\rho_{t,k}$ son configurables de forma independiente para cada una de las componentes $x(k)$ del vector de observación $\mathbf{x}_t$, siempre que la distribución de la probabilidad de observación venga dada por una única gausiana con matriz de covarianza diagonal.

En el caso de que tratemos con un modelo continuo en el que la probabilidad de observación de un cierto estado s venga dada por una mezcla de gausianas multivariadas, Potamianos y Weerackody [170] establecieron que es posible aplicar el algoritmo *weighted Viterbi* con pesos independientes a cada una de las componentes $x_t(k)$ $(k = 1, \ldots, K)$ del vector de observación. Es decir, podemos establecer un peso $\rho_{k,t}$ para cada componente k en cada instante de tiempo t modificando la probabilidad de observación a través de la siguiente expresión,

$$b_s(\mathbf{x}_t) = \sum_{m=1}^{M} c_{s,m} \prod_{k=1}^{K} \mathcal{N}(x_t(k); \mu_{s,m}(k), \sigma^2_{s,m}(k))^{\rho_{k,t}} \tag{6.61}$$

donde $\mathcal{N}(x(k); \mu_{s,m}(k), \sigma^2_{s,m}(k))$ representa una distribución gausiana para la k-ésima componente con media $\mu_{s,m}(k)$ y varianza $\sigma^2_{s,m}(k)$, y $c_{s,m}$ corresponde al peso de esa gausiana sobre la mezcla total de M gausianas.

Cardenal *et al.* [171] proponen una modificación en la que no es necesaria la suposición de que la mezcla de gausianas tenga una matriz covarianza diagonal, de modo que podemos expresar su alternativa como:

$$b_s(\mathbf{x}_t) = \sum_{m=1}^{M} c_{s,m}' \left(\frac{1}{\sqrt{(2\pi)^K |\Sigma_m|}} \right)^{\beta_t} \times \exp\left[-\frac{1}{2} (\mathbf{x}_t - \boldsymbol{\mu}_{s,m}) P_t (\Sigma_{s,m})^{-1} (\mathbf{x}_t - \boldsymbol{\mu}_{s,m})^{\mathrm{T}} \right] \tag{6.62}$$

donde,

$$P_t = \begin{pmatrix} \rho_{t,1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \rho_{t,K} \end{pmatrix} \tag{6.63}$$

es la matriz diagonal establecida por los pesos de cada componente. El coeficiente β_t actúa como una constante de normalización, mientras que los pesos de la mezcla $c_{s,m}$ son una versión modificada de los pesos originales $c_{s,m}$, tal y como muestran las siguientes expresiones,

$$\beta_t = \frac{\sum_{k=1}^{K} \rho_{t,k}}{K} \tag{6.64}$$

$$c'_{s,m} = c_{s,m} \left(\frac{1}{M}\right)^{(1-\beta_t)} \tag{6.65}$$

La idea que subyace bajo esta modificación es la de considerar la mezcla de gausianas con una covarianza modificada $\Sigma_{s,m} P_t^{-1}$. De este modo, el decremento del peso $\rho_{t,k}$ conlleva el incremento de la varianza de la componente k del vector de características. Así, cuando $\rho_{t,k}$ tiende a cero, la gausiana se transforma gradualmente en una distribución de probabilidad uniforme para esta dimensión, y la probabilidad final será la misma para cualquier observación y estado.

Puesto que en el desarrollo de este trabajo mantenemos la suposición de matrices de covarianza diagonales, como ya hicimos en la sección 6.5, trabajaremos con la solución obtenida en la ecuación (6.61) por Potamianos y Weerackody, ya que es una solución más sencilla que la propuesta por Cardenal y que nos permite la utilización de pesos independientes para cada una de las componentes del vector de características.

6.6.3. Cómputo de Pesos

Una vez introducidas las modificaciones necesarias para el pesado de los vectores de características observados en función de la confianza de las observaciones, el problema restante consiste en cómo determinar los factores de confianza dadas las observaciones.

En un sistema de reconocimiento remoto DSR, el terminal cliente envía directamente en cada paquete una versión cuantizada de los vectores de características extraídos de la voz. De este modo, la pérdida de paquetes implica la pérdida de las observaciones. Como vimos en secciones anteriores, la arquitectura NSR basada en codificadores CELP no sólo presenta este problema, sino que además se ve agravado por la propagación de error que causa el paradigma CELP. Así, si utilizamos la voz sintetizada por el decodificador, una vez concluida la ráfaga de pérdidas, los vectores de observación presentarán una cierta distorsión que hemos identificado bajo el término de *error de propagación*. Ateniéndonos

al esquema general de técnicas de mitigación de pérdidas para NSR presentado en la figura 6.1, consideraremos como vectores perdidos aquellos correspondientes a los paquetes perdidos y que vendrán marcados por el identificador $VLI = 1$. Las técnicas WVA presentadas en esta sección han sido ampliamente probadas en el contexto DSR y, bajo esta perspectiva, pueden ser directamente extendidas al esquema NSR sobre aquellos vectores marcados por el indicador $VLI = 1$. De este modo realizaremos una revisión de las múltiples estrategias que se encuentran en la literatura para DSR y terminaremos adaptando estas soluciones al problema de NSR en todas sus dimensiones.

Pesado por Vector de Características

Una primera aproximación al problema de escoger los valores de confianza para un conjunto de observaciones perdidas podría establecerse dando una confianza absoluta a aquellos vectores correspondientes a zonas de voz recibida y completamente no fiables a aquellos que se han perdido, es decir,

$$\rho_t = \begin{cases} 0 & VLI(t) = 1 \\ 1 & VLI(t) = 0 \end{cases} \tag{6.66}$$

Esta técnica se correspondería con la marginalización descrita en el ejemplo de la ecuación (6.59), donde la asignación binaria de pesos conlleva que el algoritmo de Viterbi progrese guiado por las probabilidades de transición y las probabilidades de observación correspondientes a los vectores de observación recibidos, ignorando las observaciones correspondientes a la pérdida. Esta técnica se corresponde con la aproximación *Missing Data* propuesta por Endo *et al.* en [172], que modifica la probabilidad de observación de aquellos vectores descartados sustituyéndola por un valor constante para todos los posibles estados del modelo oculto de Markov. A este respecto, Bernard y Alwan [147] demostraron que esta técnica sólo obtiene mejor rendimiento que la técnica de repetición en ráfagas de paquetes de larga duración.

Por tanto, los resultados conseguidos mediante la marginalización entre vectores completamente fiables o no fiables son susceptibles de ser mejorados mediante la integración de una técnica de interpolación o estimación. Así, los vectores de características descartados serían reemplazados por las sustituciones computadas en el bloque de reconstrucción. Puesto que la voz presenta altas correlaciones, estas sustituciones pueden resultar útiles para el reconocimiento. La adaptación de los métodos de reconstrucción con el algoritmo *weighted Viterbi* exige determinar de algún modo la confianza de esta sustitución, o lo

que es lo mismo, establecer cómo de útil es la reconstrucción empleada en el proceso de reconocimiento.

Una sencilla solución a este problema es la determinada por Cardenal *et al.* [171] en la que utilizan como técnica de reconstrucción la repetición NFR, basada en la repetición de los vectores previo ($t = 0$) y posterior ($t = T$) a una cierta ráfaga de pérdidas. Los valores de confianza se asignan aplicando una constante de confianza a aquellos vectores perdidos, es decir,

$$\rho_t = \begin{cases} C & VLI(t) = 1 \\ 1 & VLI(t) = 0 \end{cases} \tag{6.67}$$

Estos autores determinan la constante C de un modo empírico para distintos tipos de condiciones de canal, observando que el valor óptimo de C es más pequeño para aquellas condiciones de canal con longitud media de ráfaga mayor. Obviamente, cuanto mayor es la ráfaga, las reconstrucciones NFR utilizadas diferirán más de las observaciones originales, es decir, la sustitución empleada es menos confiable a medida que nos adentramos en la ráfaga de pérdidas. Por esta razón, una solución mejor que el pesado binario es establecer una función dependiente de la posición relativa dentro de la ráfaga para las reconstrucciones basadas en NFR. Cardenal *et al.* [173] establecen 2 aproximaciones. La primera de ellas atribuye los pesos de forma lineal,

$$\rho_t = \begin{cases} 1 - \alpha t & t = 1, \ldots, \lfloor T/2 \rfloor \\ 1 - \alpha(T - t) & t = \lfloor T/2 \rfloor + 1, \ldots, T - 1 \end{cases} \tag{6.68}$$

mientras que la segunda aproximación sigue una ley exponencial,

$$\rho_t = \begin{cases} \alpha^t & t = 1, \ldots, \lfloor T/2 \rfloor \\ \alpha^{(T-t)} & t = \lfloor T/2 \rfloor + 1, \ldots, T - 1 \end{cases} \tag{6.69}$$

donde $T - 1$ es el tamaño de la ráfaga de vectores reconstruidos y t es la posición relativa dentro de la ráfaga. El coeficiente α establece el factor de decaimiento de confianza de las repeticiones, obteniéndose los mejores resultados para $\alpha = 0{,}2$ en el caso lineal, y $\alpha = 0{,}7$ para el exponencial, resultados que son verificados por James y Milner [174].

Pesado por Componentes

Hasta ahora los métodos de asignación de pesos estudiados establecen un pesado único por vector de características perdido o descartado, sin distinguir entre componentes. Si tenemos en cuenta que las características dinámicas de velocidad son obtenidas a partir

de los valores estáticos considerados en una cierta ventana temporal de radio W^{Δ}, las confianzas sobre estos valores vendrán determinadas a partir de las confianzas estáticas. Esto da lugar a que, aunque las componentes estáticas de un cierto vector tengan una fiabilidad absoluta, las componentes dinámicas puedan presentar cierta incertidumbre originada por una pérdida anterior o posterior. El mismo razonamiento se puede aplicar a su vez para el cómputo de las características dinámicas de aceleración utilizando una ventana de radio $W^{\Delta\Delta}$. Este efecto fuerza a la utilización de un esquema en el que los pesos de confianza son asignados de forma independiente a cada característica haciendo uso del algoritmo *weighted Viterbi* propuesto en (6.61).

Puesto que las componentes dinámicas se obtienen de una forma determinista a partir de las estáticas, es razonable pensar que la confianza de éstas será una función de la confianza de las componentes estáticas. De este modo, las medidas de fiabilidad de las componentes de velocidad (ρ_t^{Δ}) y aceleración ($\rho_t^{\Delta\Delta}$) se computarán como una función de las medidas de confianza estáticas dentro de la ventana de cómputo correspondiente,

$$\rho_t^{\Delta} = f\left(\rho_{t-W^{\Delta}}, \rho_{t-W^{\Delta}+1}, \ldots, \rho_{t+W^{\Delta}-1}, \rho_{t+W^{\Delta}}\right) \tag{6.70}$$

$$\rho_t^{\Delta\Delta} = f\left(\rho_{t-W^{\Delta\Delta}}^{\Delta}, \rho_{t-W^{\Delta\Delta}+1}^{\Delta}, \ldots, \rho_{t+W^{\Delta\Delta}-1}^{\Delta}, \rho_{t+W^{\Delta\Delta}}^{\Delta}\right) \tag{6.71}$$

donde W^{Δ} y $W^{\Delta\Delta}$ son los radios de las ventanas de cómputo para las componentes de velocidad y aceleración, respectivamente.

En las técnicas *soft-data* estudiadas en la sección 6.5, la estructura matemática propuesta permite, bajo una serie de supuestos, establecer una medida de la confianza mediante la aproximación de la varianza de las componentes dinámicas. Sin embargo, en el caso del algoritmo *Weighted Viterbi* las confianzas de las componentes dinámicas son heurísticos ya que ésta es la naturaleza del algoritmo. A continuación presentamos algunos heurísticos ordenados de más a menos severos en la asignación de confianza:

- *Hard decoding*. Esta es la estrategia más restrictiva, puesto que establece un valor de desconfianza absoluto sobre una componente dinámica derivada de alguna característica estática no confiable, es decir,

$$\rho_t^{\Delta} = \begin{cases} 1 & \text{si } \{\rho_{t-W^{\Delta}}, \ldots, \rho_{t+W^{\Delta}}\} = 1 \\ 0 & \text{en cualquier otro caso} \end{cases} \tag{6.72}$$

  donde ρ_t^{Δ} es el valor de confianza de la componente dinámica en el instante t, y ρ_t las confianzas de las componentes estáticas de las que se deriva la dinámica.

Igualmente, se puede obtener una expresión similar para el cómputo de las confianzas $\rho_t^{\Delta\Delta}$ correspondientes a las componentes de aceleración.

- Producto de confianzas. La confianza de las derivadas temporales son obtenidas como el producto de las confianzas de los vectores estáticos en la ventana deslizante de cómputo,

$$\begin{aligned}\rho_t^{\Delta} &= \prod_{l=-W^{\Delta}}^{W^{\Delta}} \rho_{t+l} \\ \rho_t^{\Delta\Delta} &= \prod_{l=-W^{\Delta\Delta}}^{W^{\Delta\Delta}} \rho_{t+l}^{\Delta}\end{aligned} \tag{6.73}$$

- Aproximación binaria. Bernard y Alwan [147] proponen determinar si una componente dinámica es fiable en función de si la componente estática en el mismo instante de tiempo es totalmente fiable de modo que,

$$\rho_t^{\Delta} = \rho_t^{\Delta\Delta} = \begin{cases} 0 & \rho_t < 1 \\ 1 & \rho_t = 1 \end{cases} \tag{6.74}$$

- Medidas basadas en regresión. La confianza de las componentes dinámicas surge de expresiones como las propuestas por James y Milner [154],

$$\begin{aligned}\rho_t^{\Delta} &= \frac{\sum_{l=1}^{W^{\Delta}} l\rho_{t-l}\rho_{t+l}}{\sum_{l=1}^{W^{\Delta}} l} \\ \rho_t^{\Delta\Delta} &= \frac{\sum_{l=1}^{W^{\Delta\Delta}} l\rho_{t-l}^{\Delta}\rho_{t+l}^{\Delta}}{\sum_{l=1}^{W^{\Delta}} l}\end{aligned} \tag{6.75}$$

- Mínimo de confianzas. La confianza de una derivada temporal es asignada al mínimo de las confianzas de los valores estáticos de la ventana deslizante aplicada,

$$\begin{aligned}\rho_t^{\Delta} &= \min_{l=-W^{\Delta},\ldots,W^{\Delta}} \{\rho_{t+l}\} \\ \rho_t^{\Delta\Delta} &= \min_{l=-W^{\Delta\Delta},\ldots,W^{\Delta\Delta}} \{\rho_{t+l}^{\Delta}\}\end{aligned} \tag{6.76}$$

- Promedio de confianzas. La confianza de las componentes dinámicas es obtenida a través de una formulación regresiva inspirada en las ecuaciones utilizadas para el cómputo de las derivadas temporales. En el presente trabajo adoptamos este tipo de medidas de confianza, definidas como,

$$\rho_t^{\Delta} = \sum_{l=-W^{\Delta}}^{W^{\Delta}} |\omega_l^{\Delta}| \rho_{t+l}$$
$$\rho_t^{\Delta\Delta} = \sum_{l=-W^{\Delta\Delta}}^{W^{\Delta\Delta}} |\omega_l^{\Delta\Delta}| \rho_{t+l}^{\Delta} \tag{6.77}$$

  donde ω_l^{Δ} y $\omega_l^{\Delta\Delta}$ hacen referencia a los pesos utilizados para el cómputo de las componentes dinámicas. Considerando que la suma de los valores absolutos de los pesos es la unidad, entonces la expresión (6.76) resulta en un conjunto de pesos $0 \leq \rho_t^{\Delta}, \rho_t^{\Delta\Delta} \leq 1$.

En lugar de asignar pesos por grupos de características estáticas o dinámicas, el caso más general que podemos contemplar es la asignación de pesos individuales para cada una de las componentes del vector de características. Atendiendo a este esquema, Bernard y Alwan [147] establecen una medida de confianza $\rho_{k,t}$ para cada componente $x_t(k)$ reconstruida aplicando repetición hacia delante. Estos pesos se derivan a partir de la función de autocorrelación normalizada $\phi_k(t)$ de cada componente $x_t(k)$ $(1 \leq k \leq K)$. Tal medida es extendida por Cardenal y García [175] para el caso de NFR (repetición hacia delante y hacia atrás),

$$\rho_{k,t} = \begin{cases} \sqrt{\phi_k(t)} & t = 1, \ldots, \lfloor T/2 \rfloor \\ \sqrt{\phi_k(T-t)} & t = \lfloor T/2 \rfloor + 1, \ldots, T-1 \end{cases} \tag{6.78}$$

donde $T-1$ es el tamaño de la ráfaga de vectores reconstruidos y t es la posición relativa dentro de la ráfaga.

Cómputo de Pesos a partir de Estimación MMSE

Las técnicas para la asignación de pesos evaluadas hasta ahora se basan en la reconstrucción de los vectores de características perdidos en una arquitectura DSR. Concretamente, el esquema utilizado realiza la reconstrucción mediante la repetición de los vectores de características recibidos correctamente antes y después de una ráfaga (reconstrucción NFR).

En la arquitectura NSR la técnica de reconstrucción NFR ofrece unas mejoras limitadas frente a la estimación MMSE presentada en la sección 6.4, principalmente cuando se

consideran las degradaciones originadas por la propagación de error. Por este motivo, en el caso de NSR es más adecuado utilizar la estimación MMSE como técnica de reconstrucción. Sin embargo, es necesario proporcionar un método para la asignación de valores de confianza a las reconstrucciones realizadas.

En la aproximación *soft-data* se realizaron ciertas modificaciones sobre la regla de reconocimiento del reconocedor con el objetivo de incorporar la incertidumbre de las estimas en el proceso de reconocimiento. Bajo ciertos supuestos (véase la sección 6.5.2), las técnicas *soft-data* modifican la regla de reconocimiento introduciendo la media y el momento de segundo orden o varianza de la distribución $\gamma_t(i)$ obtenida durante la estimación MMSE. Concretamente, esta modificación conlleva aumentar las covarianzas de las probabilidades de observación de cada estado del reconocedor en un factor dependiente de la varianza de la estima.

Partiendo de la aproximación *soft-data*, parece acertado pensar que los valores de confianza de las reconstrucciones MMSE, necesarios para la aplicación del algoritmo *Weighted Viterbi*, tengan una relación directa con la varianza de las estimas. Así, Potamianos y Weerackody [170] establecieron la siguiente medida de confianza para un sistema DSR basado en transmisión inalámbrica,

$$\rho_{k,t} = 1 - \frac{E_{k,t}}{\sigma_k^2} \tag{6.79}$$

donde $\rho_{k,t}$ es la confianza de la característica k-ésima de un cierto vector de características, σ_k^2 es la varianza de esta componente y $E_{k,t}$ se corresponde con el error cuadrático medio entre la característica transmitida (u original) $x_t(k)$ y recibida (o corrupta) $y_t(k)$,

$$E_{k,t} = E\left[(y_t(k) - x_t(k))^2 | y_t(k)\right] = \sum_{i=0}^{N-1} (y_t(k) - x^{(i)}(k))^2 \cdot P(x^{(i)}(k)|y_t(k)) \tag{6.80}$$

donde $\{x^{(i)}(k); i = 0, \ldots, N-1\}$ es el conjunto de posibles valores cuantizados de la componente k-ésima que se pueden enviar. En el trabajo de Potamianos y Weerackody se considera un canal digital en el que las probabilidades $P(x^{(i)}(k)|y_t(k))$ son obtenidas a partir de los *soft-bit* que establece el decodificador. No obstante, esta expresión puede ser reutilizada en un entorno de pérdidas de paquetes para el cómputo de las confianzas de las estimas MMSE, reemplazando $P(x^{(i)}(k)|y_t(k))$ por la distribución $P(x^{(i)}(k)|Y)$ obtenida durante la estimación MMSE expresada por la ecuación (6.18).

Otra posible alternativa es utilizar una medida obtenida a partir de la información suministrada por la estimación MMSE. En esta estimación hemos modelado la voz como

un modelo HMM ergódico, donde cada estado corresponde a un vector prototipo $\mathbf{x}^{(i)}$, $i = 0, \ldots, N-1$. La confianza de la estima MMSE depende de la función de masa de probabilidad $\gamma_t(i) = P(\mathbf{x}_t = \mathbf{x}^{(i)}|Y)$. Si esta distribución presenta valores similares para todos los prototipos, entonces la confianza del vector reconstruido será baja. En el otro extremo, cuando esta distribución adopta un valor substancialmente alto para un cierto prototipo, entonces la estima corresponderá a un valor próximo al original. Atendiendo a este respecto, la entropía de la distribución puede ser considerada como una medida de la incertidumbre o desorden de una cierta distribución. Por tanto, podemos definir la *entropía instantánea* de una característica k reconstruida correspondiente al instante t como,

$$h_t^{(k)} = -\sum_{i=0}^{N-1} \gamma_t^{(k)}(i) \cdot \log_2 \gamma_t^{(k)}(i) \tag{6.81}$$

Es necesario hacer notar que para cada característica k es posible obtener una distribución $\gamma_t^{(k)}(i)$ $(i = 0, \ldots, N-1)$ atendiendo a la representación SVQ que realizamos en la sección 6.4.6 para la representación de los espacios vectoriales de características.

Partiendo de la entropía instantánea es posible definir un factor de confianza $\rho_t^{(k)}$ para cada componente k por medio de la siguiente expresión,

$$\rho_t^{(k)} = 1 - \frac{h_t^{(k)}}{\log_2 N} \tag{6.82}$$

Así, cuando la estimación MMSE no proporciona ninguna información para la reconstrucción de una cierta componente k, $\gamma_t^{(k)}$ presentará una distribución uniforme para todos los prototipos, de modo que la entropía de la distribución se hará máxima y adoptará el valor de $\log_2 N$. De esta forma, el factor de confianza obtenido por (6.82) es cero y el proceso de reconocimiento sólo estará guiado por las probabilidades de transición. En el caso contrario, cuando $\gamma_t^{(k)}(i) = 1$, para un cierto índice i, y $\gamma_t^{(k)}(j) = 0$ para $j \neq i$, la correspondiente entropía es nula y, por tanto, el factor de confianza se convierte en uno, de modo que la probabilidad de observación en el proceso de reconocimiento no se modifica.

6.6.4. Resultados Experimentales

Una de las aproximaciones más sencillas utilizada en la literatura consiste en utilizar como medio de reconstrucción la repetición NFR y el uso de un conjunto de pesos que decaiga desde los extremos utilizados en la reconstrucción. En las zonas próximas al principio y fin de la ráfaga de pérdidas, la repetición aproximará correctamente los vectores

de características perdidos, mientras que a medida que nos adentremos en la ráfaga la reconstrucción se alejará cada vez más de los valores originales. De este modo, tiene sentido utilizar un conjunto de pesos que establezcan cierta confianza en los valores próximos a los extremos y que decaigan a medida que nos distanciemos de éstos. Esa es la idea que subyace en la solución propuesta por Cardenal y que asigna los pesos atendiendo a la ecuación exponencial (6.69), donde el factor α óptimo fue determinado de forma empírica como $\alpha = 0{,}7$. Los resultados obtenidos aplicando esta técnica son los expuestos en las tablas 6.17 y 6.18 para los codificadores AMR y G.729, respectivamente. En ambos casos podemos observar cómo esta técnica consigue introducir notables mejoras, frente a la simple repetición NFR o la interpolación, cuando las ráfagas de paquetes perdidos son largas.

Aunque la repetición es una forma sencilla de obtener una reconstrucción de los vectores perdidos, vimos en apartados anteriores que en el caso de una arquitectura NSR sus resultados no son tan buenos como los conseguidos en DSR. El error de propagación degrada considerablemente el primer vector posterior a una ráfaga. Este hecho reduce las prestaciones de todas aquellas técnicas de mitigación que hacen uso del vector posterior a la ráfaga como si de un vector no corrupto se tratase. Consecuentemente, no tiene sentido aplicar un conjunto de pesos que decae simétricamente hacia el centro de una ráfaga de vectores descartados.

No obstante, nada nos impide hacer uso de la técnica de reconstrucción basada en estimas MMSE que tan buenos resultados mostró en secciones anteriores. Además, al igual que la técnica *soft-data*, la reconstrucción MMSE proporciona una serie de probabilidades que pueden ser utilizadas a la hora de establecer las confianzas de las reconstrucciones llevadas a cabo. Así pues, emulando la técnica *soft-data*, es posible obtener un conjunto de pesos a partir de la varianza de las estimas realizadas mediante la expresión (6.79) establecida por Potamianos y Weerackody [170], lográndose los resultados obtenidos en las tablas 6.19 y 6.20. En el desarrollo de estos experimentos se tuvieron en cuenta distintas medidas de confianza para los pesos correspondientes a las componentes dinámicas, aunque no se obtuvieron diferencias significativas en los resultados. Finalmente, se optó por utilizar un promedio de las confianzas atendiendo a la expresión (6.77). Esta opción no introduce degradaciones en las prestaciones de la reconstrucción MMSE cuando es utilizado en canales con ráfagas cortas, mientras que los métodos más restrictivos sí.

Como era de esperar, utilizar pesos que dependen de la varianza de las estimas MMSE consigue unos resultados similares a los obtenidos por la técnica *soft-data*, ya que en ambos casos las medidas de incertidumbre consideradas en el proceso de reconocimiento

Tasa de Pérdidas	*Long. media ráfaga*			
	1	*2*	*3*	*4*
5 %	97.94	97.48	96.95	96.76
10 %	96.83	95.99	95.24	94.49
15 %	94.99	93.98	93.18	92.37
20 %	91.81	91.91	91.11	89.88

Tabla 6.17: Resultados de precisión de reconocimiento (WAcc) para AMR 12.2 kbps al aplicar reconstrucción por repetición NFR y el algoritmo de reconocimiento *weighted Viterbi* con pesado exponencial variante en el tiempo ($\alpha = 0{,}7$).

Tasa de Pérdidas	*Long. media ráfaga*			
	1	*2*	*3*	*4*
5 %	97.95	97.04	96.73	96.46
10 %	96.74	95.25	94.79	93.93
15 %	95.16	93.06	92.50	91.40
20 %	93.18	90.29	89.91	88.78

Tabla 6.18: Resultados de precisión de reconocimiento (WAcc) para G.729 8 kbps al aplicar reconstrucción por repetición NFR y el algoritmo de reconocimiento *weighted Viterbi* con pesado exponencial variante en el tiempo ($\alpha = 0{,}7$).

Tasa de Pérdidas	*Long. media ráfaga*			
	1	*2*	*3*	*4*
5 %	98.32	97.91	97.42	97.05
10 %	97.73	97.08	96.27	95.11
15 %	96.54	95.87	94.64	93.31
20 %	94.63	94.44	92.92	91.24

Tabla 6.19: Resultados de precisión de reconocimiento (WAcc) para AMR 12.2 kbps al aplicar reconstrucción aditiva MMSE (considerando el error de propagación) y el algoritmo de reconocimiento *weighted Viterbi* con pesos obtenidos a partir de la varianza de la estima MMSE.

Tasa de Pérdidas	*Long. media ráfaga*			
	1	*2*	*3*	*4*
5 %	98.52	98.12	97.78	97.15
10 %	97.92	97.27	96.55	95.39
15 %	97.43	96.21	94.74	93.46
20 %	96.40	94.61	92.96	91.28

Tabla 6.20: Resultados de precisión de reconocimiento (WAcc) para G.729 8 kbps al aplicar reconstrucción aditiva MMSE (considerando el error de propagación) y el algoritmo de reconocimiento *weighted Viterbi* con pesos obtenidos a partir de la varianza de la estima MMSE.

Tasa de Pérdidas	*Long. media ráfaga*			
	1	*2*	*3*	*4*
5 %	98.35	98.06	97.64	97.34
10 %	97.75	97.35	96.76	95.77
15 %	96.79	96.26	95.58	94.24
20 %	95.34	95.33	94.09	92.65

Tabla 6.21: Resultados de precisión de reconocimiento (WAcc) para AMR 12.2 kbps al aplicar reconstrucción aditiva MMSE (considerando el error de propagación) y el algoritmo de reconocimiento *weighted Viterbi* con pesos obtenidos a partir de la entropía de las distribuciones utilizadas en la estima MMSE.

Tasa de Pérdidas	*Long. media ráfaga*			
	1	*2*	*3*	*4*
5 %	98.37	98.19	97.89	97.54
10 %	97.87	97.40	96.89	95.99
15 %	97.38	96.61	95.26	94.24
20 %	96.75	95.41	93.91	92.26

Tabla 6.22: Resultados de precisión de reconocimiento (WAcc) para G.729 8 kbps aplicando reconstrucción aditiva MMSE (considerando el error de propagación) y el algoritmo de reconocimiento *weighted Viterbi* con pesos obtenidos a partir de la entropía de las distribuciones utilizadas en la estima MMSE.

se derivan de la varianza de la distribución de probabilidad $\gamma_t(i) = P(\mathbf{x} = \mathbf{x}^{(i)}|Y)$. No obstante, es posible obtener resultados superiores si los pesos son extraídos a partir de la entropía de esta distribución, tal y como expresa la ecuación (6.82). Los resultados de esta alternativa se corresponden con los recogidos por las tablas 6.21 y 6.22 utilizando los codificadores AMR y G.729, respectivamente. Hay que hacer resaltar que esta última propuesta realizada obtiene el mejor comportamiento de las técnicas presentadas a lo largo de este capítulo.

6.7. Resumen de Resultados y Conclusiones

En este capítulo hemos llevado a cabo una revisión sobre las técnicas de mitigación basadas en el receptor, así como la propuesta de nuevas técnicas adaptadas a los sistemas de reconocimiento remoto NSR. Las figuras 6.7 y 6.8 muestran un resumen de los resultados de precisión de reconocimiento (WAcc) obtenidos a lo largo de este capítulo para los codificadores de voz AMR 12.2 kbps y G.729 8 kbps, respectivamente. Las condiciones de

Condiciones	C0	C1	C2	C3	C4
P_{loss}	0	5	10	15	20
L_{burst}	–	1	2	3	4

Tabla 6.23: Condiciones de canal para la comparativa de los resultados obtenidos por las técnicas de mitigación aplicadas.

canal se corresponden con la diagonal de las tablas de resultados mostradas a lo largo del presente capítulo, y que se resumen en la tabla 6.23. El resultado anotado como *baseline* corresponde con la tarea de reconocimiento llevado a cabo a partir de la voz decodificada incluyendo los algoritmos PLC definidos por los respectivos estándares de codificación. Además, se han incorporado los resultados obtenidos aplicando algunas técnicas básicas de mitigación, presentadas en la sección 6.3, como el esquema de repetición *NFR* o la *interpolación lineal*. Los algoritmos PLC integrados en los codificadores están basados en la repetición de parámetros y progresivo apagado o *muting*, lo que provoca una reducción considerable de la precisión del reconocimiento. Concretamente, el apagado llevado a cabo por G.729 es más rápido que el empleado en AMR, de modo que se producen más silencios artificiales para el códec G.729. Consecuentemente, estos silencios artificiales producen errores de inserción que se terminan traduciendo en un resultado *baseline* inferior. Las técnicas *NFR* e *interpolación lineal*, las cuales hacen uso de los vectores de características posterior y anterior a la ráfaga, mejoran notablemente el *baseline* ya que sustituyen el segmento afectado por el apagado (silencio artificial). Sin embargo, estas técnicas no tienen en cuenta que el vector posterior a la ráfaga se encuentra corrupto debido al error de propagación intrínseco a la arquitectura CELP de los codificadores. En este sentido, la técnica etiquetada como *FCDCN* combate la distorsión de los vectores de características posteriores a la ráfaga como si de un ruido acústico se tratara. Posteriormente, se lleva a cabo una interpolación lineal (que hace uso del vector reconstruido tras la pérdida) que obtiene una mejor reconstrucción de la pérdida y, consecuentemente, una mayor precisión de reconocimiento que la simple interpolación lineal.

Los resultados marcados como *FBMMSE* (notación seguida para dejar claro que la estimación MMSE hace uso de las recursiones *Forward-Backward*) en las figuras 6.7 y 6.8 logran mejorar los resultados FCDCN. De hecho, ambas técnicas se basan en la aplicación de un término corrector aditivo que depende del nivel de distorsión de los vectores de características a reconstruir, siendo la técnica *FBMMSE* propuesta ligeramente superior, ya que introduce correlaciones temporales de la voz a través del modelado HMM ergódico. Si además la reconstrucción MMSE se utiliza como fuente para la obtención

de medidas de confianza en el proceso de reconocimiento, las mejoras son substanciales. De este modo, los resultados marcados como *FBMMSE+soft-data* y *FBMMSE+WVA* corresponden a los dos métodos descritos en este capítulo para la consideración de la incertidumbre de las estimas MMSE. La técnica *FBMMSE+soft-data* corresponde con el incremento de las varianzas en las probabilidades de observación del reconocedor. Por otro lado, *FBMMSE+WVA* emplea el algoritmo *weighted Viterbi* con pesos obtenidos a partir de las entropías de las distribuciones utilizadas en las correcciones MMSE. Aunque el método *FBMMSE+WVA* tiene una base meramente heurística, los resultados conseguidos son superiores a los obtenidos mediante la aproximación *FBMMSE+soft-data*. La razón de esto puede encontrarse en la falta de validez de algunas de las suposiciones gausianas llevadas a cabo en la integración de la incertidumbre del método *soft-data*.

Aunque las mejoras conseguidas por los métodos propuestos basados en estimación MMSE son substanciales, la implementación de este tipo de reconstrucción presenta dos inconvenientes. El primero de ellos es la necesidad de almacenar la información correspondiente a las probabilidades de observación, lo que se traduce en el almacenamiento de $L_{burst} \times T_{EP}$ tablas, una por cada condición de distorsión considerada, para llevar a cabo la corrección del error de propagación. El segundo problema surge del empleo del algoritmo *Forward-Backward* que, en comparación con las técnicas más sencillas consideradas en este capítulo (NFR e interpolación lineal), incrementa la latencia en T_{EP} vectores de características. De cualquier modo, en el próximo capítulo veremos cómo aliviar estos problemas por medio de una extracción de características llevada a cabo directamente a partir de los parámetros del códec.

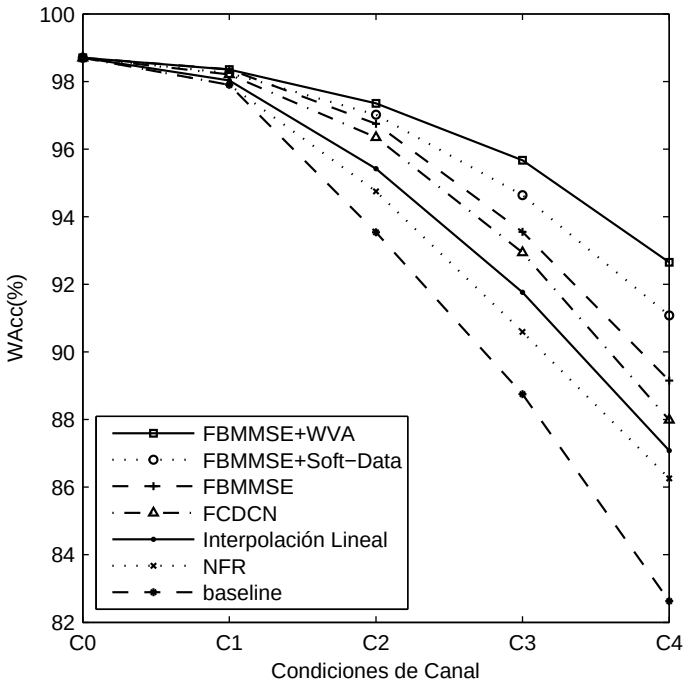


Figura 6.7: Resumen de resultados aplicando diversas técnicas de mitigación sobre voz decodificada AMR 12.2 kbps.

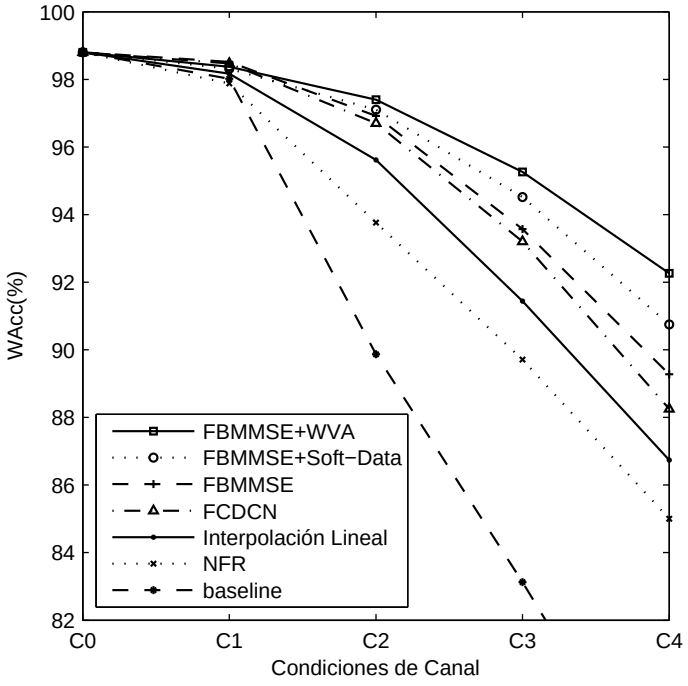


Figura 6.8: Resumen de resultados aplicando diversas técnicas de mitigación sobre voz decodificada G.729 8kbps.

Capítulo 7

Soluciones NSR basadas en Transparametrización

7.1. Introducción

En los capítulos hasta ahora desarrollados, la parametrización de reconocimiento se realizaba a partir de la voz decodificada. Sin embargo, la arquitectura NSR se puede modificar de forma que las características de reconocimiento se extraigan directamente de los parámetros del codificador. Esta variante, denominada transcodificación o transparametrización, evita la reconstrucción intermedia de la señal de voz, introduciendo un módulo de extracción de características a partir del flujo de bits recibido, es decir, los parámetros del codificador se transforman en características de reconocimiento. La nueva arquitectura de reconocimiento que surge suele recibir el nombre de B-NSR (*Bitstream-based* NSR) y, tal y como vimos en la sección 3.2.2, presenta ciertas ventajas sobre el reconocimiento a partir de voz decodificada.

El concepto de transparametrización no es nuevo y son muchas las propuestas que se han realizado desde mediados de la década de los 90. Inicialmente, los trabajos de Huerta *et al.* [176, 177] proponían la transparametrización como un modo de reducir la distorsión de codificación, ya que se evitan ciertos post-procesados perceptuales del decodificador. Gallardo *et al.* [178, 179] trabajaron en esquemas de transparametrización robustos frente a errores de canal, demostrando que esta alternativa es más robusta frente al reconocimiento de voz decodificada. Por otro lado, Kim *et al.* [146] desarrollaron un transparametrizador para el codificador IS-641, en el cual los autores incluyen técnicas de robustecimiento frente a ruido acústico. Finalmente, Peláez *et al.* [180, 181] plantean

diferentes técnicas de transparametrización para los codificadores FR, HR y G.723.1 que redundan en el robustecimiento frente a la distorsión de codificación y los errores de canal.

En los capítulos previos concluíamos que el carácter predictivo de la metodología empleada por los codificadores CELP se traduce en un comportamiento vulnerable frente a los errores del canal. En particular, veíamos que una pérdida no sólo origina la ausencia de la correspondiente información, sino que también provoca la aparición de un error posterior, el cual se traduce en una cierta distorsión de los parámetros de reconocimiento. En el capítulo anterior presentábamos un conjunto de técnicas especializadas que conseguían aliviar los efectos de la propagación de error sobre el reconocimiento a partir de voz decodificada. No obstante, 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 voz [122].

En este capítulo presentamos dos esquemas de transparametrización. El primero de ellos está orientado a los codificadores G.729 y AMR, y nos permite llevar a cabo una adaptación eficiente de las técnicas de mitigación de pérdidas presentadas en el capítulo anterior. La segunda parte del capítulo la dedicaremos al desarrollo de un transcodificador para iLBC, el cual nos permitirá aplicar técnicas de mitigación en el dominio de los parámetros del codificador.

7.2. Consideraciones Generales

A diferencia de otras propuestas encontradas en la literatura, los esquemas de transparametrización presentados en este trabajo extraen características de reconocimiento compatibles con el FE (*Front-End*) utilizado en el resto de capítulos. Esta consideración nos permite confrontar los resultados de todos los capítulos, puesto que las particularidades del proceso de extracción de características son similares en todos ellos.

Así pues, la parametrización considerada es la propuesta en la sección 4.6, es decir, la definida por el FE del estándar DSR básico [35]. En particular, la secuencia de vectores de características, extraída aplicando los correspondientes esquemas de transparametrización, se anotará como,

$$\boldsymbol{y}_t = [c_t(1),\, c_t(2), \ldots,\, c_t(12),\, c_t(0),\, \log E_t] \tag{7.1}$$

donde $c_t(k)$ $(k = 0, \ldots, 12)$ se corresponde con los 13 primeros coeficientes MFCC, mientras que $\log E_t$ hace referencia al parámetro de energía expresado de forma logarítmica.

El FE se aplica sobre tramas de 25 ms (200 muestras) cada 10 ms (80 muestras). Por otro lado, los codificadores utilizados a lo largo de este capítulo presentan tasas de trama diferentes (10 ms para G.729 y 20 ms para AMR 12.2 e iLBC), aunque todos ellos codifican la excitación en subtramas de la misma longitud ($N_{sf} = 40$ muestras). Así, de forma general, podemos decir que los vectores de características se computan sobre $L = 5$ subtramas cada $D = 2$ subtramas. La discrepancia entre las tasas de codificación y parametrización hace preciso aclarar la notación que utilizaremos. De este manera, cuando sea necesario distinguir entre parámetros obtenidos a diferentes tasas, emplearemos los siguiente índices: l para referirnos a una cierta trama de codificación; m para denotar una cierta subtrama; y t para indicar un cierto vector de características.

7.3. Transparametrización CELP

En particular, en esta sección proponemos un esquema de transparametrización para los codificadores G.729 y AMR 12.2 kbps, aunque podría ser extendido a otros codificadores CELP mediante ligeras modificaciones. Además, este esquema nos permitirá llevar a cabo una adaptación eficiente de las técnicas de mitigación de pérdidas que propusimos en el capítulo 6, reduciendo la cantidad de cómputo y la latencia introducida por los algoritmos.

7.3.1. Transparametrización de los Coeficientes LPC

Los codificadores CELP se basan en el modelo LPC de la señal de voz. En principio, tal y como vimos en la sección 2.3.4, sería posible obtener una representación cepstral LPCC directamente a partir de los coeficientes LPC. Sin embargo, para mantener un cierto grado de compatibilidad con la parametrización realizada por el FE, se realizará una etapa de análisis basada en banco de filtros sobre el espectro LPC.

Los codificadores en cuestión, AMR 12.2 kbps y G.729, responden a una estructura ACELP. Aunque la longitud de trama es distinta (10 ms para G.729 y 20 ms para AMR 12.2 kbps), la tasa de obtención de sus correspondientes parámetros es idéntica. En particular, estos codificadores extraen un conjunto de parámetros LPC (orden $p = 10$ en ambos casos) cada D subtramas (cada 10 ms), obteniéndose así una tasa equivalente a la de la parametrización del FE.

Los coeficientes MFCC, $c(k)$ $(k = 0, \ldots, 12)$, pueden obtenerse utilizando el procedimiento descrito en el FE, pero sustituyendo el espectro FFT por el módulo del espectro

LPC,

$$|H(\omega_i)| = \sigma|H'(\omega_i)| \qquad \omega_i = 2\pi/N_{FFT} \quad i = 0, \ldots, N_{FFT} - 1 \tag{7.2}$$

donde σ se corresponde con la ganancia LPC, N_{FFT} es el número de puntos sobre los que se realiza la FFT y $|H'(\omega_i)|$ es el espectro LPC normalizado, es decir,

$$|H'(\omega_i)| = \frac{1}{\left|1 - \sum_{k=1}^{p} a_k e^{-j\omega_i l}\right|} \qquad \omega_i = 2\pi/N_{FFT} \quad i = 0, \ldots, N_{FFT} - 1 \tag{7.3}$$

siendo a_k los coeficientes LPC y p el orden de predicción.

Si se aplica un banco de filtros triangulares con $F = 23$ filtros a $|H(\omega_i)|$, junto con una transformación DCT a las salidas logarítmicas del banco de filtros se deriva que,

$$c(k) = \begin{cases} F \log \sigma + c'(k) & k = 0 \\ c'(k) & k = 1, \ldots, 12 \end{cases} \tag{7.4}$$

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

7.3.2. Transparametrización de la Energía

Como vemos en la ecuación (7.4), para obtener el coeficiente $c(0)$ será necesario calcular la energía de la excitación σ^2. Además, la energía logarítmica también requiere de σ^2 ya que,

$$\log E = \frac{1}{2\pi} \int_{-\pi}^{\pi} |H(\omega)|^2 d\omega = \log \sigma^2 + \log \xi \tag{7.5}$$

donde ξ es la energía correspondiente del espectro LPC normalizado definida como,

$$\xi = \frac{1}{2\pi} \int_{-\pi}^{\pi} |H'(\omega)|^2 d\omega \tag{7.6}$$

Con el fin de evitar la síntesis de la excitación $e(n)$, dada por la expresión (5.1), podemos suponer que las entradas de los diccionarios adaptativo y fijo ($e_a(n)$ y $e_c(n)$, respectiva-

mente) están decorreladas, de modo que la energía de la excitación correspondiente a una cierta subtrama m puede ser expresada como,

$$\sigma^2(m) = g_a^2\sigma_a^2(m) + g_c^2\sigma_c^2(m) \tag{7.7}$$

donde $\sigma_a^2(m)$ y $\sigma_c^2(m)$ son las energías correspondientes a las contribuciones del diccionario adaptativo y fijo para la subtrama m. Las ganancias, g_a y g_c, y la señal de innovación se obtienen directamente a partir del flujo de bits codificados para cada subtrama. Finalmente, la energía σ_t^2 asociada al vector de características t se obtiene sumando la energía de las L subtramas correspondientes, ya que las características de reconocimiento del FE se obtienen a partir de tramas de 25 ms ($L \cdot N_{sf} = 200$ muestras).

El diccionario fijo o de innovación, que utilizan los codificadores aquí tratados, responde a una estructura algebraica, donde los códigos están formados principalmente por ceros y los valores distintos de cero se hallan permutando un conjunto de pulsos en una serie de posiciones prefijadas. De este modo, la energía de la contribución fija para una cierta subtrama m se puede obtener de forma general a partir de la siguiente expresión,

$$\sigma_c^2(m) = \sum_{n=0}^{N_{sf}-1} e_c^2(n) \tag{7.8}$$

En el caso del diccionario adaptativo, hemos de tener en cuenta que, tal y como explicábamos en el capítulo 5, éste está formado por las muestras de la excitación previa y, por tanto, no está contenido en el flujo de bits recibidos. Así, la contribución del diccionario adaptativo se determina mediante un filtro LTP con un cierto retardo T_a. No obstante, podemos aproximar la energía $\sigma_a^2(m)$ de una cierta subtrama m haciendo uso de la siguiente expresión,

$$\sigma_a^2(m) = \eta\sigma^2(m-P) + (1-\eta)\sigma^2(m-P-1) \tag{7.9}$$

donde,

$$\eta = \frac{T_a \bmod N_{sf}}{N_{sf}} \qquad P = \left\lfloor \frac{T_a}{N_{sf}} \right\rfloor$$

La expresión (7.9) indica que la energía de la contribución adaptativa $\sigma_a^2(m)$ para una cierta subtrama m se obtiene a partir de la energía de las subtramas $m-P$ y $m-P-1$, que contribuyen en proporciones η y $(1-\eta)$, respectivamente.

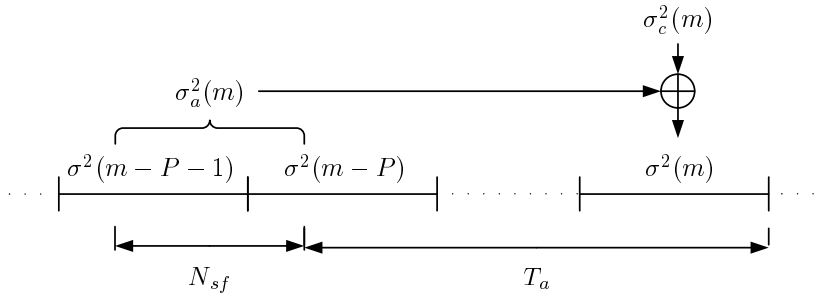


Figura 7.1: Cómputo de la energía de la excitación de una cierta subtrama a partir de los parámetros del codificador.

Teniendo en cuenta la expresión (7.9) podemos reescribir la ecuación (7.7) como,

$$\sigma^2(m) = \underbrace{g_a^2\left(\eta\sigma^2(m-P)+(1-\eta)\sigma^2(m-P-1)\right)}_{\text{Contribución Adaptativa}} + \underbrace{g_c^2\sigma_c^2(m)}_{\text{Contribución Fija}} \tag{7.10}$$

es decir, la energía de la excitación $\sigma^2(m)$ correspondiente a una cierta subtrama m se obtiene como la suma de una contribución fija, determinada a partir del vector de código, y una contribución adaptativa, dependiente de la energía de las subtramas previas. La figura 7.1 ilustra el modo de funcionamiento de la expresión (7.10).

Como acabamos de ver, la energía de la contribución adaptativa se determina llevando a cabo un promedio ponderado de las energías de excitación correspondientes a las subtramas previas que contienen el periodo de *pitch* dado por T_a. Puesto que los estándares de codificación AMR y G.729 establecen un valor mínimo para T_a de 17 y 20 muestras, respectivamente, es posible que $T_a < N_{sf}$. Esta situación supone un problema en la expresión (7.10) ya que en este caso tendríamos que $P = 0$ y, por tanto, la energía $\sigma^2(m)$ dependería de sí misma. En consecuencia, resolveremos este problema aproximando la energía de la subtrama actual, en el lado derecho de la ecuación (7.9), mediante la energía de la contribución fija, es decir, $\sigma^2(m) \simeq g_c^2\sigma_c^2(m)$.

7.3.3. Algoritmo de Mitigación de Pérdidas

El esquema utilizado en este caso es el mostrado en la figura 7.2. A diferencia del diagrama de la figura 6.1, este esquema extrae la secuencia de vectores de características $\boldsymbol{y}_t$ directamente del flujo de información recibida $\mathbf{c}_t$. El módulo *mapeador de pérdidas* encargado de

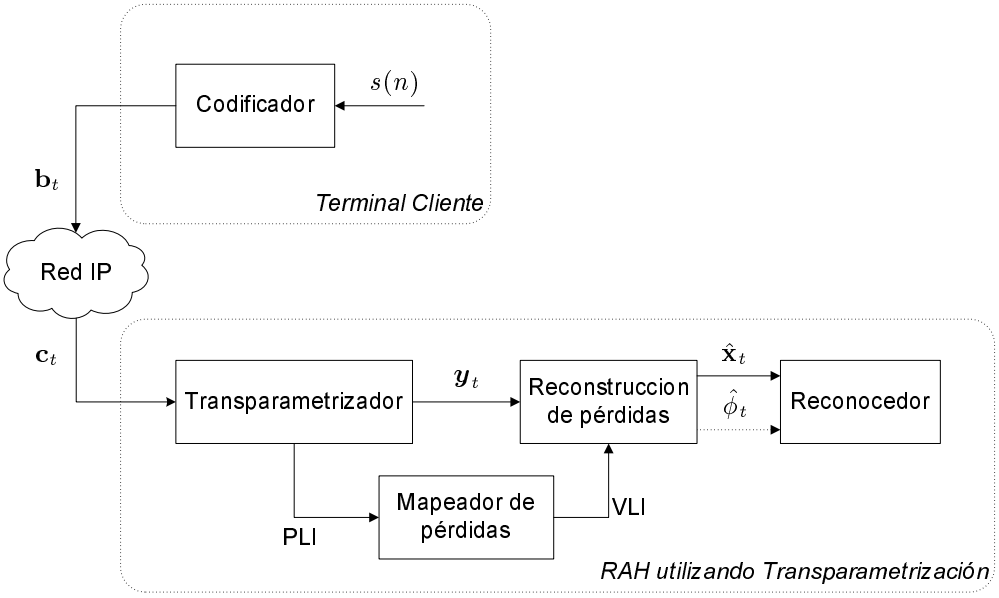


Figura 7.2: Esquema de mitigación de pérdidas propuesto para un sistema B-NSR.

llevar a cabo la correspondencia entre los indicadores PLI (*Packet Loss Indicator*) y VLI (*Vector Loss Indicator*) es el definido en la sección 6.3.4 (expresión (6.9)). Una vez realizada la correspondencia entre paquetes y vectores perdidos, el módulo de *reconstrucción de pérdidas* se encarga de obtener una estima $\hat{\mathbf{x}}_t$ de los vectores originales $\boldsymbol{x}_t$ (aquéllos que se obtendrían sin pérdidas de paquetes, es decir, cuando $\mathbf{c}_t = \mathbf{b}_t$), la cual vendrá acompañada de cierta información adicional $\hat{\boldsymbol{\phi}}_t$ que establece la fiabilidad de esta estima. La información $\hat{\boldsymbol{\phi}}_t$ podrá ser utilizada por el reconocedor para mejorar su rendimiento. A lo largo de este capítulo nos centraremos en adaptar las técnicas presentadas en el capítulo anterior al dominio B-NSR, reduciendo la complejidad y retardo de éstas.

Dentro de las técnicas descritas en el capítulo previo, las que mejores resultados presentaron fueron las basadas en la estimación MMSE. En particular, veíamos que esta estimación nos permitía considerar la propagación de error, propia de los codificadores CELP, mediante el uso de probabilidades de observación adecuadas. Para conocer los detalles precisos de este algoritmo, así como la notación utilizada, se remite a la sección 6.4. En la arquitectura B-NSR, no todas las componentes del vector de características transparametrizado se ven afectadas por las pérdidas de igual manera. Particularmente, podemos distinguir los siguientes casos:

- Las características $c_t(1) - c_t(12)$ se extraen directamente del espectro LPC, así que, en principio, no estarán afectadas por las distorsiones originadas por la propagación de error.

- Las componentes $\log E_t$ y $c_t(0)$ presentan una dependencia directa de la energía de la excitación, o lo que es lo mismo dependen de las subtramas previas (véase la ecuación (7.10)) y, por tanto, en caso de pérdidas se producirá una propagación de error sobre ellas.

En los siguientes apartados explicamos cómo aprovechar estas propiedades a la hora de llevar a cabo la reconstrucción MMSE. Además, propondremos un método recursivo para mitigar la propagación de error en las componentes $\log E_t$ y $c_t(0)$. Finalmente, determinamos cómo obtener las medidas de confianza de las reconstrucciones para incrementar el rendimiento del reconocedor, ya sea por medio de las técnicas *soft-data* o de las técnicas *weighted Viterbi* descritas en el capítulo anterior.

Reconstrucción MMSE

El diagrama de la figura 7.3 muestra cómo afecta una pérdida al esquema transparametrizador propuesto. En primer lugar, la pérdida supone un descarte de T_{VL} vectores debido a la pérdida de información (vectores marcados con $\text{VLI}(t) = 1$). Además, los estándares de codificación aplican ciertos procesados basados en el uso de predictores de media móvil (*moving average*, MA), normalmente variantes del algoritmo propuesto por Kataoka *et al.* [182], para cuantizar eficientemente los coeficientes LSF. Así, dependiendo del orden de este predictor, los primeros T_{MA} vectores tras una pérdida se encontrarán afectados por ésta (la figura 7.3 se corresponde con un predictor de segundo orden, de modo que $T_{MA} = 2$). Por contra, los parámetros de energía $c_t(0)$ y $\log E_t$, incluso cuando el espectro LPC es correcto, estarán distorsionados debido a su dependencia de la energía de la excitación, la cual a su vez depende de la energía de las tramas previas (véase la ecuación (7.7)).

Puesto que, como acabamos de ver, la propagación de error no afecta de igual manera a todas las características de reconocimiento, aplicaremos una reconstrucción MMSE diferenciada por pares de características. En particular, el cuantizador SVQ descrito en la sección 6.4.6 habilita este tipo de reconstrucción MMSE. Este cuantizador lleva a cabo el proceso de codificación de los vectores de características agrupándolos en 7 pares (uno de ellos viene dado por $c_t(0)$ y $\log E_t$). Así, la reconstrucción MMSE realmente viene dada por 7 estimaciones MMSE (una por cada par de características) que se realizan del siguiente modo:

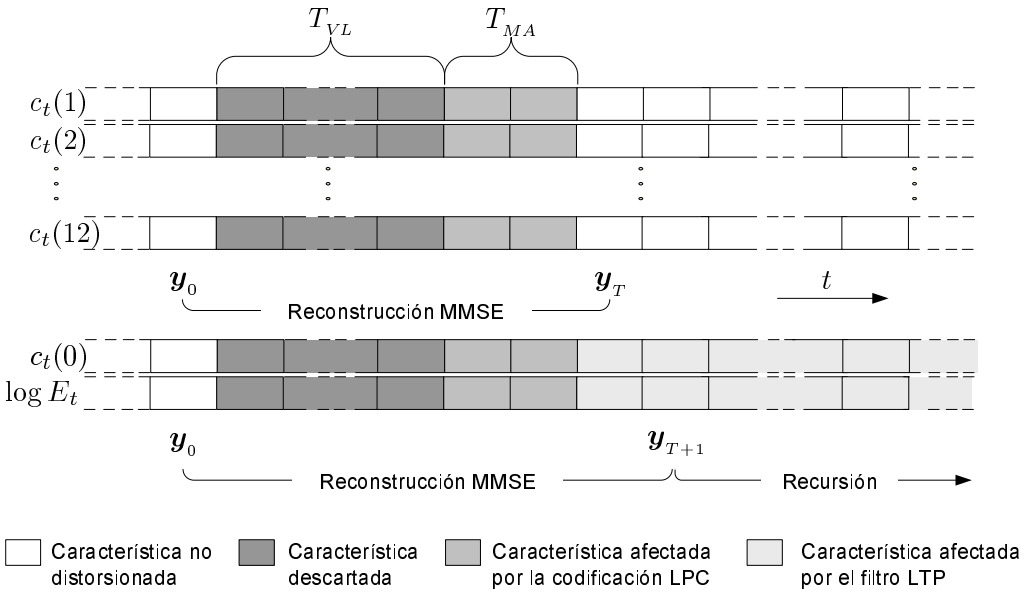


Figura 7.3: Reconstrucción de los vectores de características afectados por una pérdida en la arquitectura B-NSR.

- La reconstrucción MMSE de los 6 pares de componentes correspondientes a $c_t(1) - c_t(12)$ se lleva a cabo considerando que el error de propagación presenta una duración de T_{MA} vectores de características, en lugar de T_{EP} (longitud de propagación que considerábamos en el capítulo 6). Así, tal y como se muestra en la figura 7.3, aplicamos la estimación MMSE (*forward-backward*) desde $t = 0$ hasta $t = T$ (donde $T = T_{VL} + T_{MA} + 1$), utilizando $\mathbf{y}_0$ e $\mathbf{y}_T$ como puntos de inicialización.

- En el caso del par de características de energía (definido por $c_t(0)$ y $\log_t E$), la reconstrucción MMSE se realiza tomando como puntos de inicialización los instantes $t = 0$ y $t = T + 1$ (véase el diagrama de la figura 7.3), de modo que la última estima MMSE corresponde al instante $t = T$, aunque la degradación originada por σ^2 se propague para valores $t > T$. Consecuentemente, el retardo algorítmico de la reconstrucción MMSE descrita hasta ahora se reduce considerablemente ya que $T_{MA} + 1 \ll T_{EP}$.

La aplicación de las técnicas basadas en estimación MMSE, descrita con anterioridad, nos permite llevar a cabo la reconstrucción de los vectores de características perdidos y aquellos afectados por el predictor de media móvil. Sin embargo, las características $c_t(0)$ y $\log E_t$ presentan un error de propagación con una duración mayor debido a la existencia de un error remanente en el cómputo de la energía de la excitación para $t > T$. En

los apartados siguientes abordamos cómo tratar esta degradación de las componentes de energía.

Reconstrucción Recursiva de las Componentes de Energía

Puesto que de ahora en adelante sólo nos referiremos a las componentes energéticas $c_t(0)$ y $\log E_t$, emplearemos la notación vectorial para referirnos exclusivamente a este par de características. Concretamente, considerando las ecuaciones (7.4) y (7.5), podemos expresar, de forma general, un cierto par de características $\boldsymbol{p}_t$ como,

$$\boldsymbol{p}_t = [c_t(0),\, \log E_t] = [c'_t(0) + F \log \sigma_t,\, \log \xi_t + \log \sigma_t^2] \tag{7.11}$$

De esta forma, utilizaremos la notación $\hat{\mathbf{p}}_t$ para hacer referencia al par reconstruido en el instante t.

El último par de características reconstruido $\hat{\mathbf{p}}_t$ mediante la estimación MMSE corresponde al instante $t = T$, es decir, una vez que los coeficientes LPC han sido correctamente decodificados. Obsérvese en el diagrama de la figura 7.3 que las características $c_t(1) - c_t(12)$ en $t = T$, las cuales dependen exclusivamente de los coeficientes LPC, no se encuentran distorsionadas. De este modo, para $t \geq T$ las contribuciones del espectro LPC sobre un cierto par $\boldsymbol{p}_t$, dadas por $c'_t(0)$ y ξ_t en la expresión (7.11), se pueden calcular correctamente y, por tanto, podemos estimar la energía de la excitación correspondiente al par reconstruido $\hat{\mathbf{x}}_T$ como,

$$\hat{\sigma}_T^2 = \frac{\hat{E}_T}{\xi_T} \tag{7.12}$$

donde $\hat{E}_T$ se deriva directamente de la componente de energía logarítmica de $\hat{\mathbf{p}}_T$.

Nuestra propuesta consiste en utilizar la recursión definida por la ecuación (7.10) para extender hacia delante el último par $\hat{\mathbf{p}}_T$ reconstruido durante la fase de estimación MMSE (los valores iniciales requeridos para $\sigma^2(m-P)$ y $\sigma^2(m-P-1)$ se fijan a $\hat{\sigma}_T^2/L$), y de este modo obtener las reconstrucciones $\hat{\mathbf{p}}_t$ para $t > T$.

Medidas de Confianza de las Características Reconstruidas

Las técnicas *soft-data* y *weighted Viterbi* permiten incorporar medidas de confianza de los vectores de características observados en el proceso de reconocimiento, incrementando así su rendimiento. En particular, la reconstrucción MMSE se complementa idóneamente con estas técnicas, ya que a partir del cálculo de las probabilidades $\gamma_t(i)$ $(i = 0, \ldots, N-1)$,

definidas mediante la expresión (6.18), se derivan interesantes medidas de confianza de la reconstrucción alcanzada (para más detalles véanse las secciones 6.5 y 6.6).

No obstante, el método recursivo descrito con anterioridad para $t > T$ no permite el cómputo de las probabilidades $\gamma_t(i)$. En su lugar, con el objetivo de obtener estas medidas de confianza proponemos el empleo de las siguientes distribuciones de probabilidad,

$$\zeta_t(j) \equiv P(\hat{\mathbf{p}}_t = \mathbf{p}^{(j)} | \boldsymbol{\gamma}_T) \qquad \begin{array}{l} j = 0, \ldots, N-1 \\ t > T \end{array} \tag{7.13}$$

donde $\mathbf{p}^{(i)}$ $(i = 0, \ldots, N-1)$ se corresponde con los N pares prototipo utilizados durante la estimación MMSE, mientras que $\boldsymbol{\gamma}_T$ es un vector cuyas componentes son las probabilidades $\gamma_T(i)$ $(i = 0, \ldots, N-1)$ obtenidas durante la reconstrucción MMSE del instante T. Nótese que de esta forma estamos considerando que el error en los pares $\hat{\mathbf{p}}_t$ $(t > T)$ viene dado exclusivamente por el error cometido en $\hat{\mathbf{p}}_T$ propagado hacia delante. Esta interpretación es coherente, ya que si la última reconstrucción MMSE $\hat{\mathbf{p}}_T$ es perfecta, la reconstrucción recursiva descrita con anterioridad no introducirá ningún error.

Por tanto, resta exponer el método de cómputo de las probabilidades $\zeta_t(j)$ $(j = 0, \ldots N-1)$. Para ello, nuestra propuesta se basa en llevar a cabo un seguimiento de la evolución temporal de cada vector prototipo $\mathbf{p}^{(i)}$ utilizado en la estimación MMSE. Como veremos a continuación, la evolución temporal de cada prototipo supone una convergencia de éstos, de modo que sus correspondientes probabilidades $\zeta_t(j)$ se pueden fusionar (sumar) del mismo modo.

Ya que $\log \xi_t$ para $t \geq T$ se deriva correctamente del espectro LPC, éste valor es conocido en la expresión (7.11) y, por tanto, se puede obtener una energía de excitación $\hat{\sigma}_t^2$ asociada a cada par reconstruido $\hat{\mathbf{p}}_t$. Del mismo modo, se puede obtener una energía $\hat{\sigma}_T^2(i)$ para cada prototipo $\mathbf{p}^{(i)} = [c^{(i)}(0), \log E^{(i)}]$,

$$\hat{\sigma}_T^2(i) = \frac{E^{(i)}}{\xi_T} \qquad i = 1, \ldots, N-1 \tag{7.14}$$

y observar su evolución a partir de la siguiente expresión,

$$\hat{\sigma}_t^2(i) = G_t \cdot \hat{\sigma}_T^2(i) + \tilde{\sigma}_t^2 \qquad t > T \tag{7.15}$$

donde $\tilde{\sigma}_t^2$ $(t > T)$ es la energía de excitación obtenida tomando $\sigma_T^2(m-P) = \sigma_T^2(m-P-$

1) = 0 como condiciones iniciales en la recursión (7.10), mientras que G_t se obtiene como,

$$G_t = \frac{\hat{\sigma}_t^2 - \tilde{\sigma}_t^2}{\hat{\sigma}_{_T}^2} \qquad t > T \tag{7.16}$$

Hemos de puntualizar que, a medida que se incrementa t, G_t decrece ya que $\tilde{\sigma}_t^2$ tiende a $\hat{\sigma}_t^2$ y, por tanto, la expresión (7.15) cada vez depende menos de los valores $\hat{\sigma}_{_T}^2$. Finalmente, la anulación de G_t indicará que la reconstrucción de la energía $\hat{\sigma}_t^2$ viene dada exclusivamente por la contribución $\tilde{\sigma}_t^2$, desapareciendo la dependencia explícita de $\hat{\sigma}_{_T}^2$ (única fuente de incertidumbre). Paralelamente, a medida que G_t tiende a cero, las energías $\hat{\sigma}_t^2(i)$ se concentran en torno a $\tilde{\sigma}_t^2$. En otras palabras, esta evolución temporal puede interpretarse como si los diferentes vectores prototipo convergieran a medida que el tiempo transcurre, de modo que finalmente todos ellos convergen en un único prototipo.

Una vez obtenidas las energías $\hat{\sigma}_t^2(i)$ $(i = 1, \ldots, N-1)$ para un cierto instante de tiempo $t > T$, se pueden obtener sus correspondientes vectores de características transformados $\mathbf{z}_t^{(i)}$ a través de la siguiente expresión,

$$\mathbf{z}_t^{(i)} = [c_t^{\cdot}(0) + F \log \hat{\sigma}_t(i), \log \xi_t + \log \hat{\sigma}_t^2(i)]$$

ya que las componentes $c_t^{\cdot}(0)$ y $\log \xi_t$ son conocidas para $t \geq T$. Con el objetivo de agrupar aquellos vectores que se fusionan en el instante t, se definen los siguientes conjuntos,

$$Q_t(j) = \{i \in (0, 1, \ldots, N-1) | j = q(\mathbf{z}_t^{(i)})\} \tag{7.17}$$

donde la función $k = q(\mathbf{v})$ obtiene aquel índice k del prototipo $\mathbf{p}^{(k)}$ en el que se cuantiza el par de características $\mathbf{v}$. Estos conjuntos nos permiten calcular la evolución de la función de distribución de probabilidad $\zeta_t(j)$ para $t > T$ por medio de las siguiente expresión,

$$\zeta_t(j) = \begin{cases} \sum\limits_{i \in Q_t(j)} \gamma_{_T}(i) & Q_t(j) \neq \varnothing \\ 0 & Q_t(j) = \varnothing \end{cases} \qquad j = 0, \ldots, N-1$$

Así, la probabilidad condicional $\zeta_t(j)$ se computa sumando aquellas probabilidades $\gamma_{_T}(i)$ correspondientes al conjunto $Q_t(j)$, es decir, aquellos prototipos que convergen en $\mathbf{p}^{(j)}$ en el instante t. La evolución temporal originará una progresiva concentración de $\zeta_t(i)$ en torno al vector de características reconstruido en el instante t, disminuyendo la varianza y entropía asociada a esta distribución y, en consecuencia, la incertidumbre de la estimación

recursiva. Además, puesto que esta solución se basa en una recursión hacia delante, este método no incrementa la latencia de la estimación MMSE.

7.3.4. Resultados Experimentales

En primer lugar, es necesario conocer los resultados de reconocimiento conseguidos por la técnica de transcodificación propuesta sin introducir pérdidas de paquetes. Para llevar a cabo estas pruebas, así como el análisis de la robustez ante pérdidas de canal, se utilizó el marco experimental descrito en la sección 4.3. Los resultados de precisión de reconocimiento obtenidos fueron 98.79 % y 98.82 % para los codificadores AMR 12.2 y G.729, respectivamente. En este caso, llevamos a cabo la fase de entrenamiento utilizando como método de extracción de características el transcodificador. No obstante, puesto que el transcodificador extrae características compatibles a aquellas obtenidas por el FE, podemos utilizar este extractor para llevar a cabo la fase de entrenamiento. Así, tendríamos la ventaja adicional de disponer de un único entrenamiento tanto para NSR como para DSR. Los resultados obtenidos en este caso son de 98.79 % y 98.75 % para los transcodificadores de AMR 12.2 y G.729, respectivamente. Si comparamos estos resultados con los obtenidos a partir de voz decodificada (véase la tabla 4.2) observamos que la transcodificación obtiene, independientemente del tipo de entrenamiento utilizado, unos resultados ligeramente superiores. Una de las posibles razones que justifica este hecho es que la transparametrización evita el empleo de postprocesados de realce perceptual.

Con carácter general, para estudiar el rendimiento ante pérdidas de paquetes de las técnicas de mitigación descritas en este capítulo se han tenido en cuenta las mismas consideraciones expuestas en la sección 6.4.6. Sin embargo, en este caso se ha tomado $T_{EP} = T_{MA} + 1$ (siendo $T_{MA} = 2$ para AMR 12.2 kbps y $T_{MA} = 4$ para G.729) en lugar de $T_{EP} = 20$, lo que se traduce en un ahorro considerable en el almacenamiento de las tablas de observación y una reducción del retardo algorítmico empleado por el método de reconstrucción. Los vectores prototipo, necesarios para la aplicación de las técnicas MMSE, se corresponden con los centroides del diccionario SVQ utilizado por el estándar de codificación ETSI DSR [35]. El hecho de que la transparametrización propuesta tenga como objetivo la obtención de características compatibles con este estándar posibilita el empleo del citado conjunto de vectores prototipo. De cualquier modo, aunque se podría alcanzar una potencial mejora obteniendo un nuevo conjunto de prototipos adaptado a los vectores de características transcodificados, se optó por emplear el conjunto utilizado en el capítulo 6, de modo que la comparativa frente a aquellos resultados fuese justa. Aun

así, hay que hacer notar que, puesto que el error de propagación difiere en función del tipo de extracción de características utilizado (a partir de voz decodificada o mediante transparametrización), es preciso utilizar un conjunto de tablas de observación $b_i^{(t)}(\mathbf{y}_t)$ entrenadas específicamente para cada caso (véase sección 6.4.3).

Las tablas 7.1 y 7.3 muestran los resultados correspondientes tras aplicar el algoritmo de mitigación propuesto en esta sección mediante los métodos *soft-data* y *weighted Viterbi* (WVA), respectivamente, para el codificador AMR 12.2 kbps. Del mismo modo, las tablas 7.2 y 7.4 muestran los resultados obtenidos para el codificador G.729 8 kbps. En ambos casos, el empaquetado utilizado es el determinado en la tabla 4.1, de modo que los resultados son comparables a los mostrados en los capítulos previos. En particular, la comparación con las tablas 6.15 y 6.16 (para *soft-data*), y 6.21 y 6.22 (para *weighted Viterbi*), ilustran la superioridad de la transparametrización.

7.4. Transparametrización iLBC

En el capítulo 4 analizábamos el rendimiento de la arquitectura NSR básica, es decir, llevando a cabo el reconocimiento a partir de la voz sintetizada por códecs convencionales de voz. Bajo este esquema se evaluó la degradación introducida por la pérdida de paquetes en una red IP. Como resultado concluyente se observó que los codificadores CELP introducían degradaciones significativas que sus algoritmos PLC (*Packet Loss Concealment*) eran incapaces de mitigar. Por el contrario, el codificador iLBC conseguía ser más robusto ante la pérdida de paquetes. Sin embargo, las prestaciones de un esquema NSR sobre voz sintetizada por iLBC (resultados de la tabla 4.7) aún distan bastante de los resultados obtenidos por el esquema DSR (resultados de la tabla 4.8). Las razones que justifican estas diferencias son las siguientes:

- iLBC incluye un algoritmo de realce que contribuye a la mejora de la calidad perceptual mediante la introducción de ciertas distorsiones no lineales sobre el residuo [183].

- iLBC utiliza un algoritmo PLC orientado a las comunicaciones de voz. Éste presenta restricciones de retardo mayores que las de reconocimiento, así como otras consideraciones perceptuales. En particular, las medidas de mitigación que adopta están basadas en la atenuación progresiva de la voz cuando se pierden varios paquetes consecutivos (*muting*), la repetición de tramas sólo hacia delante (para no

Tasa de Pérdidas	*Long. media ráfaga*			
	1	*2*	*3*	*4*
5 %	98.58	98.29	97.71	97.37
10 %	98.24	97.47	96.70	95.76
15 %	97.71	96.63	95.33	93.93
20 %	96.86	95.61	93.93	92.29

Tabla 7.1: Resultados de precisión de reconocimiento (WAcc) con transparametrización sobre AMR 12.2 kbps al aplicar reconstrucción MMSE y *Soft-Data*.

Tasa de Pérdidas	*Long. media ráfaga*			
	1	*2*	*3*	*4*
5 %	98.55	98.18	97.61	97.09
10 %	98.26	97.50	96.46	95.24
15 %	97.79	96.58	94.94	93.16
20 %	97.35	95.32	93.48	91.32

Tabla 7.2: Resultados de precisión de reconocimiento (WAcc) con transparametrización sobre G.729 8 kbps al aplicar reconstrucción MMSE y *Soft-Data*.

Tasa de Pérdidas	*Long. media ráfaga*			
	1	*2*	*3*	*4*
5 %	98.59	98.29	97.90	97.61
10 %	98.23	97.69	97.06	96.31
15 %	97.65	96.94	95.96	94.75
20 %	97.08	96.07	94.81	93.61

Tabla 7.3: Resultados de precisión de reconocimiento (WAcc) con transparametrización sobre AMR 12.2 kbps al aplicar reconstrucción MMSE y WVA.

Tasa de Pérdidas	*Long. media ráfaga*			
	1	*2*	*3*	*4*
5 %	98.52	98.28	97.89	97.46
10 %	98.21	97.64	96.99	96.08
15 %	97.82	96.93	95.69	94.46
20 %	97.38	95.96	94.58	92.89

Tabla 7.4: Resultados de precisión de reconocimiento (WAcc) con transparametrización sobre G.729 8 kbps al aplicar reconstrucción MMSE y WVA.

incrementar el retardo) y el procesado realizado tras una pérdida de paquetes para evitar transiciones molestas (mezclado de tramas).

La solución adoptada en la sección 7.3 utiliza la transcodificación con el fin de reducir el efecto de las dependencias intertrama en los codificadores CELP. En el caso de iLBC, puesto que no existen dependencias, el objetivo perseguido es distinto. El transcodificador es utilizado para poder llevar a cabo algoritmos de mitigación más eficientes sobre los parámetros del códec, consiguiendo así operar en un dominio más adecuado para el reconocimiento y, por tanto, mayor robustez ante las pérdidas de paquetes.

Aunque iLBC presenta dos modos de funcionamiento [83], determinados por una longitud de trama de 20 ms (15.2 kbps) y 30 ms (13.33 kbps), el transcodificador desarrollado se basa exclusivamente en el modo de 20 ms. En particular, este modo opera sobre tramas de 160 muestras (obtenidas con una frecuencia de muestreo de 8 kHz) que son divididas en 4 subtramas de $N_{sf} = 40$ muestras. Cada trama contiene un conjunto de parámetros LSF obtenidos a partir del análisis LPC (con un orden $p = 10$) llevado a cabo sobre las 240 últimas muestras (se utilizan 80 muestras de la trama anterior).

7.4.1. Transparametrización de los Coeficientes LPC

En el caso del estándar FE, el bloque de extracción de características actúa sobre tramas de 25 ms (200 muestras) cada 10 ms (80 muestras), o lo que traducido a subtramas iLBC implica procesar $L = 5$ subtramas cada $D = 2$. La tasa a la que se obtienen los coeficientes LPC del codificador iLBC es cada 20 ms, es decir, un conjunto LPC por trama. Por tanto, existe una desadaptación entre la tasa de obtención de los vectores de características del FE y la de los coeficientes LPC. Concretamente, ésta última es la mitad que la del FE por lo que es preciso llevar a cabo una fase de interpolación de los coeficientes LPC. No obstante, el dominio LPC no es el más apropiado para realizar operaciones de interpolación por los problemas de estabilidad que implicaría [56]. Debido a esto, la interpolación se lleva a cabo en el dominio LSF en el que se encuentran codificados los parámetros LPC, ya que este dominio permite la interpolación sin el peligro de que aparezcan filtros inestables y, además, se introduce una menor distorsión [60].

La interpolación utilizada responde a la siguiente expresión [122],

$$\begin{aligned} \bar{\boldsymbol{w}}_{2l} &= \sum_{k=1}^{L} p_k \boldsymbol{w}_{l-k} \\ \bar{\boldsymbol{w}}_{2l+1} &= \sum_{k=1}^{L} q_k \boldsymbol{w}_{l-k} \end{aligned} \tag{7.18}$$

donde $\bar{\boldsymbol{w}}_{2l}$ y $\bar{\boldsymbol{w}}_{2l+1}$ hacen referencia a los conjuntos LSF asignados a las tramas del FE $t = 2l$ y $t = 2l+1$, $\boldsymbol{w}_l$ es el conjunto LSF de la trama iLBC l, L es el número de conjuntos LSF implicados en la interpolación y p_k y q_k son los coeficientes de interpolación. Nótese que, al hacer corresponder de esta forma los índices, se consigue duplicar el número de conjuntos LSF de salida respecto a los de entrada, adaptando de este modo las tasas de trama. Los pesos utilizados para la interpolación son $p_1 = q_3 = 6/20$, $p_2 = q_2 = 13/20$ y $p_3 = q_1 = 1/20$, de tal forma que $L = 3$.

Una vez llevada a cabo la interpolación de los conjuntos LPC, los coeficientes $c(k)$ $(k = 1, \ldots, 12)$ se podrían obtener a partir del espectro LPC aplicando el procedimiento definido en el apartado 7.4.1. Llegados a este punto, es necesario resaltar un detalle de implementación del codificador iLBC que marca notablemente el desarrollo del transcodificador: la expansión de ancho de banda. Esta operación consiste en introducir una modificación sobre los coeficientes LPC calculados en el codificador antes de ser transformados al dominio LSF y, por tanto, antes de su cuantización. La expansión de ancho de banda, que ya se definió mediante la expresión (5.4), se traduce en la siguiente modificación del filtro LPC,

$$H_{exp}(z) = H(\frac{z}{\gamma}) \tag{7.19}$$

donde γ es el factor de expansión (en este caso $\gamma = 0{,}9$). Desde el punto de vista de la transcodificación, la operación de expansión supone un serio inconveniente ya que introduce una fuerte distorsión en el espectro LPC. Atendiendo a la expresión (5.4), en principio cabría esperar que la operación inversa de expansión fuese fácilmente realizable. Sin embargo, el proceso de cuantización veta el uso de esta posibilidad en el decodificador, ya que al deshacer la operación pueden aparecer filtros inestables. Tal y como se observa en la figura 7.4, la expansión de ancho de banda implica que el espectro LPC no se corresponda con la envolvente espectral. Particularmente, vemos cómo la distorsión introducida se traduce en una reducción del margen dinámico del espectro y un desplazamiento de los formantes.

Para corregir esta situación es necesario extraer las características espectrales que todavía subyacen en el residuo codificado, de modo que éstas puedan ser utilizadas para refinar la respuesta en frecuencia determinada por los coeficientes LPC expandidos. Nuestra propuesta se basa en procesar el residuo codificado para obtener un nuevo conjunto de coeficientes LPC, $a_{res}(k)$ $(k = 1, \ldots, p_{res})$, que caracterizan la respuesta en frecuencia LPC del residuo $H_{res}(\omega)$. Así, mediante la siguiente expresión podemos obtener una estima mejorada $|\hat{H}(\omega)|$ del espectro LPC,

$$|\hat{H}(\omega)| = |H_{exp}(\omega)| \cdot |H_{res}(\omega)| \tag{7.20}$$

La figura 7.5 muestra el módulo del espectro de la señal original, el espectro LPC sin expandir $|H(\omega)|$ y, por último, la estimación mejorada $|\hat{H}(\omega)|$ referida, en este caso, como transparametrizador. El método propuesto ($p = 10$ coeficientes LPC expandidos y $p_{res} = 10$ coeficientes LPC calculados a partir de la excitación codificada) consigue mejorar la aproximación de la respuesta en frecuencia. Teniendo en cuenta la expresión (7.20), el espectro normalizado en ganancia de $|\hat{H}'(\omega_i)|$ puede expresarse del siguiente modo,

$$|\hat{H}'(\omega_i)| = \frac{1}{|1 - \sum_{k=1}^{p} a_{exp}(k)e^{-j\omega_i k}| \cdot |1 - \sum_{k=1}^{p_{res}} a_{res}(k)e^{-j\omega_i k}|} \tag{7.21}$$

Ya que esta nueva expresión sí obtiene una estima precisa de la envolvente espectral, podemos obtener los coeficientes $c(k)$ $(k = 1, \ldots, 12)$ llevando a cabo el proceso descrito en la sección 7.3.1 a partir de este espectro.

7.4.2. Transparametrización de la Energía

iLBC lleva a cabo una codificación sin dependencias intertrama de la excitación, lo que nos permite su fiel reconstrucción en el lado del decodificador permitiendo así el cómputo de su espectro LPC normalizado, $H'_{res}(\omega)$, y su energía, σ^2_{res} (este cómputo ha de ser realizado a la tasa correspondiente del FE, es decir, sobre tramas de 25 ms cada 10 ms). Teniendo en cuenta estos parámetros es posible determinar la ganancia $\hat{\sigma}$ correspondiente a $\hat{H}(\omega)$. Veamos cómo: en el lado decodificador la excitación es conocida, por lo que se puede computar su energía E_{res} y sus correspondientes coeficientes LPC $(a_{res}(k))$ a la tasa correspondiente del FE. Teniendo en cuenta estos parámetros es posible determinar

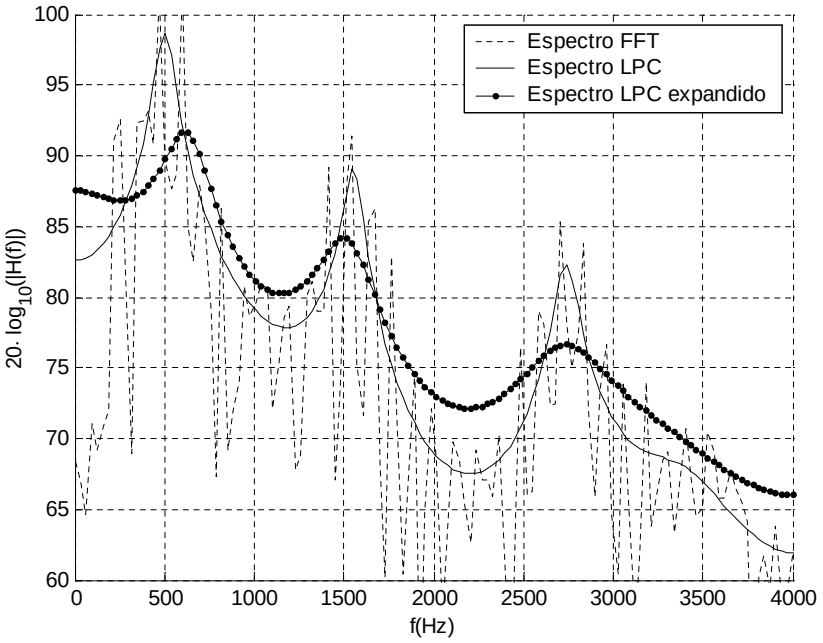


Figura 7.4: Distorsión producida por la operación de expansión de ancho de banda en el módulo de la respuesta en frecuencia LPC con $p = 10$.

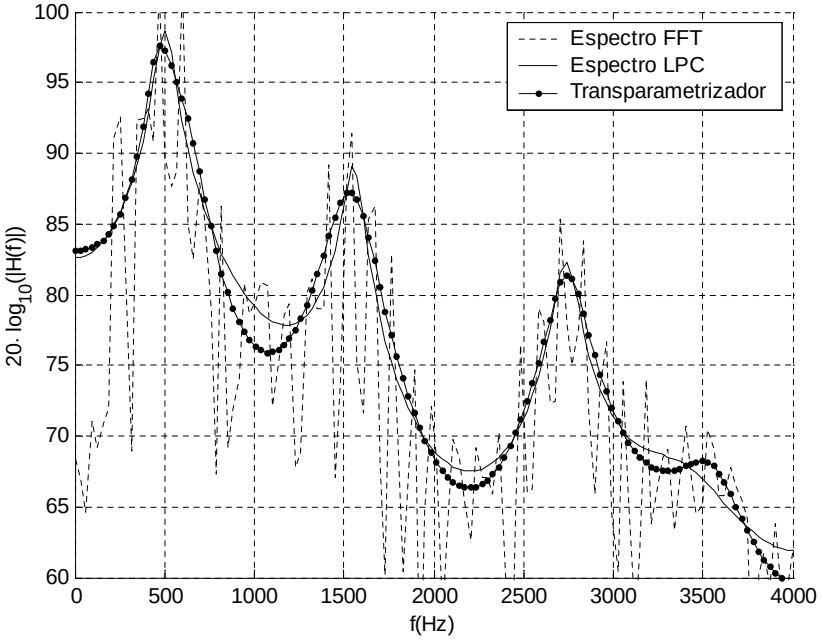


Figura 7.5: Aproximación del espectro LPC a través del esquema transparametrizador propuesto con $p_{res} = 10$.

Téc. de Mitigación	*Parámetros*		
	LSF_{exp}	LSF_{res}	σ^2_{res}
PLC_1	I	I	I
PLC_2	I	NFR	NFR
PLC_3	NFR	I	I

Tabla 7.5: Algoritmos de mitigación basados en operaciones de repetición del parámetro más cercano (NFR) e interpolación lineal (I).

la ganancia $\hat{\sigma}$, correspondiente al espectro $\hat{H}(\omega)$ mediante la siguiente expresión,

$$\hat{\sigma}^2 = \frac{E_{res}}{\frac{1}{2\pi}\int_{-\pi}^{\pi} |H_{res}^{'}(\omega)|^2 \, d\omega} \tag{7.22}$$

Finalmente, la característica $c(0)$ y $\log E$ se obtienen sustituyendo la ganancia $\hat{\sigma}$ y el espectro normalizado $\hat{H}^{'}(\omega)$ en la expresiones (7.4) y (7.6).

7.4.3. Algoritmo de Mitigación de Pérdidas

En este caso, el algoritmo de mitigación de pérdidas que proponemos trabaja sobre el dominio de los parámetros extraídos a partir del flujo de bits recibido por el decodificador, es decir, sobre el conjunto de coeficientes LPC expandidos, $a_{exp}(k)$ $(k = 1, \ldots, p)$, el conjunto de coeficientes LPC calculados a partir de la excitación, $a_{res}(k)$ $(k = 1, \ldots, p_{res})$, y la energía de la excitación σ^2_{res}. En particular, emplearemos esquemas basados en la combinación de operaciones de interpolación y repetición de estos parámetros. La hipótesis de partida sobre la que trabajaremos es que las operaciones de interpolación en el dominio de los parámetros producen transiciones suaves, respecto a la interpolación en el dominio de los parámetros MFCC, que aproximan mejor la evolución natural de la voz.

No obstante, como mencionamos con anterioridad, la interpolación directa de los parámetros LPC puede dar lugar a filtros inestables que originen graves distorsiones sobre el espectro LPC. Así pues, la interpolación de los conjuntos LPC se lleva a cabo en el dominio LSF, ya que es adecuado para este tipo de operación. En particular, se evaluaron todas las posibles combinaciones basadas en repetición e interpolación lineal de estos parámetros, siendo las combinaciones mostradas en la tabla 7.5 aquellas que obtuvieron mejores resultados.

p_{res}	0	2	4	6	8	10	12	14	16
WAcc(%)	94.35	96.45	97.58	98.48	98.75	98.94	98.90	98.92	98.88

Tabla 7.6: Resultados de precisión de reconocimiento sin pérdidas para el transparametrizador iLBC con diferentes ordenes de análisis LPC (p_{res}) sobre la señal de excitación.

7.4.4. Resultados Experimentales

El rendimiento del transcodificador propuesto está supeditado al orden del análisis LPC llevado a cabo sobre la señal de excitación en el procedimiento descrito en el apartado 7.4.1. La tabla 7.6 muestra la sensibilidad de la precisión de reconocimiento a la modificación del orden del análisis LPC p_{res} realizado sobre la excitación. Estas pruebas de reconocimiento fueron llevadas a cabo bajo el marco experimental definido en la sección 4.3 sin introducir pérdidas de paquetes. Cuando no se extrae información espectral de la excitación ($p_{res} = 0$), la expansión de ancho de banda sobre los parámetros LPC origina una notable pérdida de rendimiento. Esta situación se corrige a medida que aumentamos el orden del análisis, siendo suficiente un análisis LPC de orden $p_{res} = 10$, con el que se consigue una tasa de precisión de reconocimiento del 98.94 % (similar a la obtenida sobre voz decodificada, 98.96 %). Puesto que el transparametrizador propuesto utiliza vectores de características compatibles con los extraídos por el FE, podríamos utilizar el entrenamiento de éste para llevar a cabo la prueba de reconocimiento, obteniendo en este caso un resultado de 98.90 %.

La motivación para desarrollar el transparametrizador, propuesto en esta sección, vino dada por la hipótesis de que el dominio de los parámetros del códec es más adecuado para llevar la mitigación de pérdidas que el de los vectores de características. En esta sección, verificamos esta hipótesis mediante la realización de pruebas experimentales. Para ello, utilizamos como punto de partida los resultados obtenidos mediante la interpolación lineal y repetición por el vector más cercano (NFR, *Nearest Frame Repetition*) sobre los vectores de características FE obtenidos, en este caso, a partir de voz decodificada. Las tablas 7.7 y 7.8 muestran los resultados obtenidos empleando las técnicas NFR e interpolación lineal, respectivamente. En este caso, al contrario de lo que sucedía sobre en los codificadores CELP, la técnica NFR consigue resultados superiores a la interpolación lineal. Puesto que no existe error de propagación al utilizar este codificador, la situación se asemeja más a la de DSR. Bajo este esquema, Tan *et al.* [157] probaron que la interpolación lineal, la cual introduce un menor error cuadrático medio que la técnica NFR, origina un mal alineamiento del algoritmo de Viterbi empleado en el reconocimiento. Por

Tasa de Pérdidas	*Long. media ráfaga*			
	1	*2*	*3*	*4*
5 %	98.84	98.33	97.93	97.30
10 %	98.77	97.80	96.98	95.39
15 %	98.72	97.35	95.65	93.50
20 %	98.63	96.55	94.55	91.48

Tabla 7.7: Resultados de precisión de reconocimiento (WAcc) al aplicar la reconstrucción basada en repetición NFR sobre los vectores de características extraídos de voz decodificada con iLBC 15.2 kbps.

Tasa de Pérdidas	*Long. media ráfaga*			
	1	*2*	*3*	*4*
5 %	98.85	98.27	97.80	96.93
10 %	98.80	97.63	96.62	94.64
15 %	98.75	97.19	95.17	92.95
20 %	98.67	96.33	93.62	90.12

Tabla 7.8: Resultados de precisión de reconocimiento (WAcc) al aplicar la reconstrucción basada en interpolación lineal sobre los vectores de características extraídos de voz decodificada con iLBC 15.2 kbps.

el contrario, la reconstrucción NFR aproxima mejor estas duraciones, consiguiendo así un mejor rendimiento ante pérdidas.

Finalmente, los resultados correspondientes a los algoritmos de mitigación propuestos en la sección 7.4.3 se muestran en las tablas 7.9, 7.10 y 7.11. En términos generales, el rendimiento de cualquiera de estas técnicas es similar o superior al de la técnica NFR sobre los vectores de características utilizados durante el reconocimiento. Aunque el algoritmo PLC_1 (tabla de resultados 7.9) lleva a cabo la interpolación lineal de todos los parámetros, sus resultados son superiores a realizar la interpolación en el dominio de los vectores de características (tabla de resultados 7.8). En principio, cabría esperar que estos resultados fueran similares, sin embargo, hay que resaltar que la interpolación se realiza en dominios distintos que no son equivalentes (nótese que los vectores de características son una representación del espectro logarítmico). Además, los algoritmos PLC_2 y PLC_3 (tablas de resultados 7.10 y 7.11, respectivamente), los cuales hacen uso de una combinación de las técnicas NFR e interpolación lineal de parámetros, consiguen introducir mejoras estadísticamente significativas sobre la técnica NFR (tabla de resultados 7.7).

Tasa de Pérdidas	*Long. media ráfaga*			
	1	*2*	*3*	*4*
5 %	98.80	98.60	98.01	97.41
10 %	98.77	98.06	97.10	95.73
15 %	98.72	97.56	95.80	93.74
20 %	98.68	97.10	94.95	91.74

Tabla 7.9: Resultados de precisión de reconocimiento (WAcc) al aplicar el transcodificador para iLBC con el algoritmo de mitigación de pérdidas PLC_1.

Tasa de Pérdidas	*Long. media ráfaga*			
	1	*2*	*3*	*4*
5 %	98.84	98.62	98.11	97.61
10 %	98.76	98.08	97.25	96.10
15 %	98.72	97.62	96.01	94.30
20 %	98.69	97.17	95.60	92.67

Tabla 7.10: Resultados de precisión de reconocimiento (WAcc) al aplicar el transcodificador para iLBC con el algoritmo de mitigación de pérdidas PLC_2.

Tasa de Pérdidas	*Long. media ráfaga*			
	1	*2*	*3*	*4*
5 %	98.85	98.62	98.14	97.74
10 %	98.76	98.22	97.35	96.34
15 %	98.72	98.02	96.29	94.65
20 %	98.69	97.27	95.98	93.32

Tabla 7.11: Resultados de precisión de reconocimiento (WAcc) aplicando el transcodificador para iLBC con el algoritmo de mitigación de pérdidas PLC_3.

7.5. Resumen de Resultados y Conclusiones

En este capítulo se han presentado diversas técnicas de transparametrización para distintos codificadores. Así, la primera parte del capítulo se destinó al desarrollo de un esquema de transparametrización CELP que se aplicó a los codificadores AMR (12.2 kbps) y G.729 (8 kbps). En este caso, veíamos que esta solución B-NSR consigue delimitar las longitudes del error de propagación para cada componente del vector de características, lo que nos permite adaptar eficientemente las técnicas de mitigación MMSE propuestas en el capítulo previo. En la segunda parte del capítulo se desarrolló un transparametrizador para el codificador iLBC. En este segundo caso, el codificador no introduce propagación de error, sin embargo, el uso del transparametrizador permite desarrollar sencillos algoritmos de mitigación, basados en repetición e interpolación de los parámetros del códec, que mejoran las prestaciones ante pérdidas de paquetes.

Al igual que hicimos en la sección 6.7, las figuras 7.6 y 7.7 recogen un resumen de los resultados de precisión de reconocimiento para los codificadores AMR y G.729, respectivamente. Las condiciones de canal se corresponden con la diagonal de las tablas de resultados mostrados a lo largo del presente capítulo, y que se resumen en la tabla 6.23. Los resultados denominados *baseline* corresponden con los resultados de referencia obtenidos en el capítulo 4 a partir de voz decodificada. También se muestran los resultados obtenidos por otras técnicas encontradas en la literatura. Así, los resultados etiquetados como UD (*Uncertainty Decoding rule*) corresponden a la regla de decodificación con incertidumbre propuesta por Ion y Haeb-Umbach para NSR [163] (véase sección 6.4.3), mientras que los resultados etiquetados como WVA se refieren a la solución basada en el algoritmo *weighted Viterbi*propuesta por Cardenal *et. al* [173] aplicada en el contexto NSR (a partir de voz decodificada). En este último caso la reconstrucción realizada se corresponde con la técnica NFR, mientras que los valores de confianza ρ_t se establecen mediante la expresión $\rho_t = \alpha^{\tau}$, donde $\tau = \min(T_{VL} - t, t)$ $(t = 0, \ldots, T_{VL})$. El valor óptimo de α se determinó de forma empírica como $\alpha = 0{,}7$. Finalmente, también se muestran los resultados obtenidos por la arquitectura DSR descrita en [35], resultados que podemos considerar como referencia superior para las soluciones NSR propuestas.

Como se puede comprobar, las propuestas B-NSR basadas en la estimación MMSE (etiquetadas como *B-NSR FBMMSE+WVA* y *B-NSR FBMMSE+Soft-Data*) consiguen mejorar los correspondientes resultados obtenidos a partir de voz decodificada (soluciones *FBMMSE+WVA* y *FBMMSE+Soft-Data*). La aplicación de estas técnicas exige el conocimiento a priori de la distorsión introducida por el codificador, información que reside en

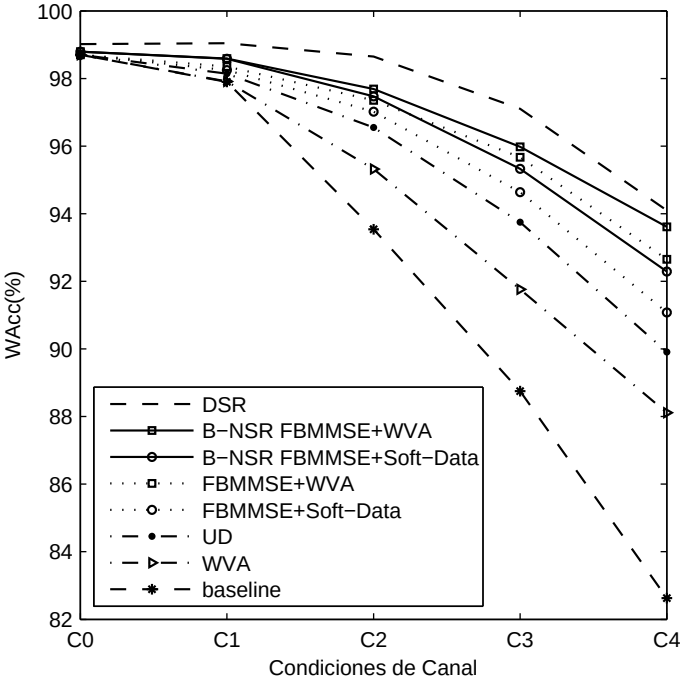


Figura 7.6: Resumen de resultados al aplicar técnicas de mitigación de pérdidas sobre esquemas NSR basados en el codificador AMR 12.2.

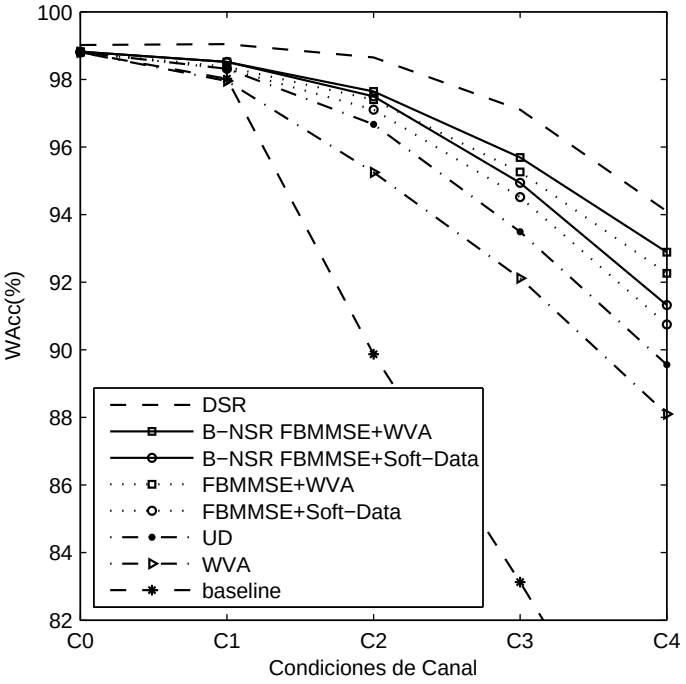


Figura 7.7: Resumen de resultados al aplicar técnicas de mitigación de pérdidas sobre esquemas NSR basados en el codificador G.729.

las tablas de probabilidad de observación. En el cómputo de probabilidades de observación que introdujimos en la sección 6.4.3, se hacía preciso almacenar una tabla para cada tamaño de ráfaga de pérdidas y posición posterior a la ráfaga, resultando en un número total de $L_{burst} \times T_{EP}$ tablas. Además, estas técnicas introducen una latencia de T_{EP} vectores de características frente a las técnicas basadas en interpolación lineal o NFR. En este sentido, la propuesta B-NSR aquí presentada consigue la ventaja adicional de limitar el valor de T_{EP} (considerado en el capítulo previo como $T_{EP} = 20$) a $T_{MA} + 1$ (donde T_{MA} adopta los valores de 2 y 4 para los codificadores AMR 12.2 y G.729, respectivamente), consiguiendo así una importante reducción de la latencia y número de tablas a almacenar.

La tabla 7.12 resume los resultados ante pérdidas de paquetes para el transcodificador iLBC. En particular, se presentan las condiciones de pérdidas establecidas en la tabla 6.23. Como referencia inferior se muestran los resultados de reconocimiento obtenidos a partir de voz decodificada, mientras que la referencia superior viene dada por la arquitectura DSR que emplea el FE básico. Los resultados restantes hacen referencia a diferentes técnicas de mitigación aplicadas sobre los vectores de características (*I-MFCC* y *NFR-MFCC*) o sobre los parámetros utilizados por el transparametrizador (*Transp.* PLC_3), ofreciendo esta última unos resultados superiores.

Finalmente, la tabla 7.13 recoge los resultados baseline (obtenidos del capítulo 4) y los mejores resultados obtenidos para cada uno de los esquemas de transparametrización de este capítulo. En esta comparativa hemos de resaltar que las soluciones propuestas consiguen reducir notablemente las diferencias con DSR y su esquema de mitigación estándar (repetición NFR). No obstante, mientras que la propagación de error propia de los codificadores CELP exige la aplicación de técnicas de mitigación de cierta complejidad (*FBMMSE+WVA*), el esquema transparametrizador iLBC, al no presentar dependencias intertrama, logra resultados similares mediante técnicas basadas en interpolación, aunque, eso sí, utilizando una tasa de codificación superior.

Téc. de Mitigación	*Condiciones*					*Valor Medio*
	C0	*C1*	*C2*	*C3*	*C4*	
baseline	98.96	98.56	96.35	92.43	87.11	94.68
I-MFCC	98.96	98.85	97.63	95.17	90.12	96.14
NFR-MFCC	98.96	98.84	97.80	95.65	91.48	96.54
Transp. PLC_3	98.94	98.85	98.22	96.29	93.32	97.12
DSR	99.04	99.04	98.65	97.10	94.10	97.59

Tabla 7.12: Resumen de resultados de precisión de reconocimiento (WAcc) ante condiciones de canal con pérdidas para la arquitectura NSR utilizando el codificador iLBC. Las técnicas evaluadas son: *baseline* o reconocimiento a partir de voz decodificada; *I-MFCC*, interpolación lineal sobre los vectores de características; *NFR-MFCC*, repetición del vector de características más cercano; *Transp.* PLC_3, propuesta de transparametrización iLBC al aplicar el algoritmo iLBC en el dominio de los parámetros. Por último, se incorporan los resultados de la arquitectura *DSR* empleando el FE básico.

Arquitectura de Recono.	*Tasa de Códec*	*Condiciones*					*Valor Medio*
		C0	*C1*	*C2*	*C3*	*C4*	
baseline G.729	*8 kbps*	98.81	98.02	89.87	83.13	75.98	89.16
baseline AMR	*12.2 kbps*	98.68	97.93	93.97	88.55	83.07	92.44
baseline iLBC	*15.2 kbps*	98.96	98.56	96.35	92.43	87.11	94.68
B-NSR G.729	*8 kbps*	98.82	98.52	97.64	95.69	92.89	96.71
B-NSR AMR	*12.2 kbps*	98.79	98.59	97.69	95.96	93.61	96.93
B-NSR iLBC	*15.2 kbps*	98.94	98.85	98.22	96.29	93.32	97.12
DSR FE	*4.75 kbps*	99.04	99.04	98.65	97.10	94.10	97.59

Tabla 7.13: Resumen de los mejores resultados obtenidos para los esquemas de transcodificación propuestos. Técnicas de mitigación aplicadas: *FBMMSE+WVA* para las arquitecturas *B-NSR G.729* y *B-NSR AMR*; algoritmo PLC_3 para *B-NSR iLBC*; repetición *NFR* para *DSR FE*.

Capítulo 8

Conclusiones

El interés del presente trabajo se centra en el análisis y tratamiento de las degradaciones causadas por las redes de conmutación de paquetes IP sobre el rendimiento de un sistema de reconocimiento remoto de voz, concretamente, cuando éste se lleva a cabo a partir de voz codificada. En este caso, la principal degradación que sufren estos sistemas viene dada por la pérdida de paquetes que afecta a la precisión en el reconocimiento del habla. En este trabajo se proponen diferentes técnicas orientadas a evitar, reducir y compensar el efecto negativo producido por las pérdidas.

8.1. Conclusiones

- Hemos descrito el estado actual del reconocimiento remoto de la voz, indicando las posibles arquitecturas así como las ventajas e inconvenientes de éstas. El hecho de que la arquitectura DSR esté diseñada desde un principio con el objetivo de reconocimiento, la convierte en la solución más potente. No obstante, la escasa aceptación por parte de la industria de los estándares DSR hace pronosticar que los futuros sistemas de reconocimiento remoto adoptarán una arquitectura NSR, ya que esta última se encuentra soportada por las, cada vez más extendidas, plataformas VoIP.

- Fundamentalmente, existen tres causas de degradación en el entorno IP impuestas por el medio de transmisión: la latencia, la dispersión en el retardo temporal y la pérdida de paquetes. En una aplicación en tiempo real, como el reconocimiento remoto, las posibles degradaciones pueden verse resumidas en pérdidas de informa-

ción, es decir, en la llegada o no de un paquete antes de un cierto tiempo de espera, pasado el cual no podrá ser utilizado por la aplicación.

- La pérdida de rendimiento ocasionada por la distorsión introducida por el proceso de codificación/decodificación puede ser notablemente reducida mediante el uso de un conjunto de modelos acústicos obtenidos a partir de voz codificada. De esta manera, la principal desventaja de la arquitectura NSR, en comparación con DSR, viene dada por la escasa robustez de los esquemas de codificación frente a pérdidas de paquetes.

- El rendimiento de la arquitectura NSR queda supeditado al esquema de codificación utilizado. El comportamiento de los codificadores basados en el paradigma CELP introduce fuertes dependencias intertrama que degradan considerablemente la precisión de reconocimiento en un entorno con pérdidas. Por contra, el esquema de codificación iLBC, que no presenta dependencias intertrama, es más robusto ante pérdidas, aunque para ello emplea una tasa de codificación superior.

- De forma general, los algoritmos de mitigación integrados por los codificadores se basan en consideraciones perceptuales que no son apropiadas para el reconocimiento. Así, durante pérdidas de varios paquetes consecutivos, dichos algoritmos realizan un apagado progresivo de la voz, también conocido como *muting*, que deriva en un aumento de los errores de inserción en el reconocedor (silencios artificiales).

- El efecto negativo de las pérdidas de paquetes sobre un esquema de codificación de voz CELP no sólo se refleja en las pérdida de las muestras correspondientes a dicho segmento, sino que causa una propagación de error en las muestras posteriores a dicha pérdida.

- Los esquemas de mitigación de pérdidas para NSR basados en el emisor sólo quedan justificados si se traducen en un rendimiento superior tanto en calidad perceptual como en precisión del reconocimiento de voz. No tiene sentido introducir modificaciones en el emisor que estén únicamente orientadas a incrementar el rendimiento de los sistemas de reconocimiento remoto, puesto que la principal ventaja de NSR es que no precisa introducir una aplicación en el terminal cliente específicamente orientada al reconocimiento.

- La combinación del esquema de codificación iLBC y CELP, mediante el intercalado de un cierto número de tramas CELP entre dos tramas adyacentes iLBC, redunda en un esquema de codificación con una tasa de codificación menor y con una cierta robustez frente a pérdidas. El grado de robustez vendrá dado por la separación de las tramas iLBC, las cuales eliminan completamente la propagación de error hacia delante.

- Alternativamente, el efecto de la propagación de error en los codificadores CELP puede mitigarse introduciendo códigos FEC específicos como información colateral. Estos códigos consisten en codificaciones de las muestras previas a la trama actual. En particular, la codificación multipulso permite eliminar casi completamente el error de propagación empleando un reducido número de pulsos.

- El esquema FEC basado en codificación multipulso permite desarrollar potentes esquemas de cuantización reutilizando los parámetros CELP de la trama actual, lo que redunda en un reducido flujo de información colateral a transmitir.

- Las técnicas basadas en la combinación de tramas y códigos FEC consiguen aumentar, principalmente, la precisión de reconocimiento ante pérdidas aisladas o de corta duración. Sin embargo, incrementar el rendimiento de los sistemas NSR ante pérdidas largas exige sustituir los algoritmos de mitigación basados en el receptor, e integrados en el códec (repetición y apagado progresivo de la señal), por otros orientados al reconocimiento de voz.

- Una de las soluciones más sencillas para NSR, dentro de las técnicas basadas en el receptor, consiste en sustituir aquellos vectores correspondientes a una pérdida por una versión interpolada. Este esquema consigue mejorar notoriamente la precisión de reconocimiento ante pérdidas cortas y largas. No obstante, esta técnica no tiene en cuenta cómo afecta la propagación de error a los vectores de características posteriores a una pérdida.

- El error propagado por una pérdida puede caracterizarse en función de tres factores: la longitud de la pérdida, la distancia del vector de características considerado respecto al final de la pérdida y el propio vector. Considerando esta distorsión y un modelo HMM de la voz, se puede emplear un proceso de estimación MMSE para la reconstrucción de los vectores de características afectados por una pérdida. Este

método de reconstrucción consigue resultados superiores a las técnicas basadas en interpolación.

- 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. Estas técnicas, denominadas técnicas basadas en el reconocedor, aprovechan el potente modelo de voz incluido en el reconocedor para mitigar las pérdidas. Además, una ventaja adicional de la técnica MMSE es que se puede complementar con las técnicas basadas en el reconocedor mediante la asignación de valores de confianza a partir de las distribuciones de probabilidad calculadas durante el proceso de estimación. En particular, podemos emplear dos métodos para la asignación de los valores de confianza a partir de la reconstrucción MMSE:

  - El primero de ellos, denominado *soft-data*, integra mediante ciertas aproximaciones matemáticas los valores de varianza de las estimas MMSE como incertidumbre en el proceso de reconocimiento.
  - Por contra, el segundo método, llamado algoritmo *Weighted Viterbi*, establece ciertos pesos heurísticos durante el proceso de reconocimiento sobre aquellos vectores de características reconstruidos. En esta última solución, los pesos se pueden computar a partir de la entropía de las distribuciones de probabilidad empleadas durante la estimación MMSE. De este modo, aquellas reconstrucciones más precisas y que, por tanto, aportan más información se ponderan más en el proceso de reconocimiento.

- El empleo de un esquema de transparametrización para el codificador iLBC combinado con un algoritmo de mitigación sencillo (basado en repetición e interpolación) en el dominio de los parámetros del códec permite obtener sustanciales mejoras sobre el reconocimiento a partir de voz decodificada, resultados ligeramente inferiores a los del estándar DSR basado en el FE básico.

- Finalmente, el empleo de esquemas de transparametrización para los codificadores basados en el paradigma CELP permite distinguir diferentes longitudes de propagación entre las diferentes componentes de los vectores de características extraídos. Así pues, podemos distinguir entre aquellos transparametrizados a partir de los coeficientes LPC y los derivados a partir de la energía de la excitación. El primer grupo

sólo se verá afectado por la propagación de error en el caso de que se empleen predictores de media móvil durante el proceso de cuantización de los coeficientes LPC, mientras que el segundo presentará una propagación más larga debida a su dependencia de las muestras previas de la excitación. Esta diferenciación permite llevar a cabo una adaptación eficiente de la reconstrucción MMSE, tanto en términos de precisión de reconocimiento como en términos computacionales y de retardo.

8.2. Contribuciones

Las principales aportaciones de esta tesis se pueden resumir en:

- Estudio de la caracterización del rendimiento de los sistemas de reconocimiento remoto a partir de voz decodificada en redes IP [184].
- Esquemas de codificación de voz robustos basados en el mezclado de tramas [185, 186].
- Técnicas de mitigación de pérdidas basadas en el uso de esquemas FEC multipulso con un incremento moderado de la tasa de codificación [187].
- Técnicas de reconstrucción de pérdidas basadas en estimación MMSE [188, 189].
- Técnicas de mitigación basadas en el propio reconocedor para introducir la confianza de las estimas realizadas [188, 190].
- Esquema de transparametrización y mitigación de pérdidas en el dominio de los parámetros para el codificador iLBC [191].
- Esquema de transparametrización para codificadores CELP y adaptación eficiente de las técnicas de mitigación basadas en MMSE [188].

8.3. Trabajo Futuro

El trabajo realizado sobre los esquemas FEC basados en multipulso se ha mostrado muy prometedor. En este trabajo las técnicas de cuantización presentadas explotaban las relaciones entre los parámetros CELP y la descripción multipulso con el fin de obtener una codificación eficiente de esta última. No obstante, estos esquemas trabajaban a partir de una descripción establecida por un único pulso. Un área interesante de investigación

sería la exploración de técnicas eficientes de cuantización de más de un pulso que no sólo explotaran las relaciones de éstos con los parámetros CELP, sino también las relaciones entre los diversos pulsos.

Además del efecto degradante introducido por las pérdidas de paquetes, otro problema del reconocimiento remoto es una mayor presencia de ruido acústico, debido a que se puede operar prácticamente desde cualquier lugar en el que se encuentre el usuario. Así, el efecto de las pérdidas puede verse potenciado por el ruido acústico en la arquitectura NSR ya que esta se encuentra supeditada al rendimiento del codificador de voz en estos entornos. Por ello, sería conveniente evaluar el rendimiento de las técnicas de mitigación basadas en el receptor, así como los esquemas de transparametrización, en presencia de ruido acústico. Igualmente, sería interesante el desarrollo de técnicas de estimación MMSE que combatieran de forma conjunta la contaminación acústica de la voz y aquellas degradaciones derivadas de las pérdidas de paquetes.

Otra posibilidad que actualmente estamos investigando es la de reemplazar la parametrización FE por su versión avanzada AFE. En este caso el AFE lleva a cabo un procesado que tiene como objetivo reducir la influencia del ruido acústico. Sin embargo, es pertinente replantear el problema ya que en este caso las pérdidas también dan lugar a una desactualización de las estados internos del extractor AFE, la cual origina una pérdida de rendimiento. En este sentido, una solución atractiva sería el desarrollo de un esquema transparametrizador que empleara las técnicas de reducción de ruido acústico del AFE.

De igual forma, 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 que no sólo tuvieran en cuenta los efectos originados por las pérdidas, sino también las condiciones acústicas resultaría de gran utilidad en este ámbito.

Por último, las técnicas de mitigación MMSE propuestas en este trabajo se han aplicado sobre el dominio de los vectores de características utilizados en el reconocimiento. Sin embargo, estas técnicas se podrían trasladar al dominio de los parámetros del codificador, de modo que pudieran ser utilizadas como algoritmos de mitigación para realzar la calidad perceptual de la voz reconstruida. En este sentido, también sería interesante explorar las posibilidades de reducción de complejidad de estas técnicas, de manera que pudieran ser fácilmente integrables como algoritmos de mitigación de pérdidas en los terminales VoIP.

Capítulo 9

Conclusions

This chapter aims to summarize the main conclusions drawn as a result of the work presented in the different chapters. We review the major contributions of this work and present several suggestions for future work.

9.1. Conclusions

- The current state of remote speech recognition, the possible architectures, and their advantages and disadvantages have been outlined. The DSR architecture is among them the most suitable solution since it was originally designed for speech recognition. However, in the near future, remote speech recognition systems could most likely implement NSR architectures which are supported by the wider and wider spread of VoIP platforms.

- There are three main types of degradation over IP networks caused by the transmission mode: latency, jitter and packet loss. In real-time applications such as remote speech recognition, all of these possible source of degradation can be reduced to losses of information, i.e., whether a packet arrives or not in a waiting time interval, after which the application can no longer use it.

- Lower performance caused by the distortion in the coding/decoding process can be notably reduced by the use of acoustic models obtained from coded speech. Therefore, the main drawback of the NSR architecture compared to DSR is the fact that the coding schemes lack robustness to packet losses.

- The performance of the NSR architecture depends on the used coding scheme. The behavior of CELP coders involves strong inter-frame dependencies that considerably diminish the recognition accuracy in the case of packet losses. On the contrary, the iLBC codec, which does not have inter-frame dependencies, is more robust to losses although it uses a higher bit-rate.

- Generally, the packet loss concealment algorithms implemented in the decoders are unsuitable for recognition tasks. Such algorithms are based on perceptual considerations that are not appropriate for recognition. Thus, when several consecutive packets are lost, decoders progressively mute, leading to an increase on the insertion errors in the recogniser (artificial silences).

- The negative effect of packet losses on a CELP coding scheme is not only given by the corresponding information loss. In addition, packet losses cause error propagation in the frames after a loss.

- Sender-driven loss concealment techniques for NSR are only justifiable if they optimise both perceptual quality and speech recognition accuracy. The main advantage of the NSR architecture is that no terminal changes are required, so modifications can not be justified only by recognition improvements.

- The combination of the iLBC codec and the CELP coding scheme, through insertion of a number of CELP frames between two adjacent iLBC frames, results in a coding scheme with a lower bit-rate and notably robustness against losses. The robustness will be determined by the distance between iLBC frames, which entirely remove forward error propagation.

- On the other hand, the effect of error propagation on CELP coders can be mitigated introducing specific FEC codes as side information. These FECs contain coded samples of the previous excitation. In particular, multipulse coding almost fully eliminates the propagation error using a reduced number of pulses as previous excitation.

- The FEC codes based on multipulse coding can be efficiently quantized reusing the CELP parameters of the given frame, leading to a small increment of the codec bit-rate.

- Frame combination techniques and FEC codes increase the perceptual quality of decoded speech and the speech recognition performance. In this last case, the improvements are mainly for isolated losses or short duration losses. However, to tackle long losses, it is necessary to replace receiver-based concealment algorithms integrated in the decoder (repetition and signal muting) by other algorithms oriented to speech recognition.

- Among the receiver-based techniques for NSR, one of the easiest approaches is to substitute the lost vectors for interpolated versions. This scheme outstandingly improves recognition accuracy in the case of short and long losses. Nevertheless, this technique disregards how the propagation error affects to feature vectors after a loss.

- The propagation error in a CELP-based NSR architecture can be characterized by considering the loss length, the distance of the feature vector being computed from the end of the loss, and the vector itself. Implementing this loss model, an HMM-based MMSE estimation can be applied in order to reconstruct those feature vectors affected by a packet loss. Such estimation, which takes into account the temporal speech evolution by means of an HMM, obtains better results than interpolation techniques.

- It is possible to treat the losses directly inside the recogniser assigning confidence values that will be taken into account during recognition. A more powerful speech model, which is given by the recognizer itself, can be thus implicitly exploited. Thus, a further advantage of the HMM-based MMSE technique is the fact that it can be complemented with techniques based on the recogniser by assigning confidence values during the estimation process. Two methods for assigning confidence values to the MMSE estimates can be used:

  - The first method, called soft-data, incorporates the variance values of the MMSE estimates as uncertainty in the recognition process.
  - On the contrary, the second method, called weighted Viterbi algorithm, determines heuristic weights for the reconstructed feature vectors which are applied to the HMM Gaussian mixtures. These weights can be calculated from the entropy of the probability distributions used for the MMSE estimation.

- The performance of the NSR architecture significantly improves with the combined use of a transparameterization scheme for the iLBC coder and a simple concealment algorithm, based on repetition and interpolation, in the codec parameter domain. The results are only slightly inferior to those ones obtained by the DSR standard.

- Finally, with the use of transparameterization for CELP coders, different propagation lengths can be differentiated for every component of the feature vectors. Therefore, those components obtained from LPC coefficients and those ones derived from the excitation energy can be distinguished. The first group will be only affected by the propagation error if moving average predictors are used during the quantization of LPC coefficients, whereas the second group will show a longer propagation due to the dependency from the previous excitation samples. Such discrimination allows an efficient adaptation of the MMSE reconstruction (developed for the recognition of decoded speech) in terms of recognition accuracy and delay as well as in computational terms.

9.2. Contributions

The main contributions of this Ph.D. dissertation can be summarized as follows:

- A study of the performance of remote recognition with decoded speech over IP networks [184].

- Packet loss robust coding schemes based on frame combination [185, 186].

- Packet loss concealment techniques based on multipulse FEC schemes with a moderate increment of bit-rate [187].

- Packet loss reconstruction techniques based on MMSE estimation [188, 189].

- Packet loss concealment techniques based on the introduction of estimate confidence measures in the speech recognition process [188, 190].

- Transparameterization and packet loss concealment in the parameter domain for iLBC coding scheme [191].

- Transparameterization for CELP coders and an efficient adaptation of concealment techniques based on MMSE [188].

9.3. Future Work

The work done on FEC schemes based on multipulse is encouraging. The quantization techniques presented in this thesis exploit relations between the CELP parameters and the multipulse description of the excitation in order to obtain an efficient coding of the later. However, these schemes worked on a description determined by a single pulse. Efficient multi-pulse quantization techniques that not only exploit the relations between pulses and CELP parameters, but also the relations between pulses, may be an interesting field of research.

In addition to the degrading effect of packet loss, the likely presence of acoustic noise is another problem that must be addressed in ubiquitous speech recognition. Acoustic noise in the NSR architecture can worsen the degrading effects of packet losses, since this architecture is subject to the speech codec performance in these environments. For this reason, it would be convenient to evaluate the performance of concealment techniques based on the receiver, as well as transparameterization schemes, when the speech signal is distorted by acoustic noise. Therefore, it would be interesting to develop MMSE estimation techniques that jointly treat acoustic noise and the degradation derived from packet loss.

The possibility of replacing feature extraction carried out by the FE front-end by its advanced version AFE in order to extract recognition features from decoded speech is presently under research. AFE is a front-end that conceals acoustic noise. Nevertheless, the problem needs the adoption of a new point of view since AFE also causes interframe dependencies, resulting in lower performance. Therefore, the development of a transcoding scheme that uses AFE techniques for acoustic noise reduction would be an attractive solution.

Similarly, packet loss treatment during speech recognition is a promising field of research. It would be useful to develop more elaborated heuristics and techniques that consider not only loss effects, but also acoustic conditions, to assign confidences values.

Finally, MMSE concealment techniques proposed in this work have been applied to the domain of feature vector domain. However, these techniques could be also applied directly to the codec parameters in order to be used as concealment algorithms to optimise perceptual quality of the reconstructed speech. It would be also interesting to reduce the complexity of these techniques, to be easily integrated in mobile terminals.

Apéndice A

Summary

A.1. Introduction

In recent years automatic speech recognition (ASR) has been strongly developed. Research on the field of recognition, on the one hand, and the development of computers -faster and cheaper with time- on the other; have given birth to undreamed-of applications. Nowadays, automatic dictation applications, voice management of operating systems or in-car voice control systems have become reality. Moreover, the exponential growth of the Internet and the variety of access technologies provide speech recognition systems with a wide range of possibilities.

IP Packet switching networks have originated a global network of networks or Internet. Voice transmission over this type of network, called Voice over IP (VoIP), has shown strong growth during the past years, and it has turned into one of the key aspects of the current state of telecommunications. In parallel with voice and data convergence provided by VoIP platforms, new standards of wireless Internet access have led to a convergence of IP and mobile telephony networks. Thus, mobile telephones with Bluetooth (PAN, Personal Area Network), wi-fi (LAN, Local Area Network) and UMTS (Universal Mobile Telecommunication System) have entered the market. Especially thanks to the increasing number of WLAN networks (Wireless LAN) in recent years, in the near future VoIP technologies are expected to spread into the wireless domain through this type of local networks [2]. In the mid-term, this trend will be strengthened by the development of other radio access technologies, such as Wi-MAX (IEEE 802.16), Mobile Broadband Wireless Access (MBWA also known as IEEE 802.20), Ultra Wideband (UWB o IEEE 802.15) or LTE (Long Term Evolution). Although such technologies cover different market segments,

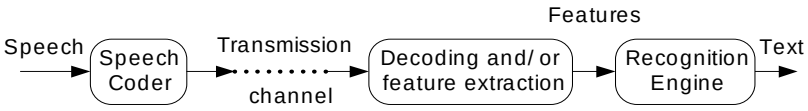
their network boundaries are more and more unclear and they no longer compete but complement each other. Many of these technologies will be incorporated in the future mobile terminal that will be able to choose the one that better suits the access depending on the situation. This paradigm will give rise to a new concept of nomadic access, hybrid of fixed and mobile access, linked to the incorporation of IP technologies and provided by suppliers of these new technologies.

Thanks to the convergence of wireless technologies, access to information on the move is nowadays a technological reality on the increase. Nevertheless, mobile phone constraints, such as the lack of keypad for size reasons, hinder the access to remote services. Oral interaction with such services arises as a new faster and more natural means of access to information with the help of automatic speech recognition that offers an interactive service and fast access to information, what benefits the user, doing without the assistance operator at the other side, what benefits the provider. Unfortunately, there are several problems to install an automatic speech recognition subsystem into mobile terminals. It is mainly their size restrictions what limits the computation capacity and, therefore, the recogniser power and flexibility. The possibility of remote speech recognition, i.e. outside the terminal, emerged to overcome these obstacles.

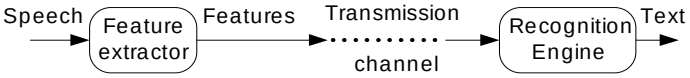
Remote Speech Recognition (RSR) allows to circumvent these hardware constraints by moving the most complex computational tasks of speech recognition to a remote server. Moreover, the structure of a remote recognition system is well suited for the IP model, since it is the provider that implements the recogniser depending on its needs. Thus, the provider can incorporate new services adapted to the present needs of users. Under this point of view, low cost terminals with limited features are connected to powerful remote computers that carry out more complex tasks for them, what leads to an optimum use of centralised resources. As shown in Fig. A.1, there are two possibilities for the implementation of an RSR system [43]:

1. Network Speech Recognition (NSR). In this approach, the whole recognition system resides in the network. Thus, the client sends the speech signal, employing a conventional speech codec, to the server where recognition is carried out.

2. Distributed Speech Recognition (DSR). The client includes a local front-end that processes the speech signal in order to obtain the specific features used by the remote server (back-end) to perform recognition.



(a) NSR



(b) DSR

Figura A.1: Different architectures for the implementation of a Remote Speech Recognition (RSR) system. a) Network Speech Recognition (NSR); b) Distributed Speech Recognition (DSR).

In DSR, the feature extractor is applied directly to the speech signal to obtain a low dimensional representation with less redundant information. Although during the last years several standards have been issued [35, 37, 39, 40], the lack of DSR codecs in the existing devices supposes a barrier for its deployment.

On the other hand, the most direct implementation of an RSR system is the speech transmission to the extreme server where the recognition task is performed, i.e., the NSR architecture. In this case, RSR is considered a value-added service of VoIP, since the coded speech is not transmitted to establish a call session, but to have access to a particular service. The main advantage of this type of application is the use of emerging IP platforms, without modifying in any respect the client terminal. However, the application has at the same time some disadvantages, since the loss information that speech coding involves may affect performance. There are also some implicit problems in remote recognition. Among them, two of the most outstanding ones are acoustic noise -the acoustic context of the terminal may vary- and degradations introduced by the communication channel. This dissertation is focused on the later, since IP networks design, which offer a best-effort service, does not guarantee to meet real time requirements -delay and jitter- nor reliability requirements of multimedia flow transmission -limited packet loss probability, availability of recovery mechanisms.

This thesis aims to develop a set of mechanisms to improve the performance of NSR systems considering the degradation of the current IP networks. In particular, such mechanisms should overcome the loss of VoIP packets over the network, optimising the performance of recognition tasks. In this sense, two types of measures can be adopted: sender-driven or receiver-based. The first type of these measures attempts to make speech coding schemes more robust against packet loss effects. In this case, modifications in the

coding scheme will be applied taking into account a double objective: improve both the subjective perceptual quality of coding systems and speech recognition. The second type of measures will develop loss concealment techniques for coded speech recognition. In this case, these techniques only require to modify the server structure, and therefore they focus on the improvement of the recogniser performance disregarding perceptual aspects.

The present summary is structured around six sections. In Section A.2 we will present the experimental framework and the baseline results that will be the starting point. Section A.3 describes several proposals for making speech codec more robust against packet losses. Such proposals focus on modifications of encoders to improve both the perceptual quality of decoded speech and the recognition from the later. However, if the coding scheme is altered, modifications on the sender should be also introduced. To overcome this problem, Section A.4 presents a set of loss concealment techniques in recognition that do not require changes in the sender. In the aforementioned sections, speech recognition is carried out from decoded speech. Nonetheless, it is possible to extract ASR parameters directly from codec parameters. Section A.5 explores this possibility, called transcoding, and proposes packet loss concealment techniques adapted to this scheme. Finally, conclusions, contributions and future lines of research of this work are shown in Section A.6.

A.2. Framework and Baseline Results

We have to take into account that conventional voice codecs show some disadvantages in speech recognition, because they were not designed for this function. Voice redundancies relative to speaker identity are still present. However, this kind of information is not relevant to speech recognition. Furthermore, the codec processing introduces some distortions into the reconstructed speech, which are detrimental to recognition tasks.

Nonetheless, recognition from synthesized voice shows highly interesting aspects. Firstly, since these codecs are used by conventional VoIP services, they will not have deployment problems. In addition, since the purpose of these codecs is to transmit the signal with the maximum possible perceptual quality, they include speaker characteristics that allow its identification. This last feature allows the implementation of interesting services, from a commercial point of view, as the authorship of bank operations or contract signing by voice.

Obviously, the performance of these systems will be determined by the codec robustness against the degradation introduced by IP networks. This section is devoted to explore

the performance of the most extensively used codecs with respect to IP networks disturbances, considering these last ones as bursts of packet losses.

A.2.1. Experimental Framework

The experimental setup is based on the framework proposed by the ETSI STQ-Aurora working group in [36]. The Aurora DSR front-end [35] provides a 14-dimension feature vector containing 13 MFCC (Mel Frequency Cepstral Coefficients) plus log-Energy. Furthermore, these vectors are extended by appending the first and second derivatives of the features. 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 8400 clean sentences and test is carried out over set A (4004 clean sentences distributed into 4 subsets).

In this work we have used two widely used CELP-based codecs: G.729A and AMR (Adaptive Multi-Rate) mode 12.2 kbps. In addition, iLBC (internet Low Bit-rate Codec) is also included due to its design is oriented to increase the robustness against packet losses. A brief description of these speech codecs and the number of frames per packet used in our experiments are given in the following paragraphs:

- The G.729A speech codec is based on conjugate-structure algebraic CELP (CS-ACELP) [81]. This codec computes a set of LPC coefficients for every 10-ms frame, achieving a bit-rate of 8 kbps. Moreover, G.729A is also used as the first layer of the G.729.1 scalable wideband codec [107]. Since G.729.1 defines a minimum of 2 frames per packet, we will use this packetization scheme.

- AMR is also an algebraic CELP (A-CELP) codec defined by ETSI, which can operate at 8 different bit-rates [78]. The frame size is 20 ms and the number of LPC sets depends on the bit-rate mode. In this work we have used the highest mode, defined by a bit-rate of 12.2 kbps and 2 LPC sets per frame, encapsulating one frame per packet (the same signal duration as described above for G.729.1).

- iLBC is a royalty free narrowband speech codec, developed by Global IP Solutions. This codec has two operative mode which have two different frame sizes 20 and 30 ms. The 20-ms mode operates at 15.2 kbps, while the 30-ms obtains a bit-rate of 13.33 kbps. iLBC is based on the linear prediction model, although, unlike

Codec	*Bit-rate (kbps)*	*Frame (ms)*	*Look-ahead (ms)*	*Packing (fram./pack.)*	*Delay (ms)*
AMR 4.75	4.75	20	5	1	25
AMR 7.95	7.95	20	5	1	25
AMR 12.2	12.2	20	0	1	20
G.729	8	10	5	2	25
iLBC 20 ms	15.2	20	0	1	20

Tabla A.1: Characteristics of speech codecs used in this work: bit-rate, frame size, look-ahead, packing (frames/packet) and algotihmic delay.

CELP-based codecs, it codes the excitation signal without introducing interframe dependencies.

As most of these codecs have several operative modes, table A.1 shows those ones tested and additional information about their bit-rates and frame sizes. Furthermore, the number of frames per packet used in each configuration is presented.

The channel burstiness exhibited by lossy packet networks is modelled by a 2-state Markov model [99], as shown in Figure A.2. The transition probabilities between states, p and q, can be set according to an average burst length (L_{avr}) and a packet loss ratio (P_{loss}) by means of the following expressions,

$$L_{burst} = \frac{1}{q} \qquad P_{loss} = \frac{p}{p+q} \tag{A.1}$$

The mitigation techniques presented along this work are compared under the channel conditions listed in Table A.3. These channel conditions simulate realistic situations of wired and wireless channels as presented in [95] and [154], respectively.

A.2.2. Baseline Results

The experimental framework is based on the simulation of a NSR (Network Speech Recognition) architecture. Once the information is coded, it is segmented into packets considering the packing information shown in Table A.1. The transmission channel degradation is indicated by means of a BFI (Bad Frame Indicator), so that the marked frames are discarded. The PLC (Packet Loss Concealment) algorithm of each codec will generate an approximation to the lost information derived from the previous correctly received packets. The more this approximation differs from the original information, the more degradation the synthesized speech will show.

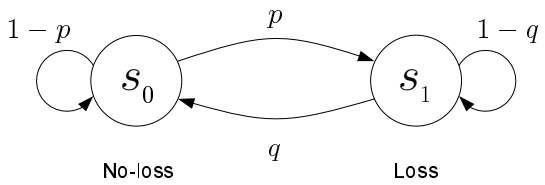


Tabla A.2: Gilbert Model.

Condition	p	q	L_{burst}	P_{loss}
C0	0	–	–	0 %
C1	0.0526	1.0000	1	5 %
C2	0.0555	0.5000	2	10 %
C3	0.0588	0.3333	3	15 %
C4	0.0625	0.2500	4	20 %

Tabla A.3: Packet loss test conditions.

As aforementioned, a two-state model is used in order to simulate the behavior of the channel. Following this model, a wide range of channel conditions were emulated with the selection of the mean loss burst length (L_{burst}) and the total loss probability ($P_{loss}(\%)$) from the sets [1 2 3 4] and [5 10 15 20], respectively.

Tables A.4, A.5 and A.6 show the results for the AMR modes with 4.75, 7.95 and 12.2 kbps, respectively. The corresponding results obtained for clean condition, i.e. without packet loss, are 98.54 %, 98.68 % and 98.70 %. The results in Tables A.7 and A.8 are those ones obtained with G.729 and iLBC, respectively. In clean condition, G.729 and iLBC obtain 98.70 % and 98.96 %, respectively. Finally, Table A.9 shows the results for the DSR standard [35], which obtains a result of 99.04 without packet loss.

The WAcc result obtained directly from original speech, i.e. whithout using any coding scheme, is 99.02 and therefore the quantization stage used in DSR does not introduce any performance reduction. In this sense, the performance reductions caused by coding are not very significant. However, this situation changes rapidly when packet losses are considered. For a given packet loss rate, the longer the average burst length the greater the reduction in performance.

In general, the codecs based on CELP (Coded Excitation Linear Predictor), as AMR, G.729 and G.723.1, do not achieve an optimum performance because they use predictive techniques. In particular, these codecs use pitch filters that propagate the distortions caused by packet losses. For example, G.729 achieves excellent results in clean conditions

Packet Loss Rate	*Av. burst length*			
	1	*2*	*3*	*4*
5 %	97.47	95.72	94.78	94.57
10 %	96.02	92.62	91.30	90.78
15 %	93.76	89.72	87.55	86.38
20 %	90.44	86.12	83.82	82.35

Tabla A.4: WAcc(%) results obtained with AMR 4.75 kbps.

Packet Loss Rate	*Av. burst length*			
	1	*2*	*3*	*4*
5 %	97.74	96.07	95.08	95.02
10 %	96.47	93.25	91.63	91.02
15 %	94.64	90.08	88.08	86.83
20 %	91.61	87.13	84.45	82.77

Tabla A.5: WAcc(%) results obtained with AMR 7.95 kbps.

Packet Loss Rate	*Av. burst length*			
	1	*2*	*3*	*4*
5 %	97.93	96.46	95.22	94.94
10 %	96.59	93.97	91.95	91.14
15 %	94.51	91.21	88.55	87.07
20 %	91.49	87.87	85.07	83.07

Tabla A.6: WAcc(%) results obtained with AMR 12.2 kbps.

Packet Loss Rate	*Av. burst length*			
	1	*2*	*3*	*4*
5 %	98.02	94.46	93.64	93.06
10 %	96.83	89.87	88.55	87.41
15 %	95.15	85.76	83.13	81.28
20 %	93.20	80.80	77.71	75.98

Tabla A.7: WAcc(%) results obtained with G.729 (8 kpbs).

Packet Loss Rate	*Av. burst length*			
	1	*2*	*3*	*4*
5 %	98.46	97.54	96.57	95.75
10 %	98.03	95.90	94.49	92.73
15 %	97.47	94.48	91.79	89.26
20 %	96.71	93.15	89.57	86.36

Tabla A.8: WAcc(%) results obtained with iLBC (15.2 kpbs).

Packet Loss Rate	*Av. burst length*			
	1	*2*	*3*	*4*
5 %	99.04	98.86	98.35	97.84
10 %	98.99	98.65	97.74	96.64
15 %	98.96	98.43	97.10	95.21
20 %	98.96	98.08	96.45	94.10

Tabla A.9: WAcc(%) results obtained with DSR FE (4.8 kpbs).

transmitting LSP coefficients by means of a differential predictive quantifier. However, this strategy makes the codec more vulnerable to continuous packet losses, since once a packet loss is finished, the LSP prediction will go on being significantly degraded. This justifies that even AMR (4.75 kbps) achieves better results than G.729 (8 kbps) for non ideal conditions. iLBC resolves these problems by removing all types of inter-frame dependencies [112] in the generation of the excitation and in the encoding of LSP coefficients. However, the price to pay is a considerable increase of bit-rate.

In general, the performance of NSR is specially lower than that of DSR when packet losses happen in bursts. The packet loss concealment algorithms implemented in the decoders are unsuitable for recognition tasks. Such algorithms are based on perceptual considerations that are not appropriate for recognition. Thus, when several consecutive packets are lost, decoders progressively mute, leading to an increase on the insertion errors in the recogniser (artificial silences).

A.3. Packet Loss Robust Speech Coding

In order to combat the reduction of speech recognition performance caused by packet losses, we propose a set of mechanisms which can be grouped into sender-driven and receiver-based techniques. In particular, we briefly summarize the sender-driven techniques proposed in this dissertation.

Nevertheless, sender-driven loss concealment techniques for NSR are only justifiable if they optimise both perceptual quality and speech recognition accuracy. The main advantage of the NSR architecture is that no terminal changes are required, so modifications can not be justified only by recognition improvements. Thus, in this section we have considered two different tests to evaluate the performance of our technique when it is applied to AMR 12.2. On one hand, an objective test is performed by means of the ITU Perceptual Evaluation of Speech Quality standard (PESQ) [135]. PESQ is able to predict subjective quality with good correlation in a very wide range of conditions, that may include coding distortions, noise, filtering, delay and variable delay. Since PESQ scores can be easily obtained by a computer, results provided by this algorithm are extensively presented along this section. However PESQ tool has an important limitation, since it does not evaluate properly the degradation caused by long bursts of lost packets [140]. For this reason, we will use a Bernoulli model (random losses) in order to simulate 8 different frame erasure ratios of 4 %, 7 %, 10 %, 13 %, 16 %, 18 %, 21 % and 23 %. On the other hand, we will verify that the proposed techniques achieve to improve the recognition accuracy using the experimental framework that we presented in the previous section.

A.3.1. Packet Loss Effects on Speech Codecs

The packet loss problem is agravated by the fact of most speech codecs are based on the CELP paradigm. CELP coding provides a high-quality speech synthesis with low bitrates, although they present a lower performance in case of packet losses due to interframe dependencies.

CELP-based codecs are based on the linear prediction model, where speech is obtained by filtering an excitation signal, $e(n)$, through a linear prediction (LP) filter, $1/A(z)$. The excitation is usually represented by the sum of two signals, namely, the adaptive vector, $e_a(n)$, and the code vector, $e_c(n)$, both weighted by their corresponding gains, g_a and g_c, that is,

$$e(n) = g_a e_a(n) + g_c e_c(n). \tag{A.2}$$

Code and adaptive vectors are chosen from a fixed and an adaptive codebook, respectively. The adaptive codebook (ACB) tries to model the long-term correlations in the excitation signal, which are related with the glottal pulses and the pitch period in voiced sounds. Thus, the entries of this codebook are dynamically determined from the previous

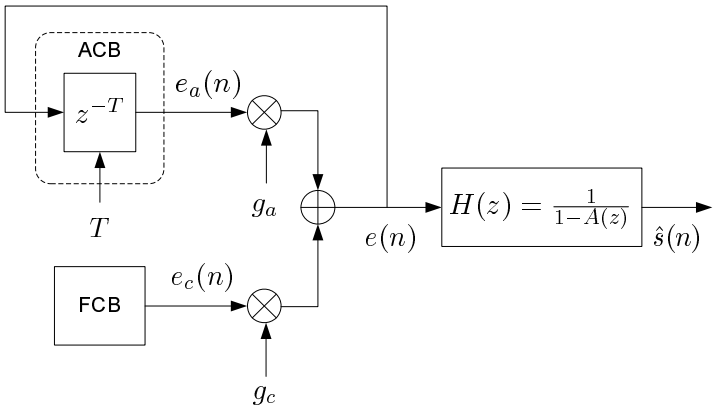


Figura A.2: Synthesis of the excitation signal in a CELP codec.

excitation by a long-term prediction (LTP) filter as,

$$e_a(n) = \sum_{k=-(q-1)/2}^{(q+1)/2} p_k e(n-(T+k)) \tag{A.3}$$

where T is the lag delay and p_k are a fixed set of prediction coefficients. On the contrary, the goal of the fixed codebook is to represent the residual signal remaining after removing the long-term redundancy. This codebook is also known as fixed codebook (FCB) since, in contrast to the adaptive one, it is not based on the previous signal. Diagram in figure A.2 depicts the processing involved in the computation of the excitation signal. Optimal CELP parameters are found by an analysis-by-synthesis procedure in which, the fixed codebook index, gains g_c and g_a, and lag delay, T, are chosen in order to minimize the error between the original speech signal and the one synthesized from the described model. To take into account the auditory masking effect, the error is usually weighted by a perceptual filter $W(z)$.

Although CELP coding is performed in a frame or subframe basis, there exist inter-frame dependencies during the decoding process which endangers the performance in packet networks. When a frame erasure happens, a concealment algorithm tries to minimize the degradation on perceptual quality by extrapolating and gradually muting the speech signal for that frame. The excitation corresponding to this concealed frame is later used by the LTP filter (ACB codebook) to compute the excitation in the next frame received after the loss. Since the concealed signal is not identical to the transmitted one, the decoder desynchronizes from the encoder and a distortion appears and propagates over the subsequent received frames, until synchrony is finally retrieved.

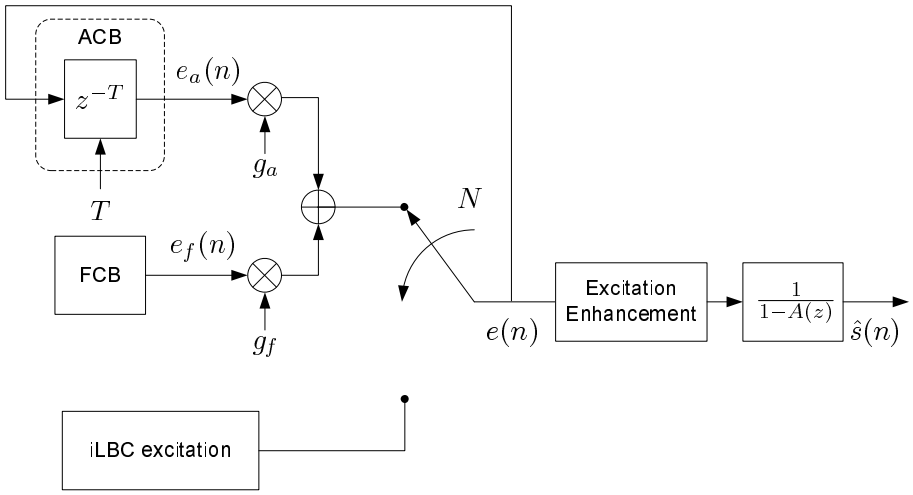


Figura A.3: Structure of the proposed decoder.

A.3.2. Coding Scheme based on Inter-Frame Dependency Limitation

iLBC is a speech codec specially conceived for packet networks, such as Internet, since it was designed to combat packet losses. To achieve this goal, iLBC does not exploit the correlation between adjacent frames in the excitation encoding. Thus, iLBC removes the interframe dependencies at the cost of a higher bit-rate than other coding techniques.

On the other hand, ACELP technique does exploit the correlation between consecutive frames to reduce the bit-rate. The main principle of this technique is called *analysis-by-synthesis* that consists of choosing an excitation signal which minimizes the error between the synthesized signal and the target signal. The excitation is produced by summing the contributions from an adaptive codebook and a fixed codebook. The fixed codebook is chosen from a number of innovation sequences, while the entries of the adaptive codebook consist of delayed versions of the excitation. Although the adaptive codebook makes possible to efficiently code quasi-periodic signals, such as voiced segments, it makes the coding scheme less robust against packet losses since, in this case, the adaptive codebook propagates the errors forward.

We propose combining both coding schemes, iLBC and ACELP, in order to obtain a robust performance against packet loss while reducing the bit-rate of iLBC. The idea is based on using iLBC and ACELP frames, as shown in Fig. A.4. Thus, in case of packet losses, the error propagation of ACELP frames is limited by the iLBC frames (key frames), which act as firewalls. Also, the insertion of ACELP frames reduces the bit-rate.

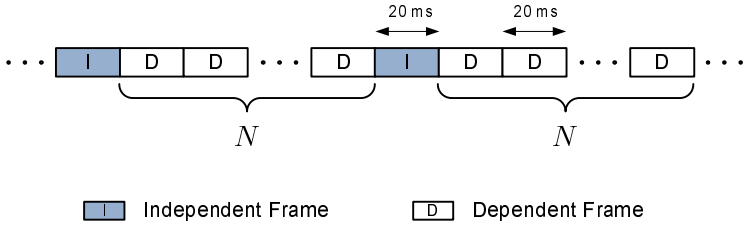


Figura A.4: Combination of different types of frames.

The reduction of the total bit-rate is controlled by the number of ACELP frames (N) inserted between two adjacent iLBC frames. Thus, a trade-off between robustness against packet loss and the bit-rate is achieved. As the distance between iLBC frames is increased, the robustness against packet losses decreases, since a larger separation of iLBC frames allows a longer propagation of errors. Regarding the delay, our proposal does not increase it, since the size of all frames is 20 ms and it is not necessary a lookahead in the encoding process. Thus, the bit-rates of our proposal is determined by the following expression,

$$B_N = \frac{B_i + N \cdot B_a}{N + 1} \tag{A.4}$$

where B_i is the bit-rate corresponding to independent frames and B_a is the bit-rates of dependent frames. Since independent frames are iLBC frames, then $B_i = 15{,}2$ kbps.

Fig. A.3 shows the structure of the decoder. It can be seen that the iLBC and CELP sections share the same LP filter. Furthermore, both types of frames use the enhancement and packet loss concealment blocks defined in [83] that works on the excitation signal and introduces an additional delay of 5 ms. The total delay of our proposal is 25 ms, just like the 20 ms mode of iLBC.

Linear Predictive Filter

The LP analysis coincides with that of iLBC for all frames. The LP coefficients are calculated once per frame using an asymmetric of 30 ms window centered in the third subframe of 5 ms. In order to quantize and interpolate the coefficients of LP filter (a_k), these ones are transformed into LSF (Line Spectrum Frequencies) parameters in the coding process. However, before being transformed, the LP coefficients are modified as,

$$\tilde{a}_k = \gamma_1^k \cdot a_k \qquad k = 1, \ldots, p$$

where $\gamma_1 = 0{,}9025$ and $p = 10$. This operation, called bandwith expansion, introduces

Parameter	*1st Subf*	*2nd Subf*	*3rd Subf*	*4th Subf*	*Total*
LSF set					20
Pitch delay	8	5	8	5	26
Algebraic code	31	31	31	31	124
Gains	8	8	8	8	32
Total					202

Tabla A.10: Bit allocation of the ACELP coding algorithm for 20 ms frames.

a distortion in the LPC spectrum, although it improves the stability of the filter. Furthermore, this operation reduces the quantization space. However, as disadvantage, some short-term redundancies remains in the residual signal.

Another advantage of the bandwidth expansion technique is the shortening of the length of the impulse response, which improves the robustness against channel errors. This is because the excitation signal distorted by channel errors is filtered by the synthesis filter, and a shorter impulse response reduces the propagation of the channel error effects to a shorter duration.

CELP Frames

A weighting filter $W(z) = 1/A(z/\gamma_2)$ is used in the analysis-by-synthesis process in order to obtain the target signal. During this process, the entries of an algebraic codebook , $e_c(n)$, based on the AMR 10.2 kbps codebook, and the adaptive codebook, $e_a(n)$, are determined, obtaining the final excitation signal through the following expression,

$$\hat{e}(n) = \hat{g}_a e_a(n) + \hat{g}_c e_c(n) \tag{A.5}$$

where $\hat{g}_a$ and $\hat{g}_c$ are the decoded pitch and code gains, respectively.

To reduce the impact of packet losses in ACELP frames, all kind of predictive techniques are avoided in the coding process. Thus, both gains are quantized using a Vector Quantizer (VQ) codebook of 8 bits, that obtains that pair ($\hat{g}_a$, $\log(\hat{g}_c)$) which minimizes the error between the synthesized speech and the target vector. This process is carried out four times per frame corresponding to 4 subframes of 5 ms. Table A.10 shows the number of bits used for coding ACELP frames. This ACELP codec presents a bit-rate of $B_a = 10{,}1$ kbps.

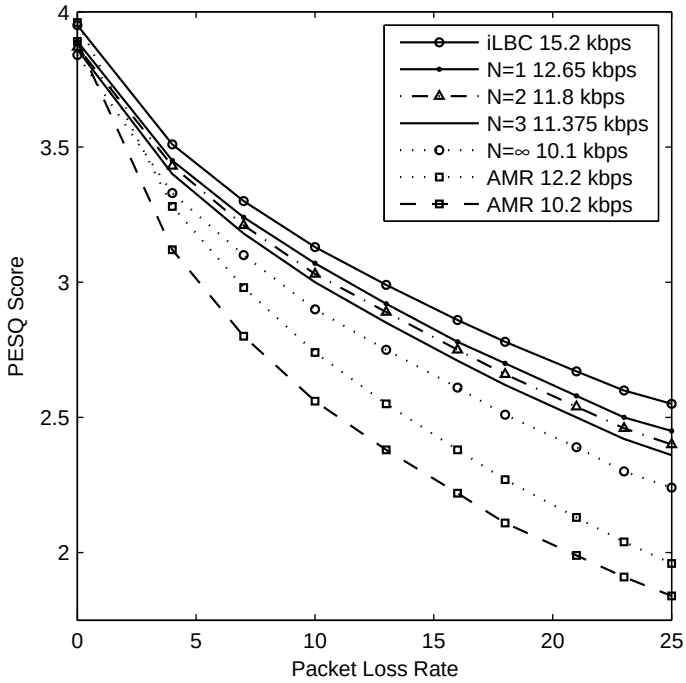


Figura A.5: PESQ results for different combinations of frames ($N = 0, 1, 2, 3, 4, \infty$). The results obtained for AMR 10.2 kbps and 12.2 kbps are also shown as references.

Experimental Results

Figure A.5 shows the PESQ performances obtained under different random packet loss rates. As shown, our proposal has been evaluated using diverse values for the number of inserted ACELP-coded frames (N). The best performance is obtained for iLBC, which corresponds to the trivial case of $N = 0$ in our proposal, i.e. inserting zero ACELP frames between two adjacent iLBC frames. The case $N = \infty$ obtains the worst result of our proposal, and it corresponds to the proposed ACELP codec with a bit-rate of 10.1 kbps (no iLBC frames). These cases limit the performance of our proposal. Four values of N have been selected with bit-rates in the range of 12.65 kbps to 11.12 kbps. Furthermore, the AMR modes of 12.2 and 10.2 kbps have been included in this figure because they present similar bit-rates and delay to some configurations of our proposal, being adequate for a comparison [142].

Particularly, the configurations with $N = 1$ (12.65 kbps) and $N = 2$ (11.8 kbps) have bit-rates close to that of AMR 12.2 kbps. Without packet loss, AMR 12.2 kbps presents a better performance (PESQ score of 3.96) than our proposal. Otherwise, the performance of AMR 12.2 is worse than any configuration of our proposal.

N	*0*	*1*	*2*	*3*	*4*	∞
Bit-rate(kbps)	15.2	12.65	11.8	11.375	11.12	10.1
PESQ score	3.94	3.89	3.87	3.86	3.86	3.84

Tabla A.11: PESQ scores for our proposal without packet losses.

Even for the case of $N = \infty$, the robustness against packet losses is higher in our proposal than in the AMR modes. Although both codecs, $N = \infty$ and AMR 10.2 kbps, use the same ACELP architecture, AMR uses predictive techniques to quantize more efficiently the codec parameters (e.g. the excitation gains are quantized using a predictor filter). This justifies that AMR 10.2 achieves better results when there is not packet loss (PESQ score of 3.89) than our proposal for $N = \infty$ (PESQ score of 3.84). However, in presence of packet losses, these predictive techniques are no suitable. This, along with the use of bandwidth-expanded LP coefficients, is the reason why our proposed codec is more robust against packet losses.

Although our proposal for $N \geq 1$ presents lower bit-rates than iLBC, the behavior against packet loss is close to iLBC. As more CELP frames are inserted between consecutive iLBC frames the robustness goes down. Nevertheless, our work provides an easy method to make the iLBC codec scalable with a small PESQ performance degradation in absence of packet loss, as shown in Table A.11. Furthermore, in comparison with a scalable CELP coding scheme such as AMR, the performance of our proposal is clearly higher. More degree of scalability could be reached using an ACELP codec with lower bit-rate (instead of 10.1 kbps) in combination with iLBC.

A.3.3. Multipulse FEC codes for CELP codecs

A naive approach to solve this problem could be that of sending the complete ACB codebook to the receiver. That is, we would transmit a FEC code consisting of the previous excitation samples for every frame. Of course, this completely alienates the CELP coding idea, increasing the bitrate up to unusable limits. Instead, we propose to encode only the most representative excitation samples by means of a multipulse scheme.

In multipulse coding, the LP excitation is represented by a few pulses with different amplitude and position. The same approach could be used to encode the excitation of the previous frame with a few bits. This side information could then be transmitted as a FEC code along with the CELP parameters. Thus, after a frame loss, a multipulse version of the previous frame excitation would be available, allowing a partial ACB reconstruction.

Since this direct scheme will be useful to develop the rest of our techniques, we will briefly recall the multipulse approach in the following. As mentioned before, multipulse coding approaches the excitation as a sum of a L pulses at different time instants n_l and amplitudes b_l,

$$\hat{e}(n) \equiv \sum_{l=0}^{L-1} b_l \delta(n - n_l) \tag{A.6}$$

Pulse positions and amplitudes are chosen in order to obtain the least square error (LSE criterion) between the target signal and the synthesized one. The square error is defined by,

$$E = \sum_{n=0}^{N-1} (s(n) - \hat{s}(n))^2 = \sum_{n=0}^{N-1} (s(n) - h(k) * \hat{e}(n-k))^2 \tag{A.7}$$

where $s(n)$ is the original speech signal, $\hat{s}(n)$ is the synthesized one, $h(n)$ the impulse response of the LP filter and $\hat{e}(n)$ the coded excitation. To take into account the properties of human auditory perception, the error signal is commonly weighted by a perceptual filter $w(n)$, so that,

$$\begin{aligned} E_w &= \sum_{n=0}^{N-1} (w(n) * (s(n) - \hat{s}(n)))^2 && \text{(A.8)} \\ &= \sum_{n=0}^{N-1} (w(n) * s(n) - w(n) * h(k) * \hat{e}(n-k))^2 && \text{(A.9)} \\ &= \sum_{n=0}^{N-1} (s_w(n) - h_w(k) * \hat{e}(n-k))^2 && \text{(A.10)} \end{aligned}$$

which remains identical to equation (A.7) except for the LP impulse response $h_w(n)$, and the target signal $s_w(n)$, which are now convolved with the perceptual filter. It must be noted that these signals are already available during CELP coding since the same LSE criterion is also applied.

By replacing equation (A.6) in (A.8), the following well-known expression is obtained,

$$E_w = \sum_{n=0}^{N-1} (s_w(n) - \sum_{l=0}^{L-1} b_l h_w(n - n_l))^2 \tag{A.11}$$

where, assuming that pulse positions are known, optimal LSE amplitudes b_l can be computed as,

$$\mathbf{b}^* = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{s} \tag{A.12}$$

where $\mathbf{b} = [b_0, b_2, \ldots, b_{L-1}]^T$, $\mathbf{s} = [s_w(0), s_w(1), \ldots, s_w(N-1)]^T$ and $\mathbf{H}$ is a $N \times L$ matrix given by,

$$\mathbf{H} = \begin{bmatrix} h_w(0-n_0) & h_w(0-n_1) & \ldots & h_w(0-n_{L-1}) \\ h_w(1-n_0) & h_w(1-n_1) & \ldots & h_w(1-n_{L-1}) \\ \vdots & \vdots & \ddots & \vdots \\ h_w(N-1-n_0) & h_w(N-1-n_1) & \ldots & h_w(N-1-n_{L-1}) \end{bmatrix} \tag{A.13}$$

A more common notation is given by reordering equation (A.12) as,

$$(\mathbf{H}^T\mathbf{H})\mathbf{b}^* = \mathbf{H}^T\mathbf{s} \tag{A.14}$$

$$\mathbf{\Phi}\mathbf{b}^* = \mathbf{c} \tag{A.15}$$

which provides the following simultaneous equations,

$$\sum_{k=0}^{L-1} b_k \phi_{n_k,n_j} = c_{n_j}, \qquad 0 \leq j \leq L-1 \tag{A.16}$$

where,

$$\phi_{n_k,n_j} = \mathbf{\Phi}[k,j] = \sum_{n=0}^{N-1} h_w(n-n_k)h_w(n-n_j), \quad c_{n_j} = \mathbf{c}[j] = \sum_{n=0}^{N-1} s_w(n)h_w(n-n_j) \tag{A.17}$$

so that, ϕ_{n_k,n_j} is often approached by the auto-correlation of the LP impulse response, $R_{hh}(n_k - n_j)$, and c_{n_j} by the cross-correlation between the LP impulse response and the target signal.

The predicted LSE error for a determined combination of pulse positions with optimum amplitudes can be obtained as,

$$E_w^* = \mathbf{s}^T[\mathbf{I} - \mathbf{H}(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T]\mathbf{s} \tag{A.18}$$

Thus, the optimal computation of pulses and amplitudes could be achieved by testing all the available combinations of L pulses in N positions and by choosing that providing the lowest error. However, this is computationally impracticable as N and L increase, since there exist $N!/l!(N-l)!$ possible combinations. Due to this, a suboptimal algorithm, as that proposed in [143], is generally followed in practice. In each step, this algorithm searches for the optimal position and amplitude of only one pulse. In the next iteration, this pulse is considered as fixed and its contribution removed from the target signal. A

usual improvement is to recompute the optimal amplitudes for all the pulses in the last step. Since ϕ_{n_k,n_j} and c_{n_j} are intensively used and, eventually, all positions n_k, n_j are evaluated, $\phi_{k,j}$ and c_j are precomputed and stored for every possible position ($0 \leq k, j \leq N-1$).

LTP Multipulse Approach for Propagation Error Recovery

The direct multipulse approach optimizes pulse parameters considering the previous frame signal. The current frame signal and its CELP parameters are neglected during this procedure. However, these could be relevant for ACB resynchronization and potentially reduce the side information to be transmitted.

In this section, we propose a modification of the multipulse technique whereby pulse optimization is performed taking into account the parameters of the current frame. As before, a multipulse description of the previous frame excitation is sent as a FEC code in every CELP frame. After a frame erasure, this alternative excitation is used instead of that provided by the PLC algorithm in order to prevent the ACB de-synchronization. However, this multipulse description is optimized now to minimize the perceptual error with respect to the current frame. In order to do so, the previous frame samples are seen as a memory where some pulses can be set. These pulses are later transformed by the LTP filter, scaled by the adaptive gain and added to the code vector (also scaled by the code gain) producing the excitation of the current frame. The speech signal is finally obtained as the LP filter response to this excitation. As expected, depending on the position and amplitude of the pulses, different response signals can be obtained. Then, our goal now is the search of the optimum pulse parameters in a LSE sense.

Removing the known fixed contribution To simplify the problem, we first remove the code contribution to the excitation from the target signal. To this end, we redefine the CELP excitation as the sum of two signals, namely, the zero state and the zero input excitation. The zero state excitation is computed by considering that the samples before the current frame are zero (i.e. no pulses on the memory). On the other hand, the zero input excitation is obtained by considering that the code vector is zero for the current frame. The idea is that the excitation can be seen as the resulting signal of a filter $P(z)$ applied over code vector $e_c(n)$ (see figure A.2). Therefore, through the superposition principle, excitation signal can be obtained as the sum of the zero state and zero input responses from that filter.

Assuming that all the parameters have been received for the current frame, the zero state excitation can be perfectly reconstructed as it does not require any previous samples. After being LP filtered, its contribution can be removed from the target signal. Then, the square error to be minimized can be expressed in terms of the zero input excitation as,

$$\begin{aligned} E_w &= \sum_{n=0}^{N-1}(s_w(n) - h_w(k) * \hat{e}(n-k))^2 \\ &= \sum_{n=0}^{N-1}(s_w(n) - h_w(k) * (e_{zs}(n-k) + \hat{e}_{zi}(n-k))^2 \qquad \text{(A.19)} \\ &= \sum_{n=0}^{N-1}(s_w(n) - s_{zs}(n) - h_w(k) * \hat{e}_{zi}(n-k))^2 \qquad \text{(A.20)} \end{aligned}$$

where $e_{zs}(n)$ is the zero state excitation, $s_{zs}(n)$ is its LP response and $\hat{e}_{zi}(n)$ is the zero input excitation that we have to obtain. As can be seen, the problem is now simplified since the code vectors are not needed to compute the zero input excitation and, therefore, they are not required by the optimization procedure.

Pulse-based definition of the excitation After removing the zero state excitation from the target signal, the frame excitation is given by a recursion of only adaptive vectors which eventually depends on some initial pulses. In this section we focus on the relationship between these initial pulses and the final structure of the excitation.

In order to do so, we will initially assume that the LTP filter is just a delay filter. In such a way, we can consider that pulses are successively replicated in each subframe so that the excitation signal after LTP filtering can be seen as a sequence of individual pulses. Figure A.6 shows an example in which three initial pulses at time instants n_p ={-55,-45,-10} with amplitudes b_p ={1, 0.5, 1.25}, respectively, are LTP filtered. Four subframes are considered, whose LTP lags are T ={60, 50, 25, 42} and adaptive gains g_a ={1, 2, 0.5, 1.25}. As can be seen, given a pulse, this is replicated with different amplitudes several times in the frame. While these positions depend exclusively on subframe lags, their amplitudes depend on both the initial pulse amplitudes and the adaptive gains accumulated along the subframes. In the following, we will refer to these accumulated gains as weights. This allows us, applying the superposition principle, to represent the frame (zero input) excitation as a combination of replicas of every initial pulse p ($p = 0, 1, \ldots, P-1$) as

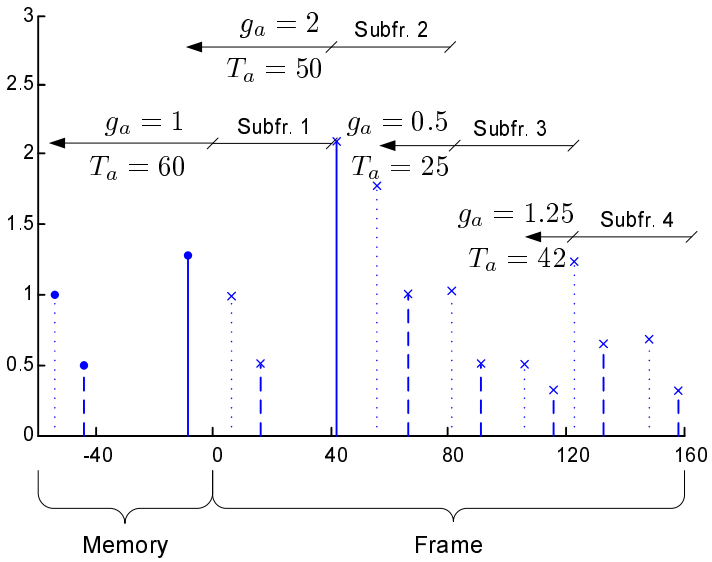


Figura A.6: Zero input excitation pulses (x) for a set of initial pulses (o) obtained by simplifying the LTP filter in each subframe as a delay filter.

follows,

$$\hat{e}_{zi}(n) \equiv \sum_{p=0}^{P-1} b_p \sum_{j=0}^{L-1} w(n_{p,j})\delta(n - n_{p,j}) \tag{A.21}$$

where b_p are the initial pulse amplitudes, and $n_{p,j}$ and $w(n_{p,j})$ are, respectively, the position and the weight for replica j of a pulse p. It must be noted that each pulse p can generate an arbitrary number of pulse replicas, L_p, as figure A.6 depicts. However, in order to simplify equation (A.21) we can take its maximum value, that is, $L = \max L_p$, $0 \leq p < P$, and set the weights of the excess replicas of every pulse to 0.

Provided the previous definition for the excitation, the synthesized zero-input signal can be obtained as,

$$\begin{aligned} \hat{s}_{zi}(n) &= h_w(n) * \hat{e}_{zi}(n) = \textstyle\sum_{k=0}^{N-1} h_w(k) \sum_{p=0}^{P-1} b_p \sum_{j=0}^{L-1} w(n_{p,j})\delta(n - n_{p,j} - k) \\ &= \textstyle\sum_{p=0}^{P-1} b_p \sum_{j=0}^{L-1} w(n_{p,j}) \sum_{k=0}^{N-1} h_w(k)\delta(n - n_{p,j} - k) = \\ &= \textstyle\sum_{p=0}^{P-1} b_p \sum_{j=0}^{L-1} w(n_{p,j}) h_w(n - n_{p,j}) \end{aligned} \tag{A.22}$$

Therefore, the square error of equation (A.20) can be expressed as,

$$E_w = \sum_{n=0}^{N-1} (s_{zi}(n) - \sum_{l=0}^{P-1} b_p g_p(n - n_p))^2 \tag{A.23}$$

where,

$$s_{zi} = s_w(n) - s_{zs}(n) \tag{A.24}$$

$$g_p(n - n_p) = \sum_{j=0}^{L-1} w(n_{p,j}) h_w(n - n_{p,j}) \tag{A.25}$$

As can be seen, equation (A.23) is essentially identical to equation (A.11). By following equations (A.12) to (A.16), optimal amplitudes b_p could be obtained provided the initial pulse positions n_p (positions of replicas $n_{p,j}$ are derived from them). Then, the same algorithm used in the multipulse scheme could be used to provide a quasi-optimal solution for pulse parameters. However, this approach have several limitations:

- In modern CELP codecs, the LTP filter is more complex than a delay filter. As an example, AMR uses a fractional pitch filter which includes several prediction coefficients to calculate the adaptive vector. When a simple delay filter tries to approximate these filters, the fine details of the adaptive vector are lost.
- The LP filter response, $h(n)$, is assumed to be fixed during the whole frame. In contrast to direct multipulse coding, minimization can not be performed in a subframe basis since the parameters of the initial pulses affect all the subframes at the same time. In this case, individual subframe LP response can only be approximated by a common LP set obtained as an average (in the LSF domain) over all the subframes.

To cope with these limitations, we propose a different approach where the (zero-input) excitation is based on complete signals obtained from initial pulses. This is developed in the following subsection.

Shape-based definition of the excitation Instead of individual replicas, we can consider now that each pulse p causes a shape signal, $w_p(n)$, defined for the whole frame ($0 \leq n \leq N-1$). The pulse-based excitation of the previous subsection is then a particular case in which the shape signal is defined as (see equation (A.21)),

$$w_p(n) = \sum_{j=0}^{L-1} w(n_{p,j}) \delta(n - n_{p,j}) \tag{A.26}$$

However, shape signal is no longer restricted to a set of pulses. Now, it can be as complex as required, representing more sophisticated responses. Figure A.7 shows the

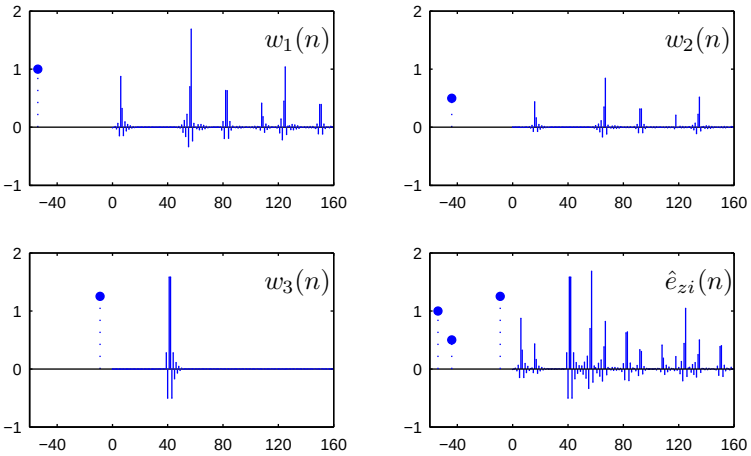


Figura A.7: Zero input excitation $\hat{e}_{zi}$ obtained from a set of three initial pulses as the sum of their corresponding shape signal $(w_1(n), w_2(n), w_3(n))$ with a fractional LTP filter per subframe.

shape signals corresponding to the initial pulses considered in the example of figure A.6, but when a fractional LTP filter is considered. Shape signals are simply calculated by LTP filtering each individual pulse along the whole frame. As can be observed, the final excitation is obtained as a sum of these individual signals.

In general, the excitation signal is now defined in terms of shape signals as,

$$\hat{e}_{zi}(n) \equiv \sum_{p=0}^{P-1} b_p w_p(n) \tag{A.27}$$

Then, the synthesized signal is given by,

$$\begin{aligned} \hat{s}_{zi}(n) &= h_w(n) * \hat{e}_{zi}(n) = \textstyle\sum_{k=0}^{N-1} h_w(k) \sum_{p=0}^{P-1} b_p w_p(n-k) \\ &= \textstyle\sum_{p=0}^{P-1} b_p \sum_{k=0}^{N-1} h_w(k) w_p(n-k) \\ &= \textstyle\sum_{p=0}^{P-1} b_p g_p(n) \end{aligned} \tag{A.28}$$

where,

$$g_p(n) = \sum_{k=0}^{N-1} h_w(k) w_p(n-k) = h_w(n) * w_p(n) \tag{A.29}$$

that is, $g_p(n)$ is the LP response to the shape signal corresponding to pulse p.

Thus, the square error between the synthesized signal and the original one can be

expressed as,

$$E_w = \sum_{n=0}^{N-1}(s_{zi}(n) - \sum_{l=0}^{P-1} b_p g_p(n))^2 \tag{A.30}$$

Assuming that the initial pulse positions are known, optimal amplitudes b_p can be obtained through LSE equations defined in (A.11) and (A.12) by simply replacing matrix $\mathbf{H}$ by matrix $\mathbf{G}$, defined as,

$$\mathbf{G} = \begin{bmatrix} g_0(0) & g_1(0) & \dots & g_{P-1}(0) \\ g_0(1) & g_1(1) & \dots & g_{P-1}(1) \\ \vdots & \vdots & \ddots & \vdots \\ g_0(N-1) & g_1(N-1) & \dots & g_{P-1}(N-1) \end{bmatrix} \tag{A.31}$$

so that,

$$\mathbf{b}^* = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{s} \tag{A.32}$$

In addition, the suboptimal algorithm used to jointly compute pulse amplitudes and positions described in Section A.3.3 can be also applied by redefining $\mathbf{\Phi}$ and $\mathbf{c}$ as,

$$\mathbf{\Phi} = \mathbf{G}^T\mathbf{G} \qquad\qquad \mathbf{c} = \mathbf{G}^T\mathbf{s} \tag{A.33}$$

Along with the ability to represent arbitrary LTP filters, the proposed scheme introduces additional advantages. Thus, the shape-based definition for zero-input excitation not only allows a straightforward implementation but can also cope with different LP sets for each subframe, since LP response to the shape signal, $g_p(n)$, can be computed in a subframe basis.

Application and Evaluation of LTP Multipulse Scheme in Practice

In order to render the proposed scheme usable, the amplitude and the position of memory synchronization pulses must be represented with a finite precision. As expected, the amount of side information must be kept as small as possible but, on the other hand, a strong quantization of pulse parameters may be detrimental for the performance scheme. In this section we focus on coding just one initial pulse, as it provides the highest relative improvement in comparison with using two or more pulses. In particular, AMR 12.2 is considered for testing purposes, although encoding is easily extensible to other CELP codecs. On the other hand, the proposed quantization scheme could be reused to encode additional pulses, but possible relations between them will not be exploited.

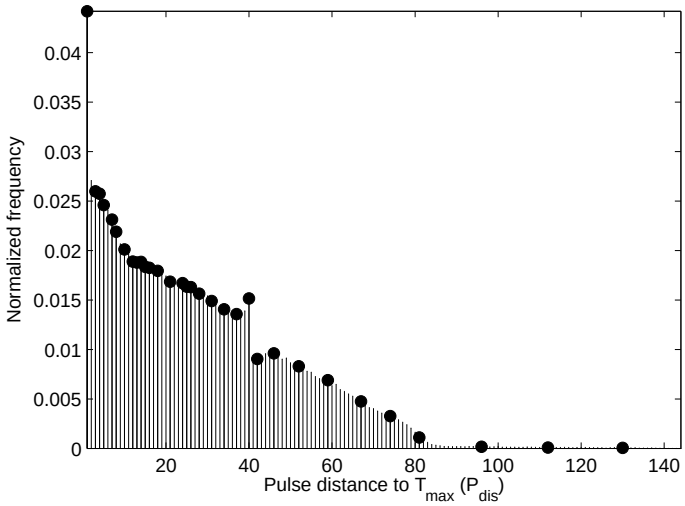


Figura A.8: Distance distribution on TIMIT training database of pulses used to avoid error propagation in AMR 12.2 codec. Marked distances represents those selected as available by a Lloyd-Max quantizer with 32 centers.

After error minimization, the pulse amplitude can be encoded with a non-uniform quantizer. Quantization intervals can be obtained by applying the Lloyd-Max algorithm to a training set (in our case, that from TIMIT database).

In contrast to pulse amplitude, optimal pulse position cannot be computed and then quantized, as error turns out critical in this case. Instead, we must limit, prior to the error minimization process, the number of available positions where a pulse can be set.

Initially, we can represent the pulse position as an absolute integer value with N bits. Then, we could limit the pulse optimization to only the $P_{pos} = 2^N$ positions previous to the frame. In this case, the most frequently used samples are not farther than 128 samples so that $N = 7$ bits would be enough to encode the pulse position.

This coding can be refined if, instead of using an absolute value, we consider a relative value referred to the frame lag. For each frame we can obtain a maximum lag value, T_{max}, as follows,

$$T_{max} = \max[T_0, T_1 - N_s, T_2 - 2 \cdot N_s, \ldots, T_i - i \cdot N_s, \ldots, T_(M-1) - (M-1) \cdot N_s] \quad \text{(A.34)}$$

where T_i is the integer lag of subframe i, M is the number of subframes per frame, and N_s is the subframe length. Since the samples before the maximum lag are not used in the LTP filtering, the synchronization pulse is assured to be after T_{max}. Therefore, its position can be encoded as the number of samples after T_{max}, that is, $P_{dis} = P_{pos} - T_{max}$.

This approach can also be easily implemented by considering only the 2^N positions after T_{max} (and before the frame) during error minimization. Figure A.8 shows the distribution of pulse distances to T_{max}. Memory synchronization pulses have been obtained for the AMR 12.2 codec using the training set of the TIMIT database. As can be observed, a significant number of pulse positions coincides with the maximum lag whilst most of them are not farther than 64 samples ($N = 6$) from maximum lag.

The previous encoding has the disadvantage that a group of consecutive pulse positions is neglected (those $P_{dis} > 2^N$). Since these positions are not frequent and generally do not correspond to voiced sounds, this does not seem a critical issue. However, a further improvement can be obtained by encoding P_{dis} with a non-uniform quantizer. The idea is to have a high density of available positions for the most frequent pulse distances, and a few ones for those less frequent. In order to constraint the pulse optimization, the shape signals corresponding to non available positions can be set to 0. In such a way, it is assured these positions will not be considered during error minimization. Again, the Lloyd-Max algorithm can be used to obtain a non-uniform quantizer. Figure A.8 also shows the selected 32 P_{dis} values (i.e. available positions) for the memory synchronization pulses obtained by this algorithm.

Finally, figure A.9 shows the performance of the aforementioned lag coding schemes with 16, 32 and 64 available positions (4, 5 and 6 bits) for the resynchronization pulse in AMR. Pulse amplitude is also coded with 4, 5 and 6 bits. In order to reduce the number of results, only the average PESQ score over all the adverse channels (from 4 % to 23 % of frame erasures) is presented. As a reference, mean PESQ score achieved without quantization is 2.84. In general, the performance decreases as less bits are devoted to represent the initial pulse. As can be observed, non-uniform quantization provides the best results for lag coding, in particular at low bitrates. Given these results, a pulse quantization with 6 bits for position and 5-bit amplitude could be a good encoding choice, as a negligible reduction of performance is introduced in comparison with using an unquantized pulse.

Perceptual Quality Evaluation

In addition to PESQ scores, MUSHRA methodology is applied to evaluate our proposed technique with real listeners. Three different schemes for propagation error concealment have been compared: the one included in AMR (i.e. no explicit concealment technique), a complete ACB memory restoration where all the previous samples are available for the

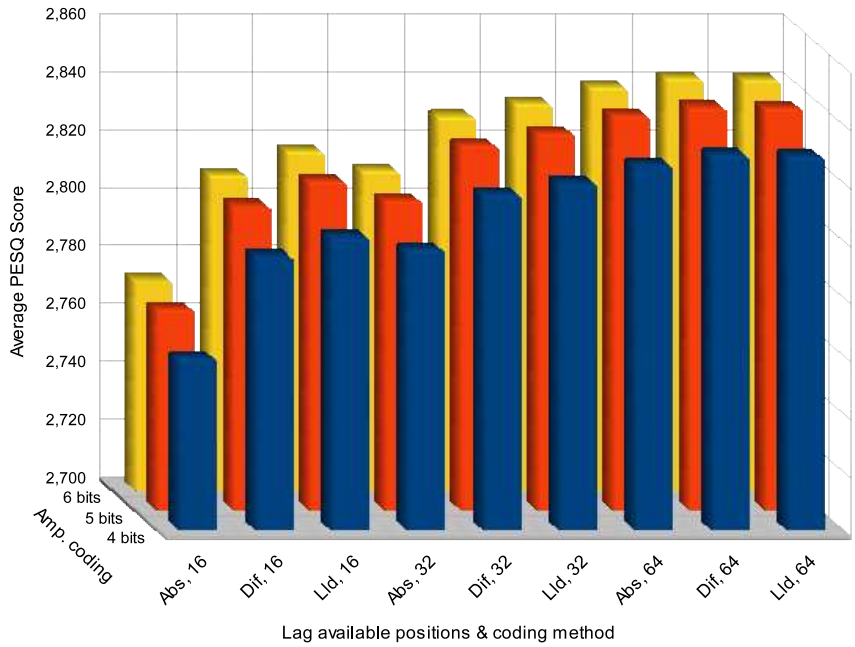


Figura A.9: Quantization effect on resynchronization pulse in terms of mean PESQ score (over adverse channels) for the AMR 12.2 codec. Pulse lag is quantized with 16, 32 and 64 available positions with absolute (Abs.), difference (Dif.), and non-uniform (Lld.) coding schemes, while 4, 5 and 6 bits are used for the amplitude.

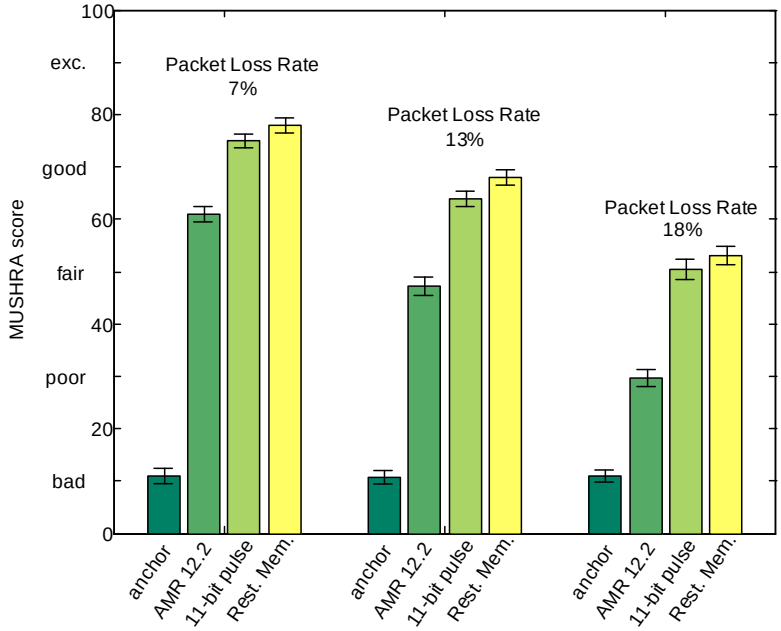


Figura A.10: MUSHRA scores achieved by AMR 12.2 standard codec, with a complete restoration of ACB memory and by using a 11-bit resynchronization pulse with 5 bits for amplitude and 6 bits for position.

LTP filter, and a shape-derived LTP multipulse scheme with only one initial pulse quantized with 11 bits (6-bit position and 5-bit amplitude). In our evaluation, ten listeners participated in the experiments. Each listener evaluated 16 different test items obtained from the phonetic Albayzin database [144]. This database was selected in order to provide the listeners sentences in their native language (Spanish). The selected utterances are phonetically balanced and uttered by female and male speakers in the same proportion. The test conditions are given by 7 %, 13 % and 18 % of packet losses. Figure A.10 summarizes the mean scores obtained through MUSHRA evaluation (confidence intervals have been set to 95 %).

As can be observed, the results predicted by PESQ are consistent with those provided by real listeners. However, perceptual quality differences between techniques become more relevant through MUSHRA evaluation. Thus, the shape-derived LTP multipulse scheme achieves a notable improvement over the standard codec. This improvement is almost the same offered by a complete ACB restoration. However, while bandwidth increase is not affordable for the last solution, our proposal only requires an increment of 0.55 kbps in the total bit-rate.

A.3.4. Speech Recognition Results

Table A.12 summarises the speech recognition results obtained for the proposals presented in this section. In particular, the baseline results are given by the AMR standard (mode 12.2 kbps). The other two methods corresponds to the frame-combination (FC) and the multipulse FEC proposals. The parameters of both methods were chosen in order to obtain a similar bit-rate and, therefore, a fair comparison. Thus, FC was carried out using only one ACELP frame between two adjacent iLBC frames ($N = 1$). On the other hand, the FEC solution is based on a multipulse representation using an only pulse encoded with 11 bits (5 bits for amplitude and 6 bits for position). The experimental framework is the presented one in Section A.2, while the packet loss conditions are given by Table A.3.

The PESQ results presented along this section show that our proposals achieve perceptual improvements. However, we have worked under the hypothesis that these improvements will be translated into recognition accuracy increments. The results showed in Table A.12 verifies this hypothesis. In particular, the objective of both proposals is to reduce the error propagation. For this reason, the relative improvements are greater when packet losses are short. On the contrary, when packet losses are long, the packet loss concealment algorithms integrated in decoders progressive mute the speech signal. In this

Speech codec	*Bit-rate (kbps)*	*Packet Loss Cond.*				
		C0	*C1*	*C2*	*C3*	*C4*
AMR	12.2 kbps	98.70	97.93	93.97	88.55	83.07
FC $N = 1$	12.65 kbps	98.65	98.54	96.04	91.66	85.90
AMR+FEC	12.75 kbps	98.70	98.46	95.90	91.79	86.36

Tabla A.12: Summary of word accuracy (WAcc(%)) results applying the sender-driven proposals: Frame Combination (FC) $N = 1$; AMR+FEC 1 pulse (12.65 kbps).

last case, although our proposals limit the error propagation, the recognition performance is determined by the muting (artificial silences), leading to an increase on the insertion errors. In the next section, we will see how to improve the recognition accuracy against long losses.

A.4. Receiver-based PLC Algorithms

As we saw in Section A.2, the use of conventional speech codecs degrades the performance of NSR systems in presence of packet losses, mainly with CELP-based codecs. In this section, we briefly summarize those receiver-based reconstruction techniques proposed in this dissertation. In first place, we study the degradations introduced by CELP codecs in lossy packet networks. Later, we propose a reconstruction technique based on MMSE (Minimum Mean Square Error) estimation using Hidden Markov Models (HMM). This approach also allows to obtain reliability measures associated to each estimate. We show how to use this information to improve the recognition performance by means of soft-data decoding and weighted Viterbi algorithm. Finally, we conclude this section with the experimental results, which are obtained for two well-known CELP codecs, G.729A and AMR 12.2 kbps.

A.4.1. Impact of Packet loss in CELP-based Codecs

Unfortunately, the inter-frame dependencies in the encoding process endanger the performance in packet networks. Once a packet loss is over, the predictive schemes used by the encoder prevent to obtain the correct decoded parameters. Furthermore, even when the decoder has already correct parameters, there exists an error propagation caused by the ACB contribution to the excitation signal.

Fig. A.11 shows an example of how a loss burst affects the ASR feature extraction. We have simulated a burst of 4 lost frames for a CELP-based codec (AMR 12.2 kbps [78]). In

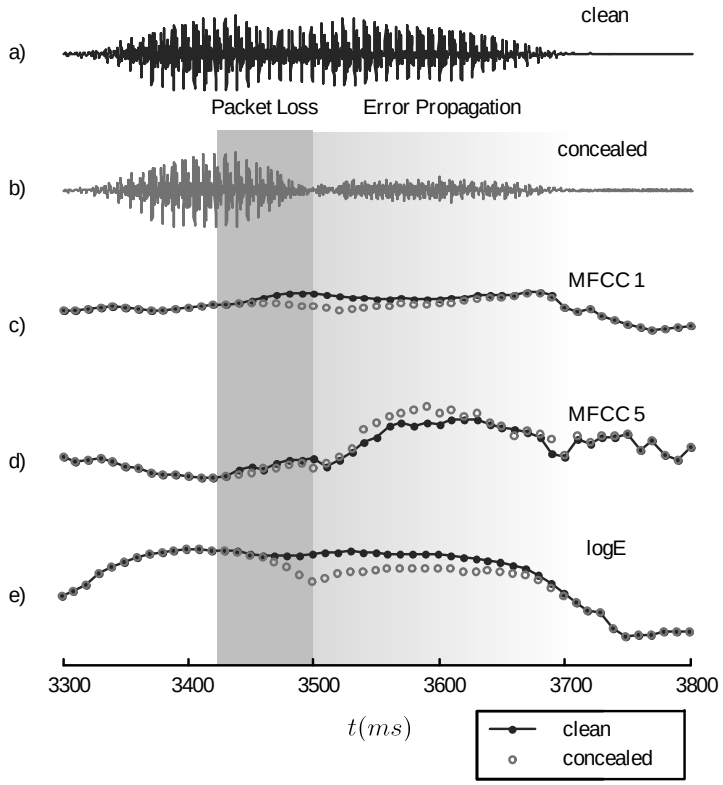


Figura A.11: Effect of a packet loss on CELP-based speech synthesis (using AMR 12.2 kbps) and ASR feature extraction. a) Speech decoded in *clean* channel condition (i.e. without packet loss); b) Speech decoded in a lossy channel condition using the packet loss concealment algorithm included by AMR 12.2 kbps; c), d) and e) Speech recognition features ($MFCC(1)$, $MFCC(5)$, and $\log E$, respectively) extracted from *clean* and *concealed* speech waveforms.

the first two plots we show the speech waveforms corresponding to a *clean* transmission (without packet loss) and the packet loss *concealed* synthesis using the corresponding PLC algorithm [192]. In the following plots, it is shown the influence of the loss burst on some ASR parameters. Two different effects can be observed. On the one hand, those parameters corresponding to the packet loss area are extracted directly from the signal generated by the PLC algorithm. These algorithms try to mitigate the errors in the synthesis using perceptual considerations (e.g. repetition and muting). Evidently, these considerations are not suitable for speech recognition. On the other hand, the ASR features after the packet loss are still distorted due to the propagated error, which can have a duration even longer than the lost segment.

To illustrate the impact of error propagation on speech recognition features, Fig. A.12 shows the relation between the variance of the *clean* log-Energy parameter, σ_x^2, and the mean squared error caused after a packet loss. Thus, this relation is given by:

$$\Gamma(t_{ep}) = 10\log_{10}\frac{\sigma_x^2}{E\left[\left(x_{t_{ep}} - y_{t_{ep}}\right)^2\right]} \tag{A.35}$$

where $x_{t_{ep}}$ and $y_{t_{ep}}$ are the uncorrupted (without packet loss) and distorted (affected by propagation error) log-Energy parameters at frame t_{ep} after a packet loss, respectively. The results were obtained using the AMR codec (mode 12.2 kbps and 1 frame per packet) by testing several burst lengths of packet loss (L_{burst}) in a training database, and averaging the distortions caused at the subsequent parameters after burst.

As can be seen, there exists a dependency between the number of consecutive lost packets and the error propagation caused by the loss. The result is consistent because the longer burst is, the more mismatch appears in the adaptive codebook. Furthermore, the propagated error is reduced as the feature vector is more distant from the end of the packet loss, as showed in the example of Figure A.11. For these reasons, the mitigation techniques proposed in this work will model the distortion caused by error propagation taking into account the length of the burst and the relative position after it.

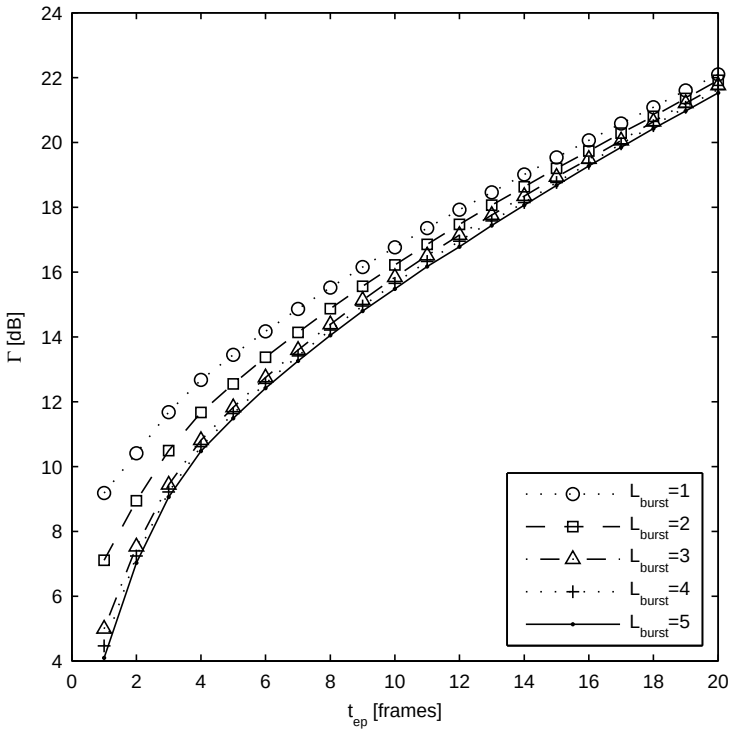


Figura A.12: Ratio between the uncorrupted log-Energy variance (without packet loss) and the mean squared error caused by packet loss bursts of different lengths (L_{burst}) at frame t_{ep} after the burst.

A.4.2. HMM-based MMSE Estimation

In order to obtain a reconstruction of the uncorrupted feature vector $\boldsymbol{x}_t$, we use the following MMSE estimation,

$$\bar{\mathbf{x}}_t = E[\mathbf{x}_t|Y] = \sum_{i=0}^{N-1} \mathbf{x}^{(i)} P(\mathbf{x}_t = \mathbf{x}^{(i)}|Y) \tag{A.36}$$

where $\{\mathbf{x}^{(i)}; i = 0, \ldots, N-1\}$ is a set of prototype vectors that represents the uncorrupted feature space ($\mathbf{x}_t$ is the result of quantizing $\boldsymbol{x}_t$ with this prototype vector set), and $Y = (\mathbf{y}_0, \ldots, \mathbf{y}_T)$ represents a quantized version of those vectors $(\boldsymbol{y}_0, \ldots, \boldsymbol{y}_T)$ affected in some way by a packet loss, where $\mathbf{y}_0$ is the observed vector before the packet loss and $\mathbf{y}_T$ is the first vector after the packet loss that is not affected by propagated error.

To carry out this estimation we model the *a priori* knowledge about the speech source through an Hidden Markov Model (HMM) [156], where each state $s^{(i)}$ $(i = 0, \ldots, N-1)$ represents a prototype $\mathbf{x}^{(i)}$. This HMM is described by means of observation probabilities

$b_i(\mathbf{y})$ and transition probabilities a_{ij}, defined as,

$$b_i(\mathbf{y}) \equiv P(\mathbf{y}|\mathbf{x}^{(i)}) \tag{A.37}$$

$$a_{ij} \equiv P\left(\mathbf{x}_t = \mathbf{x}^{(j)}|\mathbf{x}_{t-1} = \mathbf{x}^{(i)}\right) \tag{A.38}$$

As previously mentioned, every observation vector $\mathbf{y}_t$ $(t = 0, \ldots, T)$ is a quantized version of an original corrupted vector $\boldsymbol{y}_t$. This quantization is carried out with the objective of obtaining a discrete observation distribution in eq. (A.37), so our HMM model is also discrete, which leads to a simpler development than in the case of using a continuous HMM model. Although we could obtain a specific quantizer for the corrupted space, we will use for this purpose the prototype vector set $\{\mathbf{x}^{(i)};\, i = 0, \ldots, N-1\}$ for further simplicity. Of course, we assume that this prototype set provides an accurate representation of the feature space. Anyway, we will see in this section how to deal with any possible degradation introduced by this quantization.

Applying the defined model, we can now compute the conditional probabilities used in MMSE estimation (A.36) as,

$$P(\mathbf{x}_t = \mathbf{x}^{(i)}|Y) = \gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=0}^{N-1} \alpha_t(j)\beta_t(j)} \tag{A.39}$$

where,

$$\alpha_t(i) = P(\mathbf{x}_t = \mathbf{x}^{(i)}|\mathbf{y}_0, \ldots, \mathbf{y}_t) \tag{A.40}$$

$$\beta_t(i) = P(\mathbf{y}_{t+1}, \ldots, \mathbf{y}_T|\mathbf{x}_t = \mathbf{x}^{(i)}) \tag{A.41}$$

These forward and backward conditional probabilities can be obtained through the following recursions,

$$\alpha_t(i) = \left[\sum_{j=0}^{N-1} \alpha_{t-1}(j)a_{ji}\right] b_i(\mathbf{y}_t)/K_t \quad t > 0 \tag{A.42}$$

$$\beta_t(i) = \sum_{j=0}^{N-1} a_{ij}b_j(\mathbf{y}_{t+1})\beta_{t+1}(j) \qquad t \leq T \tag{A.43}$$

where K_t is a normalization factor applied at each step in the recursion. Further details about this MMSE estimation can be found in [156].

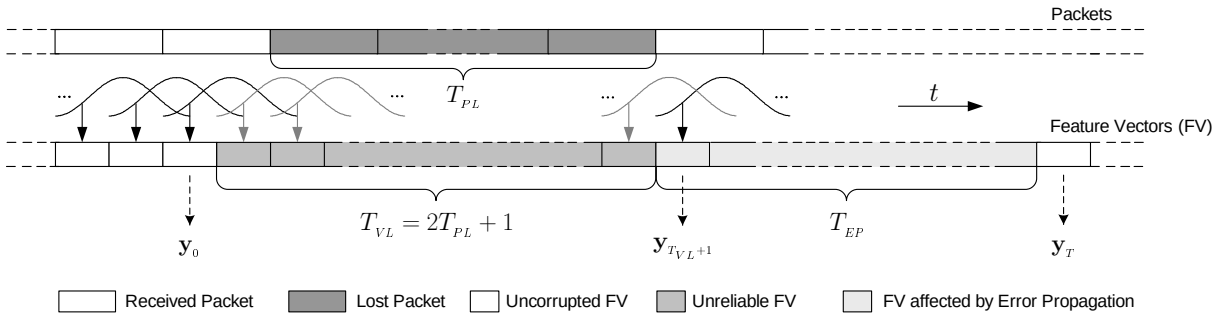


Figura A.13: Scheme of feature vectors (FV) affected by a burst of T_{PL} lost packets.

Observation Probabilities

Unlike other applications, where we can consider that the degradation is approximately constant during an error burst [156, 161], in packet loss mitigation we must consider different types of corrupted data. First, during a packet loss burst there are not channel outputs available. Also, error propagation affects several feature vectors after a packet loss, so they cannot be considered totally correct. Furthermore, distortion due to error propagation varies (decreases) with time. Finally, vectors $\mathbf{y}_0$ and $\mathbf{y}_T$ are correctly decoded, and they are used for initialization of the forward and backward recursions, respectively. Therefore, unlike in the previous formulation of MMSE estimation, we will need different sets of observation probabilities for the different types and levels of degradation.

Fig. A.13 shows a diagram of how a burst of T_{PL} lost packets affects the feature vector extraction. The observed vector sequence at the receiver is $Y = (\mathbf{y}_0, \ldots, \mathbf{y}_T)$. As shown, the vectors in the interval between $t = 1$ and $t = T_{VL}$ correspond to the lost segment and, therefore, they are marked with $VLI = 1$. Since these vectors are extracted from speech generated by the PLC algorithm, we will consider that they are totally unreliable. In addition, error propagation causes a distortion in the following T_{EP} vectors after the burst, so $T = T_{VL} + T_{EP} + 1$. In order to suitably model this variable degradation, we will use a specific observation probability distribution $b_i^{(t)}(\mathbf{y}_t)$ for every t $(t = 0, \ldots, T)$. We will consider the following cases:

- Assuming that vectors at time $t = 0$ or $t = T$ have been obtained from speech correctly decoded, the corresponding observation probabilities must be set as,

$$b_i^{(t)}(\mathbf{y}_t) = \begin{cases} 1 & \mathbf{y}_t = \mathbf{x}^{(i)} \\ 0 & \mathbf{y}_t \neq \mathbf{x}^{(i)} \end{cases} \qquad t = 0;\ t = T \tag{A.44}$$

It must be taken into account that $\mathbf{y}_0$ and $\mathbf{y}_T$ do not have always to be correct vectors, since they will be corrupted in the case of bursts at the beginning or end of the utterance. Equally, a packet loss can happen when the propagated error of a previous loss is still present. In these situations, the following cases must be considered.

- In the case that feature vectors correspond to a lost segment between $t = 1$ and $t = T_{VL}$(that is, during the packet loss burst), the observation probability is

$$b_i^{(t)}(\mathbf{y}_t) = \frac{1}{N} \qquad 1 \leq t \leq T_{VL} \tag{A.45}$$

  There are no valid observations for these time instants, so all the states $\mathbf{x}^{(i)}$ have the same conditional probability given the observation. As the observation probability does not provide any information, the forward-backward algorithm will be only guided by the transition probabilities.

- Finally, we pay attention to those feature vectors which are partially reliable because they are affected by propagated error. As shown in Section A.4.1 (see Fig. A.12), error propagation causes a higher distortion to the first vectors after a packet loss, and diminishes as time goes on. In addition, the distortion level also depends on the packet loss length. For this reason, we model the observation probability as,

$$b_i^{(t)}(\mathbf{y}_t = \mathbf{x}^{(j)}) = P(\mathbf{y}_t = \mathbf{x}^{(j)} | \mathbf{x}^{(i)}, l_{burst}, t_{ep}) \qquad T_{VL} < t < T \tag{A.46}$$

  where l_{burst} is the length of the loss burst and $t_{ep} = t - T_{VL}$ denotes the time instant from the end of the burst. In practice, the observation probabilities can be computed from a stereo training speech database by systematically simulating bursts of lost packets. Thus, given a burst length l_{burst}, the observation probabilities at time $t = T_{VL} + t_{ep}$ are obtained as the following frequencies of appearance,

$$b_i^{(t)}(\mathbf{y}_t = \mathbf{x}^{(j)}) = \frac{n_{j|i}}{\sum_i n_{j|i}} \qquad T_{VL} < t < T \tag{A.47}$$

  where $n_{j|i}$ is the number of times that $\mathbf{y}_t = \mathbf{x}^{(j)}$ given that $\mathbf{x}_t = \mathbf{x}^{(i)}$. When the length of a packet loss burst is greater than a certain number L_{burst} of vectors, the increase in distortion can be considered negligible. Also, we can assume that propagation error disappears beyond T_{EP} feature vectors after a packet loss. Thus,

we only need to store $L_{burst} \times T_{EP}$ different observation distributions. In the case that $l_{burst} > L_{burst}$, the distributions corresponding to L_{burst} are applied.

MMSE Initialization

In order to apply the recursion of equations (A.42) and (A.43) a set of initial conditions is required. Initially, we can consider a packet loss that is sufficiently distant from the previous and next losses, then there is no overlap between their error propagations. In such a case, the following initial conditions are applied to $t = 0$ and $t = T$,

$$\begin{aligned} \alpha_0(i) &= P_i b_i^{(t)}(\mathbf{y}_0)/K_0 \\ \beta_T(i) &= 1 \end{aligned} \qquad (i = 0, 1, \ldots, N-1) \tag{A.48}$$

where P_i is the *a priori* probability of $\mathbf{x}^{(i)}$.

However, a new packet loss could happen before the propagated error of the current burst has disappeared. In this case, we take the same initial conditions as in (A.48), but instead of initializing $\beta_t(i)$ at time T we do it at T' corresponding to the first discarded vector of the next packet loss. Thus, the MMSE reconstruction finishes at $T' - 1$. For the reconstruction of the next packet loss burst, since the feature vector just before this burst is also corrupted, we take $\alpha_0(i) = \gamma_{T'-1}(i)$.

MMSE Reconstruction

MMSE estimation (A.36) computes a reconstructed feature vector $\bar{\mathbf{x}}$ from the quantized vector sequence Y and the prototypes $\{\mathbf{x}^{(i)}; i = 0, \ldots, N-1\}$. However, this approach could lead us to a lower performance because of the applied quantization. To solve this problem, instead of using directly the MMSE estimate of eq. (A.36), we apply the following reconstruction,

$$\hat{\mathbf{x}}_t = \boldsymbol{y}_t + \mathbf{r}(\boldsymbol{y}_t) \tag{A.49}$$

where $\mathbf{r}(\boldsymbol{y}_t)$ is an additive correction factor that is function of the unquantized vector $\boldsymbol{y}_t$.

We empirically verified that $\mathbf{r}(\boldsymbol{y}_t)$ varies smoothly with $\boldsymbol{y}_t$, so we can approximate,

$$\mathbf{r}(\boldsymbol{y}_t) \simeq \mathbf{r}(\mathbf{y}_t) = \bar{\mathbf{x}}_t - \mathbf{y}_t \tag{A.50}$$

Although this approach could be applied to static and dynamic features, we only apply

it directly to the static ones, while the dynamic ones are computed as,

$$\Delta\hat{x}_t = \sum_{l=-L}^{L} w_l \hat{x}_{t+l} \tag{A.51}$$

where $\hat{x}_t$ is the reconstructed sequence of the corresponding static feature, and $\{w_l\}$ are the weights used to compute $\Delta\hat{x}_t$.

A.4.3. MMSE and Soft-Data Decoding

Despite of MMSE estimation technique previously presented is a powerful tool for the reconstruction of distorted feature vectors, we must take into account that the resulting estimates are not fully reliable. Instead of assuming that we are dealing with deterministic data, we can consider a soft-data approach where the estimate has a *pdf* associated [162].

We will consider that, in the recognizer, the observation probability $b_s(\boldsymbol{x}_t)$ of a vector $\boldsymbol{x}_t$ for a given HMM state s of a given acoustic unit is modeled by a mixture of M Gaussians with diagonal covariance matrix,

$$b_s(\boldsymbol{x}_t) = \sum_{m=1}^{M} C_{s,m} \prod_{k=1}^{K} \mathcal{N}\left(x_t(k); \mu_{s,m}(k), \sigma^2_{s,m}(k)\right) \tag{A.52}$$

where $\mathcal{N}\left(x_t(k); \mu_{s,m}(k), \sigma^2_{s,m}(k)\right)$ represents a univariate Gaussian distribution function for the k^{th} feature $x_t(k)$ with mean $\mu_{s,m}(k)$ and variance $\sigma^2_{s,m}(k)$, and $C_{s,m}$ is the corresponding mixture weight.

Assuming that the *pdf* of the estimate is Gaussian, we can consider the uncertainty in the estimation by adding the variances of the MMSE estimates $\sigma^2_{\hat{x}_t}(k)$ to the variances $\sigma^2_{s,m}(k)$ of the HMM Gaussians used during the Viterbi decoding. The variances of the k^{th} feature of the reconstructed vector can be easily computed as,

$$\sigma^2_{\hat{x}_t}(k) = E\left[\left(x_t(k) - \hat{x}_t(k)\right)^2 | Y\right] \tag{A.53}$$

In order to compute the variance $\sigma^2_{\Delta\hat{x}_t}$ of a dynamic feature $\Delta\hat{x}_t$, we assume independence between the random variables $\hat{x}_{t+l}$ $(-L \leq l \leq L)$ in eq. (A.51), so we can express,

$$\sigma^2_{\Delta\hat{x}_t} \simeq \sum_{l=-L}^{L} w_l^2 \sigma^2_{\hat{x}_{t+l}} \tag{A.54}$$

that only depends on the variances $\sigma^2_{\hat{x}_{t+l}}$ of the static feature at times $t+l$. Obviously, this assumption is not true, however, as shown in [169], it does not introduce any meaningful degradation.

A.4.4. MMSE and Weighted Viterbi Algorithm

Weighted Viterbi algorithm (WVA) is other possibility to introduce reliability information of the estimates during Viterbi decoding [170]. The Viterbi algorithm is modified to apply a weighting coefficient on those unreliable vectors to reduce their contribution on the final decoding. This technique is refined in [147] by considering a time-varying reliability factor for every component of the observed feature vector. Thus, the overall weighted probability can be computed transforming (A.52) into,

$$b_s(\hat{\mathbf{x}}_t) = \sum_{m=1}^{M} C_{s,m} \prod_{k=1}^{K} \mathcal{N}\left(\hat{x}_t(k); \mu_{s,m}(k), \sigma^2_{s,m}(k)\right)^{\rho_t^{(k)}} \tag{A.55}$$

where the exponent $\rho_t^{(k)}$ is a weighting factor ($\rho_t^{(k)} \in [0,1]$) applied to each feature k at time instant t. When the component $\hat{x}_t(k)$ is fully unreliable, then $\rho_t^{(k)} = 0$ and, therefore, this component is not taken into account in eq. (A.55). On the other hand, $\rho_t^{(k)} = 1$ indicates that the component $\hat{x}_t(k)$ is fully reliable.

Our problem is to determine a reliability function for MMSE estimates. In this case, we have modeled the speech as an ergodic HMM, where every state corresponds to a prototype vector $\mathbf{x}^{(i)}$, $i = (0, \ldots, N-1)$. The reliability of a MMSE estimate depends on the probability mass function $\gamma_t(i)$ defined in (A.39). If this distribution takes similar values for all prototypes, then the confidence of the reconstructed feature vector is low. On the other hand, if $\gamma_t(i)$ is substantially higher for one prototype, then the estimate should be close to the original. In this regard, entropy can be considered as an uncertainty measure of a given distribution. In our case, we can define the *instantaneous entropy* $h_t^{(k)}$ for a specific reconstructed feature k at time t as,

$$h_t^{(k)} = -\sum_{i=0}^{N-1} \gamma_t^{(k)}(i) \log_2 \gamma_t^{(k)}(i) \tag{A.56}$$

Note that we are assuming that a particular probability distribution $\gamma_t^{(k)}(i)$ can be obtained for every feature k. In the next subsection we will see that this is straightforwardly

accomplished through a SVQ (Split Vector Quantization) partition of the whole feature space.

Now, we can easily define a relation with the reliability factor $\rho_t^{(k)}$ by means of the following expression,

$$\rho_t^{(k)} = 1 - \frac{h_t^{(k)}}{\log_2 N} \tag{A.57}$$

When MMSE estimation does not provide any information to reconstruct a feature component k, $\gamma_t^{(k)}$ presents a uniform distribution for all prototypes, so the entropy of this distribution is maximum and adopts a value equal to $\log_2 N$. In this case, the reliability factor obtained by (A.57) is zero and the decoding in the recognizer is only guided by the transition probabilities. In the opposite case, when $\gamma_t^{(k)}(i) = 1$ for a given index i, and $\gamma_t^{(k)}(i) = 0$ for the rest of indices, then the corresponding entropy is zero, and the reliability factor becomes one, so the observation probability in the decoding stage is not modified.

The reliability factor $\rho_t^{\Delta(k)}$ of a dynamic feature $\Delta\hat{x}_t(k)$ corresponding to a static feature $\hat{x}_t(k)$ at time t can be computed as the following weighted sum,

$$\rho_t^{\Delta(k)} = \sum_{l=-L}^{L} |w_l| \rho_{t+l}^{(k)} \tag{A.58}$$

Considering that $|w_l| < 1$ and $\sum_l |w_l| = 1$, then (A.58) verifies that $\rho_t^{\Delta(k)} \leq 1$.

A.4.5. Experimental Results

In order to apply the proposed techniques, we have used the centroids of the Split Vector Quantization (SVQ) of the ETSI DSR standard [35] as the set of prototype vectors $\{\mathbf{x}^{(i)}; i = 0, \ldots, N-1\}$. This vector set provides an accurate representation of the feature space [47]. The features are grouped into pairs and quantized by means of seven SVQ with 64 centroids, except the last one (MFCC(0) and $\log E$) which has 256 centroids, so, without loss of generality, the MMSE estimation described in (A.36) is really applied seven times (one for each SVQ). Therefore, probabilities $\gamma_t(i)$ are computed for every feature pair. Additionally, our techniques consider different conditions of error propagation depending on the length of the packet loss and the position after it. We consider that the propagated error at $T_{EP} = 20$ feature vectors after a packet loss has practically disappeared. Furthermore, the maximum length of burst considered in this work corres-

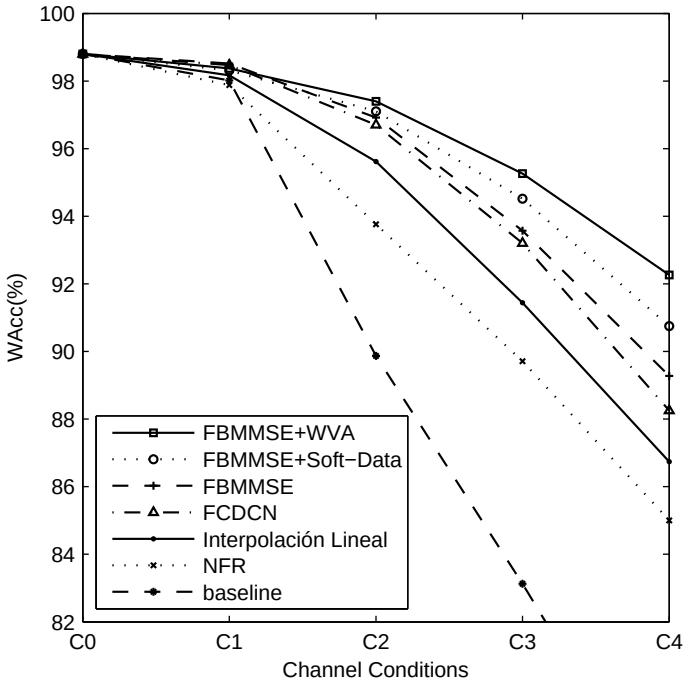


Figura A.14: WAcc results from decoded speech using G.729A codec.

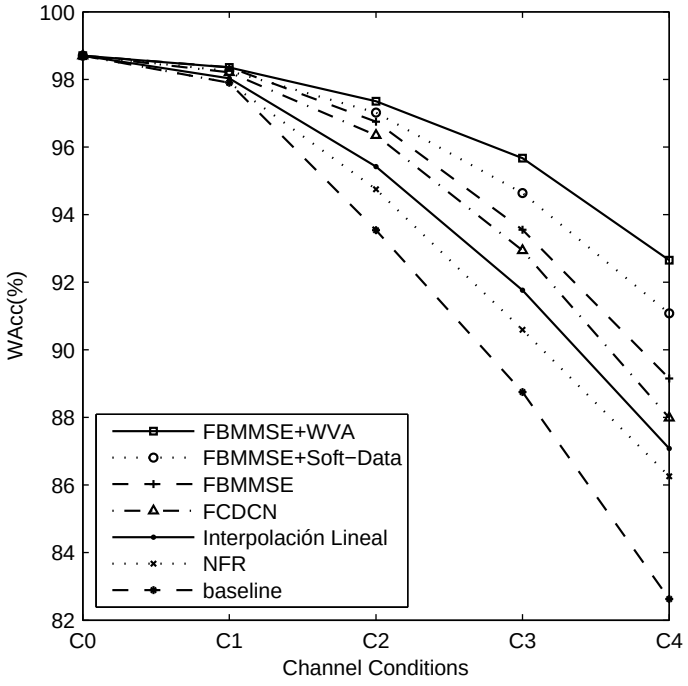


Figura A.15: WAcc results from decoded speech using AMR 12.2 kbps codec.

ponds to $L_{burst} = 5$ packets, since the distortion increment caused by longer bursts can be considered negligible.

We compare now the effectiveness of some common concealment techniques found in the literature with our MMSE estimation and soft-data decoding techniques. Figures A.14 and A.15 show the word accuracy (WAcc) obtained on the Aurora 2 database (see Section A.2) by G.729A and AMR mode 12.2 kbps, respectively. The baseline corresponds to carrying out the recognition task from decoded speech including the PLC defined by the standard codec. Furthermore, we have included the results obtained by applying some basic mitigation techniques as nearest frame repetition (NFR) and linear interpolation. As mentioned in Section A.4.1, the PLC algorithms in standard codecs are usually based on repetition and progressive muting, which decreases the recognition performance. The muting operation is faster in G.729A than in AMR, so the baseline result is lower for the first one. NFR and linear interpolation, which use the feature vectors before and after a burst to conceal the loss, improve noticeably the baseline performance, since they substitute feature vectors affected by muting (artificial silence). However, these basic techniques do not take into account that the vectors just after the burst are corrupted by error propagation. In our previous work [122], we considered error propagation for EFR (a CELP-based codec) as an acoustic noise. In order to compensate it, we used FCDCN (Fixed Codework-Dependent Cepstral Normalization) [155], and linear interpolation to reconstruct the burst. As shown in figures A.14 and A.15, FCDCN outperforms the results achieved by linear interpolation.

We can also see in figures A.14 and A.15 that our basic MMSE estimation (referred to as FBMMSE, to make clearer the use of Forward-Backward recursions) achieves results slightly better than FCDCN. In fact, both techniques are quite similar, since they apply a correction vector that depends on an instantaneous SNR (or distortion level), although FBMMSE is enhanced with the introduction of speech temporal correlations by means of HMM modelling. Moreover, substantial improvements are obtained when the reliability of the FBMMSE estimates are used in the decoding stage of the recognizer. The results named as FBMMSE+Soft-Data and FBMMSE+WVA correspond to the two approaches proposed in this work. The FBMMSE+soft-data approach is the one based on the increase of the acoustic-model variances, and FBMMSE+WVA is the approach based on WVA, whose weights are computed dynamically from the distribution entropy used in MMSE estimation. Although the FBMMSE+WVA method is heuristic, its performance is much better than that of the FBMMSE+soft-data approach, which can be caused by the lack of validity of some adopted assumptions in the FBMMSE+soft-data approach. However, the

proposed FBMMSE methods has two drawbacks when they are applied to the features extracted from decoded speech. The first one is that they need to store the observation probability tables corresponding to $L_{burst} \times T_{EP}$ considered conditions. In addition, the FBMMSE increases the delay in $T_{EP} = 20$ feature vectors in comparison with the other techniques considered here. In the next section, we will see how to relieve these problems by means of a feature extraction carried out directly from the codec parameters.

A.5. Transcoding-based Solutions

The NSR architecture can be modified in order to obtain the recognition features directly from the codec parameters, which are encoded in the bitstream generated by the speech coder. This approach avoid the intermediate reconstruction of the speech signal by introducing a bitstream-based feature extraction that directly transforms the codec parameters into recognition features. We will refer to this architecture as bitstream-based NSR (B-NSR). The feature extraction can also be viewed as a *trans-parameterization*, or transcoding, if we take into account that the speech signal could also be reconstructed from recognition features (as in the ETSI XFE or XAFE DSR standards).

In this section, we develope two transparameterization methods. The first one is devoted to G.729 and AMR 12.2 kbps, although it could be applied to any CELP-based Codecs through slight modifications. The second one is devoted to the iLBC codec. Furthermore, we propose efficient packet loss concealment algorithms for both proposals.

A.5.1. Bit-stream Feature Extraction for CELP-based Codecs

G.729A and AMR 12.2 kbps compute a set of LPC coefficients every 10 ms, transmitting the corresponding quantized set of linear spectrum pairs (LSP). Furthermore, both codecs encode the rest of parameters, which are dedicated to build the excitation signal, using a subframe basis of 5 ms ($N_{sf} = 40$ samples).

Mel Frequency Cepstral Coefficients $mfcc(k)$ $(k = 0, \ldots, 12)$ are obtained following the procedure described in [35] but substituting the FFT spectrum by the LPC spectrum,

$$|H(\omega_i)| = \frac{\sigma}{\left|1 + \sum_{l=1}^{10} a_l e^{-jw_i l}\right|} \tag{A.59}$$

where $\omega_i = 2\pi i/D\,(i = 0, \ldots, D-1 = 255)$, and σ is the LPC gain. From this approximation, we can easily derive the following relations,

$$mfcc(k) = \begin{cases} F\log\sigma + mfcc'(k) & k = 0 \\ mfcc'(k) & k = 1, \ldots, 12 \end{cases} \tag{A.60}$$

where $mfcc'(k)$ represents the cepstral coefficients corresponding to the normalized LPC spectrum $|H'(\omega)| = |H(\omega)|/\sigma$, and F is the number of filters used in the filter-bank defined in [35].

Log-Energy can be computed from the LPC spectrum and the excitation power σ^2 by means of the following expression,

$$\log E = \log\sigma^2 + \log\xi \tag{A.61}$$

where ξ is the energy corresponding to the normalized LPC spectrum defined as,

$$\xi = \frac{1}{2\pi}\int_{-\pi}^{\pi} |H'(\omega)|^2 d\omega \tag{A.62}$$

To avoid the synthesis of the excitation $e(n)$, we assume that the entries of the adaptive and fixed codebooks are uncorrelated, so the excitation power (squared LPC gain) corresponding to subframe m is expressed as,

$$\sigma^2(m) = g_p^2\sigma_p^2(m) + g_c^2\sigma_c^2(m) \tag{A.63}$$

where $\sigma_p^2(m)$ and $\sigma_c^2(m)$ are the powers of the adaptive and fixed codebook contributions, $e_p(n)$ and $e_c(n)$, respectively, in the subframe m. The gains, g_p and g_c, and the innovative signal, $e_c(n)$, are obtained directly from the received bit-stream for every subframe. Finally, σ^2 can be obtained by averaging the excitation power $\sigma^2(m)$ of the $N_f = 5$ corresponding subframes, since recognition features are computed over segments of 25 ms ($N_{sf} \cdot N_f = 200$ samples).

Now, we need $\sigma_c^2(m)$ and $\sigma_p^2(m)$. First, σ_c^2 can be directly computed as,

$$\sigma_c^2(m) = \frac{1}{N_{sf}}\sum_{n=0}^{N_{sf}-1} e_c^2(n) \tag{A.64}$$

In order to obtain $\sigma_p^2(m)$, we must take into account that $e_p(n)$ is not available in the received bit-stream, but it is obtained from the past excitation signal through a long-term

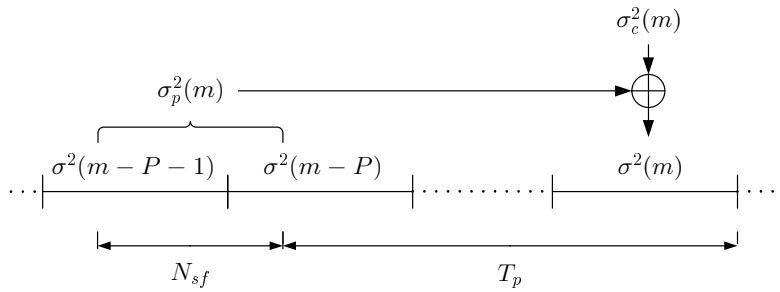


Figura A.16: Excitation power computation from codec parameters at subframe rate.

predictor with a pitch lag T_p. Thus, we approximate $\sigma_p^2(m)$ by means of the following expression,

$$\sigma_p^2(m) = \eta\sigma^2(m-P) + (1-\eta)\sigma^2(m-P-1) \tag{A.65}$$

where,

$$\eta = \frac{T_p \bmod N_{sf}}{N_{sf}} \qquad P = \left\lfloor \frac{T_p}{N_{sf}} \right\rfloor$$

Thus, the contribution of the adaptive signal is computed as a weighted average of the total powers of the subframes which contains the previous pitch period (see Fig. A.16). Nevertheless, when $T_p < N_{sf}$ (the minima of T_p are 17 and 20 for AMR and G.729, respectively) $\sigma_p^2(m)$ depends on the current subframe power itself. To solve this problem, we will approximate the power of the current subframe by the innovation power computed in eq. (A.64), so $\sigma^2(m) \simeq g_c^2\sigma_c^2(m)$, in order to compute the adaptive contribution in (A.65).

This feature extraction method obtains WAcc(%) results of 98.82 and 98.79 for G.729A and AMR (12 kbps), respectively. These results are slightly higher than those ones presented in Section A.2 for speech recognition from decoded speech.

Feature Reconstruction

The mitigation techniques proposed in Section A.4 can be straightforwardly extended to B-NSR. However, we can take advantage of the feature extraction process from bit-stream parameters in order to obtain an efficient implementation of these techniques. Features $mfcc(1)$–$mfcc(12)$ are extracted directly from the LPC spectrum, so, in principle, they avoid the distortion caused by error propagation. On the other hand, $\log E$ and $mfcc(0)$, which also depend on the excitation energy, still suffer from error propagation. Taking advantage of these characteristics, we employ the MMSE-based technique to reconstruct

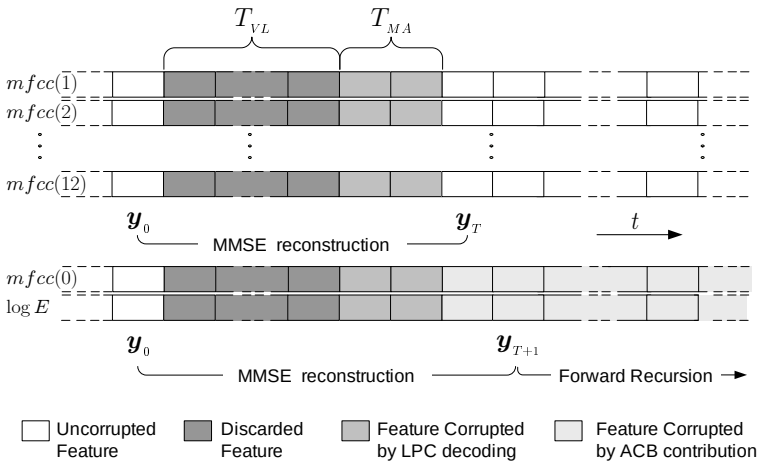


Figura A.17: Diagram of ASR feature reconstruction in B-NSR architecture.

only those vectors more affected by packet loss. Furthermore, we propose a forward recursive method to mitigate the propagation error in log-Energy and $mfcc(0)$.

MMSE Reconstruction The diagram in Fig. A.17 can be useful to understand how a packet loss affects this B-NSR approach. In first place, T_{VL} feature vectors are discarded due to data loss. Furthermore, standard codecs usually apply a moving-average (MA) predictor to reduce the bit-rate of LPC coefficients. Thus, depending on the order of this predictor, the first T_{MA} vectors after a packet loss will be damaged (in Fig. A.17, a second-order predictor was supposed, hence $T_{MA} = 2$). Due to this, in order to reconstruct $mfcc(1)$–$mfcc(12)$ we employ the MMSE-based techniques described in Section A.4 (with appropriate observation probabilities), but considering that error propagation has a duration of T_{MA} vectors instead of T_{EP}. Therefore, as can be seen in Fig. A.17, (forward-backward) MMSE estimation is applied from $t = 0$ to $t = T$ ($T = T_{VL} + T_{MA} + 1$), using $\mathbf{y}_0$ and $\mathbf{y}_T$ for initialization. The delay caused by the mitigation technique is considerably reduced because $T_{MA} \ll T_{EP}$.

On the other hand, log-Energy and $mfcc(0)$ parameters show a long error propagation due to their dependency on σ^2. Thus, even when the LPC spectrum is correct, these parameters will be distorted due to the adaptive contribution σ_p^2, which depends on the energy of previous subframes (see eq. (A.65)). In this case, we will apply MMSE estimation from $t = 0$ to $t = T + 1$, so the last MMSE estimates will correspond to T although the error is propagated beyond $T + 1$. This remaining degradation is treated with the forward recursion described in the following.

Forward Recursive Reconstruction of Energy Features The MMSE-based techniques are used to mitigate the discarded feature vectors and those ones affected by the moving-average predictor. However, propagation error is longer for $mfcc(0)$ and log-Energy, due to the error remaining in the excitation power. From now on, since $mfcc(0)$ and $\log E$ are the only considered features, we will use the vector notation to represent this feature pair. Considering expressions (A.60) and (A.61), we can express the extraction of the feature pair $\boldsymbol{x}_t$ as,

$$\boldsymbol{x}_t = \begin{cases} mfcc_t(0) = mfcc'_t(0) + F \log \sigma_t \\ \log E_t = \log \xi_t + \log \sigma_t^2 \end{cases} \tag{A.66}$$

As said above, the last MMSE reconstructed feature pair $\hat{\mathbf{x}}_t$ is obtained at $t = T$, i.e. once the LPC set have already been correctly decoded. Therefore, from this point on, the LPC contributions $mfcc'_t(0)$ and ξ_t can be computed correctly, and the only possible source of error in (A.66) will be given by σ_t^2. Thus, at time $t = T$, the excitation power of the last MMSE reconstructed vector can be estimated as,

$$\hat{\sigma}_{_T}^2 = \frac{\hat{E}_{_T}}{\xi_{_T}} \tag{A.67}$$

where $\hat{E}_{_T}$ is derived form the log-Energy component of $\hat{\mathbf{x}}_{_T}$.

Considering expressions (A.63) and (A.65), the excitation power of the current subframe can be computed by means of the following recursion,

$$\begin{aligned} \sigma^2(m) \;=\; & \underbrace{g_p^2 \left(\eta\sigma^2(m-P) + (1-\eta)\sigma^2(m-P-1)\right)}_{\text{Adaptive Contribution}} \\ & + \underbrace{g_c^2\sigma_c^2(m)}_{\text{Fixed Contribution}} \end{aligned} \tag{A.68}$$

Our proposal is to use this recursion in order to extend forward the last MMSE estimate $\hat{\mathbf{x}}_{_T}$. Thus, we will compute the subsequent reconstructed vectors by using the value of $\hat{\sigma}_{_T}^2$ obtained in (A.67) as initial conditions. Finally, since the recursion works on a subframe basis, an average of N_f subframes is carried out in order to obtain the excitation power estimate $\hat{\sigma}_t^2$ at a suitable rate, and subsequently the reconstructed feature pairs $\hat{\mathbf{x}}_t$ $(t > T)$.

Reliability Measures of Reconstructed Features During MMSE reconstruction, the reliability measures for the *soft-data* and *weighted Viterbi* approaches (variances and

reliability factors of the reconstructed features) are obtained, as described in Sections A.4.3 and A.4.4, from probabilities $\gamma_t(i)$ (defined in (A.39) for MMSE estimation).

Since MMSE estimation is not used for $t > T$, these probabilities are not available. Alternatively, in order to obtain the required reliability measures for $t > T$, we propose the use of the following probability distribution,

$$\zeta_t(j) \equiv P\left(\hat{\mathbf{x}}_t = \mathbf{x}^{(j)} | \boldsymbol{\gamma}_{_T}\right) \quad j = 0, \ldots, N-1; \;\; t > T \tag{A.69}$$

where $\boldsymbol{\gamma}_{_T}$ is a vector containing the last conditional probabilities $\gamma_{_T}(i)$ $(i = 0, \ldots, N-1)$ obtained from MMSE reconstruction. Observe that we are considering that the error in $\hat{\mathbf{x}}_t$ $(t > T)$ will be given by the error in $\hat{\mathbf{x}}_{_T}$ propagated forwardly.

In order to compute probabilities $\zeta_t(j)$, we will track the time evolution of every prototype vector $\mathbf{x}^{(i)}$. We will see that these (evolved) vectors tend to merge as time goes on, so probabilities $\zeta_t(j)$ can be merged (summed) in the same way. Since the LPC contributions in eq. (A.66) are known, we can consider a power $\hat{\sigma}^2_{_T}(i)$ for every $\mathbf{x}^{(i)}$ (from its log-Energy parameter) and observe its evolution by means of the following expression,

$$\hat{\sigma}^2_t(i) = G_t \cdot \hat{\sigma}^2_{_T}(i) + \tilde{\sigma}^2_t \quad t > T \tag{A.70}$$

where $\tilde{\sigma}^2_t$ is the average of excitation powers (N_f subframe powers) obtained by taking $\tilde{\sigma}^2_{_T} = 0$ as initial condition in recursion (A.68), and G_t can be computed as,

$$G_t = \frac{\hat{\sigma}^2_t - \tilde{\sigma}^2_t}{\hat{\sigma}^2_{_T}} \quad t > T \tag{A.71}$$

As time goes on, G_t will decrease since $\tilde{\sigma}^2_t$ will tend to be equal to $\hat{\sigma}^2_t$. When G_t is zero, the reconstructed power $\hat{\sigma}^2_t$ (obtained by the recursion described in the previous subsection) is given exclusively by $\tilde{\sigma}^2_t$ and the explicit dependency with $\hat{\sigma}^2_{_T}$, and hence uncertainty, disappears. This evolution can be interpreted as if the different prototype vectors merge as time goes on, so finally there is only one surviving prototype.

Once every $\hat{\sigma}^2_t(i)$ is computed for $t > T$, we can obtain its corresponding transformed feature pair $\mathbf{z}^{(i)}_t$ through (A.66). In order to group the prototype vectors merging at time t, we define the following sets,

$$Q_t(j) = \{i \in (0, 1, \ldots, N-1) \,|\, j = q(\mathbf{z}^{(i)}_t)\} \tag{A.72}$$

where the function $k = q(\mathbf{p})$ returns the index k of the prototype $\mathbf{x}^{(k)}$ which is the nearest

neighbor to the transformed feature pair $\mathbf{p}$. These sets allow us to compute the evolution of the probability mass function $\zeta_t(j)$ $t > T$ by means of the following expression,

$$\zeta_t(j) = \begin{cases} \sum_{i \in Q_t(j)} \gamma_T(i) & Q_t(j) \neq \varnothing \\ 0 & Q_t(j) = \varnothing \end{cases} \quad (j = 0, \ldots, N-1) \tag{A.73}$$

Thus, the conditional probability $\zeta_t(j)$ is computed by adding those $\gamma_T(i)$ corresponding to the set $Q_t(j)$, i.e. those prototypes which are merged in $\mathbf{x}^{(j)}$ at time t. Time evolution will cause a progressive concentration of $\zeta_t(i)$ around the observed feature at time t, diminishing its associated variance and entropy, and hence the uncertainty in the recursive estimation. In addition to this convergence, this solution is based on a forward recursion, so it does not introduce any delay.

A.5.2. iLBC Transparameterization Approach

The iLBC codec (mode 15.2 kbps) operates on speech frames of 160 samples which are divided into four sub-frames. Each iLBC frame contains one set of LSFs (Line Spectrum Frequencies) obtained from a 10th order linear prediction analysis carried out once every frame using an asymmetric window centered in the third subframe. On the other hand, the DSR feature extraction algorithm is performed over 200 samples (25 ms) every 80 samples. Because of the differences in the speech signal analysis between DSR and iLBC, the following interpolation of the LSF coefficients is applied,

$$\begin{aligned} \overline{LSF}_{2 \cdot n} &= \frac{6 \cdot LSF_{n-1} + 13 \cdot LSF_n + 1 \cdot LSF_{n+1}}{20} \\ \overline{LSF}_{2 \cdot n+1} &= \frac{1 \cdot LSF_{n-1} + 13 \cdot LSF_n + 6 \cdot LSF_{n+1}}{20} \end{aligned} \qquad n = 1, 2, \ldots \tag{A.74}$$

where $\overline{LSF}_{2 \cdot n}$ and $\overline{LSF}_{2 \cdot n+1}$ are the LSF sets of DSR frames $2n$ y $2n+1$ and LSF_n is the LSF set of the iLBC frame n. Thereby, we double the number of LSF sets provided by the iLBC coder.

MFCC coefficients can be computed following the DSR standard replacing the FFT spectrum by the following LPC spectrum,

$$|H'(\omega_i)| = \sigma |H(\omega_i)| = \frac{\sigma}{1 + \sum_{i=1}^{10} a(k) e^{-j \omega_i k}} \tag{A.75}$$

where σ is the LPC gain, $\omega_i = 2\pi i / N$ ($i = 0$, ..., $N-1$) and $|H(\omega_i)|$ is the gain-normalized LPC spectrum evaluated with $N = 256$ points [122].

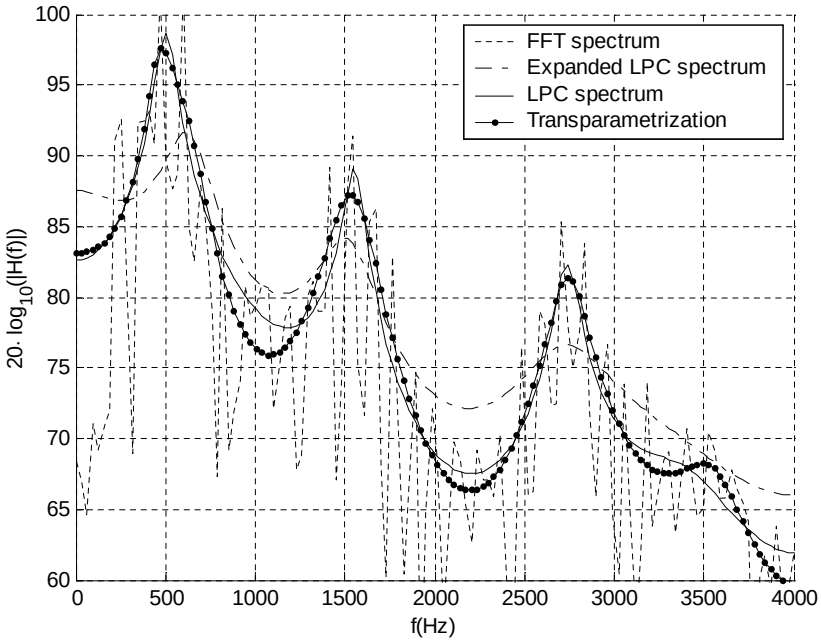


Figura A.18: *Distortion produced by the spectrum expansion in the LPC analysis and the proposed approximation used by the transparametrization approach.*

At this point, a remarkable characteristic of the LPC analysis of the iLBC coder has to be discussed because of its effect in the proposed approach. That is the spectrum expansion which is performed according to,

$$H_{exp}(z) = H(\frac{z}{\gamma})$$

where γ is the expansion factor (equal to 0.9) and $H_{exp}(z)$ is the expanded LPC spectrum. This operation has several effects (see figure A.18). The dynamic range of the spectrum is reduced, therefore the length of the impulse response of the synthesis filter becomes shorter. Thus, in case of a packet loss, the filter does not excessively propagate the error in the filter memory. Furthermore, the location of the poles around the origin is compressed reducing the quantization space.

However, this expansion is a serious inconvenient for our proposal because it introduces a considerable distortion in the LPC spectrum. It can be argued that this expansion can be reversed in the decoder. Nevertheless, the LSF quantization process prevents this possibility in the decoder since it would lead to unstable LPC filters.

To cope with this situation it is needed to consider the spectral characteristic of the

coded residual signal. In this way, the decoded residual signal is processed to obtain a new set of LPC parameters ($a_{res}(k)$) which characterizes the LPC spectrum of the residual signal $H_{res}(\omega_i)$. Now, we can use these coefficients to obtain an improved LPC spectrum estimation by means of the following expression,

$$|\hat{H}(\omega_i)| = H_{exp}(\omega_i) \cdot H_{res}(\omega_i) = \\ = \frac{1}{\left|1 + \sum_{l=1}^{10} a_{exp}(l) e^{-j\omega_i l}\right|} \cdot \frac{1}{\left|1 + \sum_{k=1}^{10} a_{res}(k) e^{-j\omega_i k}\right|} \tag{A.76}$$

This expression accounts for the non-flat shape of the residual spectrum envelope after the expansion operation. Figure A.18 shows the new spectrum estimation $|\hat{H}(\omega_i)|$, with its corresponding gain $\hat{\sigma}$, marked as "B-NSR". It is observed that approximates the FFT spectrum even better than the original LPC spectrum.

The MFCC coefficients are obtained by following the process described in the previous section, but replacing the LPC spectrum by the estimation obtained in equation (A.76).

At the decoder side, the energy E_{res} and the corresponding LPC coefficients ($a_{res}(k)$) are computed from the residual signal at the corresponding DSR rate (every 10 ms in frames of 25 ms long). The gain $\hat{\sigma}$ is computed using the following expression:

$$\hat{\sigma}^2 = \frac{E_{res}}{\frac{1}{2\pi}\int_{-\pi}^{\pi} \left|\frac{1}{1+\sum_{k=1}^{10} a_{res}(k) e^{-j\omega k}}\right|^2 d\omega} \tag{A.77}$$

Finally, MFCC coefficients are computed according (A.74) using the decoded LSF parameters and $\log E$ is computed with the following expression:

$$\log E = \log\left(\frac{1}{2\pi}\int_{-\pi}^{\pi} |\hat{\sigma}\hat{H}'(\omega)|^2 d\omega\right) \tag{A.78}$$

The integration operations, which appears in expressions (A.77) and (A.78), are computed in the time domain by means of sums.

Packet Loss Concealment

In practice, all kind of errors due to an IP channel can be considered as packet losses. To alleviate these packet losses, a PLC algorithm is required. In the DSR Aurora standard a repetition scheme of correctly MFCC coefficients is implemented, since it achieves better results than interpolation.

In our proposal, a mitigation technique is implemented in the parameter domain using: LSF_{res} (calculated from $a_{res}(k)$), E_{res} and $\overline{LSF}$. Several combinations of mitigation techniques have been tested and we found that the highest performance was obtained using lineal interpolation for parameters derived from the residual signal (LSF_{res}, E_{res}) and a repetition technique (similar to the DSR concealment algorithm) for $\overline{LSF}$.

A.5.3. Experimental Results

In order to apply the MMSE-based techniques we have taken into account the same considerations explained in Section A.4.5, but taking $T_{EP} = T_{MA} + 1$ ($T_{MA} = 2$ for AMR 12.2 and $T_{MA} = 4$ for G.729A) instead of $T_{EP} = 20$. Figures A.19 and A.20 show a comparison of the results obtained by the B-NSR approach (solid lines) for G.729A and AMR mode 12.2 kbps, respectively. We have also included the best results obtained in Section A.4 (dotted lines), and the baseline from decoded speech. In addition, we also show the performance of other techniques found in the literature. The results marked as UD corresponds to the uncertainty decoding rule proposed for NSR in [163], and those ones named as WVA refer to the weighted Viterbi solution given in [173] applied to the NSR context. In this last case, the lost feature are reconstructed by applying NFR, and their confidence values are computed as $\rho_t = \alpha^{\tau}$, where $\tau = \min(T_{VL} - t, t)$ $(t = 0, \ldots, T_{VL})$. The optimal value for α was empirically determined as 0.7. Finally, we also include the results obtained by the DSR architecture described in [35], which can be considered as an upper limit for NSR and B-NSR.

As can be seen, our proposals clearly improve the results obtained by the other NSR solutions. We have to highlight that the performances of our B-NSR approaches are higher than those ones obtained from decoded speech. Furthermore, the B-NSR approaches only increase the delay in $T_{MA} + 1$ feature vectors against $T_{EP} = 20$ for the approaches based on decoded speech. Equally, the observation probability tables are reduced in the same proportion. Although the results of both codecs are not directly comparable, since they do not have the same bit-rate, the B-NSR proposals based on AMR slightly outperforms those ones based on G.729A. This result could be justified because T_{MA} is shorter for AMR. Finally, note that the best results are obtained by FBMMSE+WVA, which reduces considerably the difference with DSR with a moderate increase of latency.

Finally, Table A.13 shows a brief summary of results. In first place, we show the baseline results for G.729, AMR 12.2 and iLBC. In addition, the best results obtained for our B-NSR proposals are also included. In this comparison, we must highlight that our

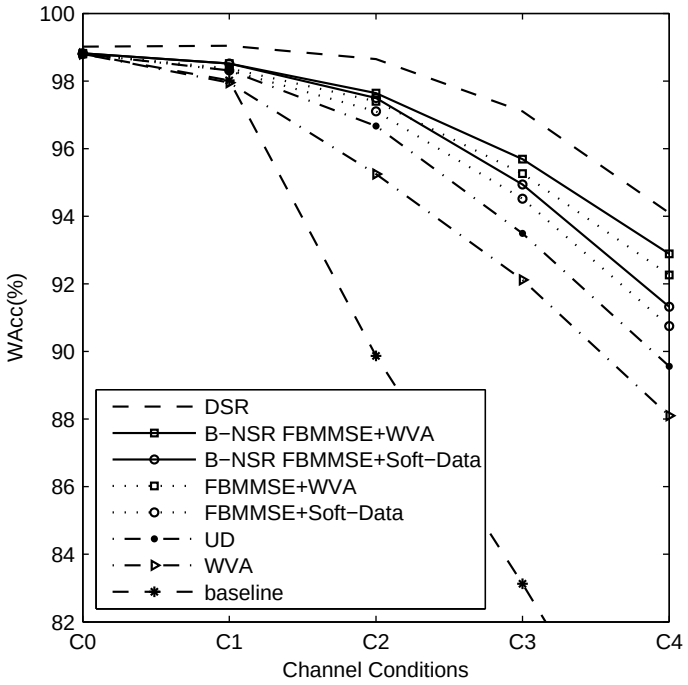


Figura A.19: WAcc results for G.729A.

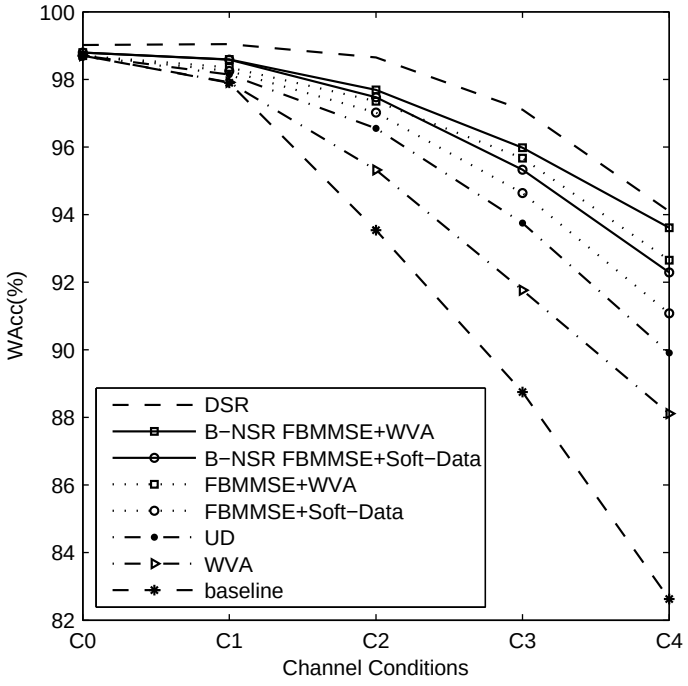


Figura A.20: WAcc results for AMR mode 12.2 kbps.

ASR Architecture	*Bit-Rate (kbps)*	*Channel Conditions*					*Avg. Value*
		C0	*C1*	*C2*	*C3*	*C4*	
G.729 baseline	*8*	98.81	98.02	89.87	83.13	75.98	89.16
AMR baseline	*12.2*	98.68	97.93	93.97	88.55	83.07	92.44
iLBC baseline	*15.2*	98.96	98.56	96.35	92.43	87.11	94.68
G.729 B-NSR	*8*	98.82	98.52	97.64	95.69	92.89	96.71
AMR B-NSR	*12.2*	98.79	98.59	97.69	95.96	93.61	96.93
iLBC B-NSR	*15.2*	98.94	98.85	98.22	96.29	93.32	97.12
DSR FE	*4.75*	99.04	99.04	98.65	97.10	94.10	97.59

Tabla A.13: Summary of B-NSR results.

proposals clearly outperform the speech recognition from decoded speech and they achieve to reduce the performance differences against DSR using its own PLC algorithm (NFR). Nevertheless, the CELP-based B-NSR solutions require more complex PLC algorithms than that of iLBC B-NSR approach (repetition and interpolation of parameters), although this last one presents a higher bit-rate.

A.6. Conclusions

Please refer to Chapter 9, to read the main conclusions extracted from this Ph.D. Thesis.

Bibliografía

[1] *The status of Voice over Internet Protocol (VoIP) Worldwide, 2006*, ITU New Initiatives Programme: The Future of Voice, 2007. Disponible en linea: http://www.itu.int/osg/spu/ni/voice/papers/FoV-VoIP-Biggs-Draft.pdf 1.1

[2] P. Chandra y D. Lide, *WiFi telephony: Challenges and solutions for voice over WLANs*. Elsevier Inc., 2007. 1.1, A.1

[3] L. R. Rabiner y B. H. Juang, *Fundamentals of Speech Recognition*. Prentice-Hall, 1993. 2.2, 2.3

[4] K. F. Lee, *Automatic Speech Recognition (The Development of the Sphinx System)*. Kluwer Academic Publishers, 1989. 2.2.1, 2.5.2

[5] N. Sugamura, K. Shikano, y S. Furui, "Isolated word recognition using phoneme-like templates," *in proceedings of IEEE ICASSP*, 1983. 2.2.1

[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. 2.2.1

[7] Y. Gong, "Speech recognition in noisy environments: A survey," *Speech Communication*, vol. 16, núm. 3, págs. 261–291, 1995. 2.2.1

[8] L. R. Rabiner y R. W. Schafer, *Digital Processing of Speech Signals*. Prentice-Hall, 1978. 2.3, 3.5.1

[9] S. Davis y P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 28(4), págs. 357–366, 1980. 2.3, 2.3.3, 2.3.4, 2.3.4

[10] A. V. Oppenheim y R. W. Schafer, *Digital signal Processing.* Prentice-Hall Inc., 1975. 2.3, 2.3.4, 4.6.1

[11] J. Markel y A. Gray, *Linear Prediction of Speech.* Springer-Verlag, 1976. 2.3.1

[12] A. V. Oppenheim, R. W. Schafer, y J. R. Buck, *Discrete-time Signal Processing.* Prentice Hall, 1975. 2.3.1, 4.6.1

[13] J. Makhoul, "Linear prediction: A tutorial review," *Proceedings IEEE*, vol. 63, págs. 561–580, 1975. 2.3.2

[14] J. R. Deller, J. G. Proakis, y J. H. L. Hansen, *Discrete-time processing of speech signals.* Prentice-Hall, 1993. 2.3.2, 2.3.3, 2.3.4

[15] E. Zwicker y H. Fastler, *Psychoacoustics.* Springer-Verlag, 1990. 2.3.3

[16] J. C. Junqua y J. P. Haton, *Robustness in automatic speech recognition.* Kluwer Academic Publishers, 1996. 2.3.3

[17] L. R. Rabiner, M. M. Sondhi, y S. E. Levinson, "A vector quantizer incorporating both LPC shape and energy," *in proceedings of IEEE ICASSP*, 1984. 2.3.5

[18] S. Furui, "Speaker-independent isolated word recognition using dynamic features of speech spectrum," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 34, págs. 52–59, 1986. 2.3.5, 2.3.5

[19] B. Hanson y T. Applebaum, "Robust speaker-independent sord recognition using static, dynamic and acceleration features: experiments with lombard and noisy speech," *in proceedings of IEEE ICASSP*, págs. 857–860, 1990r. 2.3.5

[20] R. E. Bellman, *Dynamic Programming.* Princenton Univ. Press, 1957. 2.4.1

[21] T. K. Vintsjuk, "Recognition of words of oral speech by dynamic programming," *Kibernetica*, vol. 4, núm. 1, págs. 81–88, 1968. 2.4.1

[22] 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. 2.5

[23] S. E. Levinson, L. R. Rabiner, y M. 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. 2.5

[24] L. R. Rabiner, B. H. 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. 2.5

[25] 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. 2.5.1

[26] R. Bakis, "Continuous speech recognition via centisecond acoustic states," *The Journal of the Acoustic Society of America*, vol. 59, 1976. 2.5.2

[27] 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. 2.5.2, 6.6.1

[28] L. R. Rabiner y B. H. Juang, "An introduction to Hidden Markov Models," *IEEE Acoustics, Speech and Signal Processing Magazine*, págs. 4–16, 1986. 2.5.2

[29] 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. 2.5.2

[30] L. Baum, "An inequality and associated maximization technique in statistical estimation of probabilistic functions of markov processes," *Inequalities*, págs. 1–8, 1972. 2.5.2

[31] 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. 2.5.2

[32] 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

[33] Z. Tan y I. Varga, *Automatic Speech Recognition on Mobile Devices and over Communication Networks*. Springer-Verlag, 2008, cap. Network, Distributed and Embedded Speech Recognition: An Overview, págs. 1–23. 3.2

[34] P. Haavisto, "Speech recognition for mobile comunications," *in proceedings of the COST Workshop on Robust Methods for Speech Recognition in Adverse Conditions*, 1999. 3.2.1

[35] *Distributed speech recognition; Front-end feature extraction algorithm; Compression algorithms*, ETSI ES 201 108 v1.1.2, 2000. 3.2.1, 4.6.1, 4.7, 6.3.3, 6.3.4, 6.3.4, 6.4.6, 7.2, 7.3.4, 7.5, A.1, A.2.1, A.2.2, A.4.5, A.5.1, A.5.1, A.5.3

[36] 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. 3.2.1, 4.6, A.2.1

[37] *Advanced front-end feature extraction algorithm*, ETSI ES 202 050 v1.1.1, 2002. 3.2.1, 6.3.3, 6.3.4, A.1

[38] 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. 3.2.1

[39] *Distributed Speech Recognition; Extended Front-end Feature Extraction Algorithm; Compression Algorithm, Back-end Speech Reconstruction Algorithm*, ETSI ES 202 211, 2003. 3.2.1, 6.3.3, 6.3.4, A.1

[40] *Distributed Speech Recognition; Extended Advanced Front-end Feature Extraction Algorithm; Compression Algorithm*, ETSI ES 202 212, 2005. 3.2.1, 6.3.3, 6.3.4, A.1

[41] 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. 3.2.1, 6.3.3

[42] H. Schulzrinne, R. Frederick, y V. Jacobson, "RTP: A transport protocol for Real-Time applications," *RFC 1889*, 1996. 3.2.1

[43] A. M. Peinado y J. C. Segura, *Speech Recognition over Digital Channels: Robustness and Standards*. Wiley, 2006. 3.2.2, 3.2.3, 4.2.3, A.1

[44] J. A. X. Zhong y M. Clements, "Speech coding and transmission for improved automatic recognition," *in Proceedings of ICSLP*, 2002. 3.2.2

[45] J. S. Yoon, G. H. Lee, y H. K. Kim, "A mfcc-based celp speech coder for server-based speech recognition in network environments," *IECIE Transactions on Electronics, Communications and Computer Sciences*, vol. E90-A, págs. 626–632, 2007. 3.2.2

[46] J. Turunen y D. Vlaj, "A study of speech coding parameters in speech recognition," *in proceedings of INTERSPEECH*, 2001. 3.2.2

[47] V. V. Digalakis, L. G. Neumeyer, y M. Perakakis, "Quantization of cepstral parameters for speech recognition over the world wide web," *IEEE Journal on Selected Areas in Communications*, vol. 17, págs. 82–90, 1999. 1, 6.4.6, A.4.5

[48] A. S. Spanias, "Speech coding: a tutorial review," *Proceedings of the IEEE*, vol. 82, núm. 10, págs. 1541–1582, Oct. 1994. 3.3

[49] *Pulse Code Modulation (PCM) of voice frequencies*, ITU Recommendation G.711, October 1988. 3.4

[50] N. S. Jayant, "Digital coding of speech waveforms: Pcm, dpcm, and dm quantizers," *Proceeding of the IEEE*, vol. 62, págs. 611–632, May 1974. 3.4

[51] J. Gibson, "Adaptive prediction in speech diferential encoding systems," *Proceedings of IEEE*, núm. 68, págs. 488–525, 1982. 3.4

[52] *32 kb/s adaptive differential pulse code modulation (ADPCM)*, ITU Recommendation G.721, October 1988. 3.4

[53] *Extensions of Recommendation G.721 ADPCM to 24 and 40 kbits/s for DCME applications*, ITU Recommendation G.723, October 1988. 3.4

[54] *7 kHz audio coding within 64 kbits/s*, ITU Recommendation G.722, October 1988. 3.4

[55] J. G. Proakis y D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms and Applications*. Prentice-Hall, 1996. 3.5.1

[56] W. B. Kleijn y K. K. P. et al, *Speech coding and synthesis*, 1º ed., K. W.B. y P. K.K., Edr. ELSEVIER, 1995. 3.5.1, 3.5.1, 7.4.1

[57] J. D. Markel y A. H. Gray, *Linear Prediction of Speech.* Berlin: Springer Verlag, 1976. 3.5.1

[58] R. Viswanathan y J. Makhoul, "Quantization properties of transmission parameters in linear predictivesystems," *IEEE Transactions Acoustic, Speech, Signal Processing*, vol. ASSP-23, págs. 309–321, 1975. 3.5.1

[59] A. Gray y J. Markel, "Quantization and bit allocation in speech processing," *IEEE Transaction on Acoustic, Speech and Signal Processing*, vol. ASSP-24, págs. 459–473, 1976. 3.5.1

[60] F. Itakura, "Line spectrum representation of linear predictive coefficients of speech signals," *J. Acoust. Soc. Am.*, vol. 57, April 1975. 3.5.1, 7.4.1

[61] F. K. Soong y B. H. Juang, "Line spectral pairs (LSP) and speech data compression," *in Proceedings of IEEE ICASSP*, págs. 1.10.1–1.10.4, 1984. 3.5.1

[62] T. Tremain, "The goverment standard linear predictive coding algorithm: Lpc-10," *Speech Technology*, págs. 40–49, April 1982. 3.5.1

[63] R. J. McAuley y T. F. Quatieri, "Speech analysis-synthesis based on a sinusoidal representation," *IEEE Transactions on Acoustic, Speech and Signal Processing*, vol. ASSP-34, págs. 744–754, 1986. 3.5.2

[64] "Inmarsat-m voiced codec," *Thirty-Sixth Inmarsat Council Meeting*, vol. Appendix I, July 1990. 3.5.2

[65] S. Dimolitsas, F. L. Corcoran, C. Ravishankar, R. S. Skaland, y A. Wong, "Evaluation of voice codec performance for the inmarsat mini-m system," *Proceedings 10th International Digital Satellite Conference*, May 1995. 3.5.2

[66] B. Atal y J. Remde, "A new model of LPC excitation for producing natural-sounding speech at low bit-rates," *in proceedings of IEEE ICASSP*, vol. 7, págs. 614–617, 1982. 3.6.1

[67] *Full rate speech;Transcoding*, ETSI EN 300 961 V8.0.2, 2000. 3.6.1

[68] R. P. Ramachandran y P. Kabal, "Pitch prediction filters in speech coding," *IEEE Transactions on Acoustics, Speech and Signal Processing*, págs. 467–478, 1989. 3.6.1

[69] P. Kroon y B. Atal, "On the use of pitch predictors with high temporal resolution," *IEEE Transactions on Signal Processing*, núm. 39(3), págs. 733–735, 1991. 3.6.1

[70] B. S. Atal y M. R. Schoroeder, "Predictive coding of speech signals and subjective error criteria," *IEEE Transactions on Acoustics, Speech and Signal Processing*, págs. 247–254, June 1979. 3.6.1

[71] 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. 3.6.2

[72] J. Campbell, V. Welch, y T. Tremain, "An expandable error-protected 4800 bps CELP coder (US Federal Standard 4800 bps voice coder)," *in proceedings of IEEE ICASSP*, vol. 2, págs. 735–738, 1989. 3.6.2

[73] J. Chen, "High quality 16 kbit/s speech coding with a one-way delay less than 2 ms," *in proceedings of IEEE ICASSP*, págs. 453–456, 1990. 3.6.2

[74] I. A. Gerson y M. A. Jasiuk, "Vector sum excitation linear prediction (VSELP) speech coding at 8kbps," *in proceedings IEEE ICASSP*, págs. 461–464, 1990. 3.6.2

[75] W. Gardner, P. Jacobs, y C. Lee, *QCELP: a variable rate speech coder for CDMA digital cellular*. Kluwer Academic Publishers, 1993. 3.6.2

[76] C. Laflamme, J. Adoul, H. Su, y S. Morisette, "On reducing computational complexity of codebook search in CELP coders through the use of algebraic codes," *in proceedings of IEEE ICASSP*, págs. 177–180, 1990. 3.6.2

[77] *Enhanced Full rate (EFR) speech transcoding*, ETSI EN 300 726 v8.0.1, 2000. 3.6.2, 4.4, 6.3.4

[78] *AMR speech Codec; Transcoding Functions*, 3GPP TS 26.090. 3.6.2, 4.4, 5.4.2, A.2.1, A.4.1

[79] *TDMA Cellular/PCS - Radio Interface Enhanced Full-Rate Voice Codec*, TIA/EIA/IS-641-A, 1996. 3.6.2

[80] *Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s*, Recommendation ITU-T G.723.1, 1996. 3.6.3

[81] *Coding of Speech at 8 kbit/s using Conjugate-Structure Algebraic-Code-Excited-Linear-Prediction (CS-ACELP)*, Recommendation ITU-T G.729, March 1996. 3.6.3, 4.4, A.2.1

[82] G. I. SOUND, "ilbc - designed for the future," Tech. Rep., October 2004. 3.6.3

[83] S. Andersen, A. Duric, H. Astrom, R. Hagen, W. Kleijn, y J. Linden, "Internet low bit rate codec (ilbc)," *RFC 3951*, 2004. 3.6.3, 4.4, 5.4, 5.4.1, 7.4, A.3.2

[84] L. Besacier, *Automatic Speech Recognition on Mobile Devices and over Communication Networks*. Springer-Verlag, 2008, cap. Speech Coding and Packet Loss Effects on Speech and Speaker Recognition, págs. 27–39. 4.1

[85] *Packet-based multimedia communications systems*, Recommendation ITU-T H.323, June 2006. 4.2.1, 4.4

[86] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, y E. Schooler, "Sip: Session initiation protocol," *RFC 3261*, 2002. 4.2.1

[87] H. Schulzrinne y J. Rosenberg, "A comparison of sip and h.323 for internet telephony," *in Proc. NOSSDAV*, págs. 83–86, 1998. 4.2.1

[88] *General Recommendations on the transmission quality for an entire international telephone connection*, Recommendation ITU-T G.114, 2003. 4.2.2

[89] C. Boulis, M. Ostendorf, E. A. 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. 4.2.2, 6.3.2

[90] S. Basagni, M. Conti, S. Giordano, y I. Stojmenovic, Edr., *Mobile Ad-Hoc Networking*. IEEE-Wiley, 2004. 4.2.2

[91] K. Wesolowsky, *Mobile Communication Systems*. Wiley, 2002. 4.2.2

[92] A. H. Nour-Eldin, H. Tolba, y D. O'Shaughnessy, "Automatic recognition of bluetooth speech in 802.11 interference and the effectiveness of insertion-based compensation techniques," *in Proceedings of IEEE ICASSP*, 2004. 4.2.2

[93] J. Bolot, "End-to-end packet delay and loss behavior in the internet," *ACM Sigcomm*, págs. 289–298, 1993. 4.2.3, 4.5

[94] 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. 4.2.3, 4.2.3, 4.2.3, 4.5

[95] 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. 4.2.3, 4.5, A.2.1

[96] N. F. Maxemchuk y S. Lo, "Measurement and interpretation of voice traffic on the internet," *in proceedings of ICC*, 1997. 4.2.3

[97] J. Bolot, "Adaptive FEC-based error control for Internet telephony," *in proceedings of IEEE INFOCOM*, vol. 3, págs. 1453–1460, 1999. 4.2.3

[98] 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. 4.2.3, 4.2.3

[99] 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. 4.2.3, 4.2.3, 4.5, A.2.1

[100] K. Salamatian y S. Vaton, "Hidden markov modeling for network communication channels,," *Proc. ACM SIGMETRICS*, June 2001. 4.2.3

[101] B. Milner y A. James, "An analysis of packet loss models for distributed speech recognition," *in Proceedings of INTERSPEECH-ICSLP*, 2004. 4.2.3, 4.2.3

[102] *QoS measurements for Voice over IP. Technical Report*, ETSI TIPHON TS 101-329-5, 2000. 4.2.3

[103] F. Metze, J. McDonough, y H. Soltau, "Speech recognition over netmeeting connection," *in proceedings of Eurospeech*, 2001. 4.3

[104] *Codec for circuit switched multimedia telephony service; General description*, 3GPP TS 26.110. 4.4

[105] *Packet switched conversational multimedia application; Default codecs*, 3GPP TS 26.235. 4.4

[106] J. Sjoberg, M. Westerlund, A. Lakaniemi, y Q. Xie, "Real-time transport protocol (rtp) payload format and file storage format for the adaptive multi-rate (amr) and adaptive multi-rate wideband (amr-wb) audio codecs," *RFC 3267*, 2002. 4.4

[107] *G.729 based Embedded Variable bit-rate coder: An 8-32 kbit/s scalable widebandcoder bitstream interoperable with G.729*, Recommendation ITU-T G.729.1, 2006. 4.4, A.2.1

[108] R. 3551, "Rtp profile for audio and video conferences with minimal control," *IETF*, July 2003. 4.4

[109] A. Sollaud, "Rtp payload format for the g.729.1 audio codec," *RFC 4749*, October 2006. 4.4

[110] A. Duric y S. Andersen, "Real-time transport protocol (rtp) payload format for internet low bit ratecodec (ilbc) speech," *RFC 3952*, December 2004. 4.4

[111] B. W. Wah y B. Sat, "Analysis and evaluation of skype and google talk," Department of Electrical and Computer Engineering and the Coordinated ScienceLaboratory," Technical Report, January 2006. 4.4

[112] S. V. Andersen, W. B. Kleijn, R. Hagen, J. Linden, y M.Ñ. M. J. Skoglund, "Ilbc - a linear predictive coder with robustness to packet losses," *IEEE 2002 Workshop on Speech Coding*, December 2002. 4.4, 5.2, A.2.2

[113] *PacketCable^{TM} 1.5 Specifications Audio/Video Codecs*, CableLabs, 2005. Disponible en linea: http://www.cablelabs.com/specifications/PKT-SP-CODEC1.5-I02-070412.pdf 4.4

[114] V. Paxson, "End-to-end internet packet dynamics," *Procedings of SIGCOMM*, 1997. 4.5

[115] *HTK3 - Hidden Markov Model Toolkit*, Cambridge University, 2004. Disponible en linea: http://htk.eng.cam.ac.uk 4.6.2

[116] *Aurora project database 2, 3 & 4*. Disponible en linea: http://www.elda.org/rubrique18.html 4.6.3

[117] D. Pearce y H.-G. Hirsch, "The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions," *Proceedings of ICSLP*, 2000. 4.6.3

[118] *Transmission Performance Characteristics of Pulse Code Modulation Channels*, Recommendation ITU G.712. 4.6.3

[119] C. Perkins, O. Hodson, y V. Hardman, "A survey of packet-loss recovery techniques for streaming audio," *IEEE Network Magazine*, 1998. 5.1, 6.3.2

[120] W. Jiang y H. Schulzrinne, "Speech recognition performance as an effective perceived quality predictor," *In proceedings IEEE International Workshop on Quality of Service*, págs. 269–275, 2002. 5.1

[121] W. M. Liu, K. A. Jellyman, J. S. D. Mason, y N. W. D. Evans, "Assessment of objective quality measures for speech intelligibility estimation," *in proceedings of IEEE ICASSP*, vol. 1, 2006. 5.1

[122] A. M. Gómez, A. M. Peinado, V. Sánchez, y A. J. Rubio, "Recognition of coded speech transmitted over wireless channels," *IEEE Transactions on Wireless Comunications*, 2006. 5.2, 6.3.3, 6.3.4, 6.4.6, 7.1, 7.4.1, A.4.5, A.5.2

[123] M. Serizawa y H. Ito, "A packet loss recovery method using packet arrived behind the playout time for celp decoding," *in proceedings of IEEE ICASSP*, vol. 5, págs. 2555–2562, 2004. 5.2, 5.4.3

[124] M. Chibani, R. Lefebvre, y P. Gournay, "Fast recovery for a CELP-like speech codec after a frame erasure," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, págs. 2485–2495, 2007. 5.2, 5.4.3

[125] V. Eksler y M. Jelinek, "Transition mode coding for source controlled celp codecs," *in proceedings of IEEE ICASSP*, págs. 4001–4004, 2008. 5.2, 5.4.3

[126] A. M. Gómez, A. M. Peinado, V. Sánchez, y A. J. Rubio, "Combining media-specific FEC and error concealment for robust distributed speech recognition over loss-prone packet channels," *IEEE Transactions on Multimedia*, 2006. 5.2

[127] C. Xydeas y F. Zafeiropoulos, "Model-based packet loss concealment for amr coders," *in proceedings of IEEE ICASSP*, vol. 1, págs. 112–115, 2003. 5.2

[128] H. Ehara y K. Yoshida, "Decoder initializing technique for improving frame-erasure resilience of a celp speech codec," *IEEE Transactions on Multimedia*, vol. 10, págs. 549–553, 2008. 5.2, 5.4.3

[129] *Methods for subjective determination of transmission quality*, Recommendation ITU P.800, 1996. 5.3.1

[130] *Subjective performance assessment of telephone-band and wideband digital codecs*, Recommendation ITU P.830, 1996. 5.3.1

[131] *Method for subjective assessment of intermediate quality level of coding systems*, Recommendation ITU-R BS.1534-1, 2003. 5.3.1

[132] *Methods for the subjective assessment of small impairments in audio systems including multichannel sound systems*, Recommendation ITU-R BS.1116-1, 1997. 5.3.1

[133] P. Vary y R. Martin, *Digital Speech Transmission: Enhancement, Coding and Error Concealment*. Wiley, 2006. 5.3.2

[134] *Objective quality measurement of telephone-band (300-3400 Hz) speech codecs*, Recommendation ITU P.862, 1998. 5.3.2

[135] *Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs*, Recommendation ITU P.862, 2001. 5.3.2, A.3

[136] *Mapping function for transforming P.862 raw result scores to MOS-LQO*, Recommendation ITU P.862.1, 2003. 5.3.2

[137] R. Salami, C. Laflamme, J.-P. Adoul, y D. Massaloux, "A toll quality 8 kb/s speech codec for the personal communications sistems (PCS)," *IEEE Transactions on Vehicular Technology*, vol. 43, págs. 808–816, 1994. 5.4.2

[138] L. Lamel, R. Kassel, y S. Seneff, "Speech database development: design and analysis of the acoustic-phonetic corpus," *Proc. Speech Recognition Workshop (DARPA)*, págs. 100–110, 1986. 5.4.3, 5.5

[139] J. S. Garafolo, "The structure and format of the DARPA TIMIT CD-ROM prototype," *Documentation of DARPA TIMIT*. 5.4.3

[140] S. Pennock, "Accuracy of the perceptual evaluation of speech quality (PESQ) algorithm," *in Proc. MESAQIN*, 2002. 5.4.3, A.3

[141] T. Vaillancourt, M. Jelinek, R. Salami, y R. Lefebvre, "Efficient frame erasure concealment in predictive speech codecs using pulse resynchronisation," *in proceedings of IEEE ICASSP*, vol. IV, págs. 1113–1116, 2007. 5.4.3

[142] R. Lefebvre, P. Gournay, y R. Salami, "A study of design compromises for speech coders in packet networks," *in proceedings of IEEE ICASSP*, vol. 1, págs. 265–268, 2004. 5.4.3, 6.3.4, A.3.2

[143] S. Singhal y B. Atal, "Amplitude optimization and pitch predicition in multipulse coders," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, págs. 317–327, 1989. 5.5, 5.5, A.3.3

[144] A. Moreno, D. Poch, A. Bonafonte, E. Lleida, J. Llisterri, J. Mariño, y C.Ñadeu, "ALBAYZIN speech database: Design of the phonetic corpus," *in proceedings of Eurospeech*, vol. 1, págs. 175–178, 1993. 5.5.3, A.3.3

[145] 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. 6.3.1

[146] 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. 6.3.1, 7.1

[147] 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. 6.3.1, 6.6.2, 6.6.3, 6.6.3, 6.6.3, A.4.4

[148] V. Hardman, "Reliable audio for use over the internet," *in proceedings of INET*, 1995. 6.3.2

[149] 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. 6.3.2

[150] R. M. Warren, *Auditory Perception.* Pergamon Press Inc., 1982. 6.3.2

[151] B. Milner y S. Semnani, "Robust speech recognition over IP networks," *in proceedings of IEEE ICASSP*, vol. 3, págs. 1791–1794, 2000. 6.3.3

[152] A. M. Gómez, A. M. Peinado, V. Sánchez, y A. J. Rubio, "Mitigation of channel errors in EFR-based speech recognition," *in proceedings of IEEE ICASSP*, vol. 1, págs. 1021–4, 2004. 6.3.3, 6.3.3, 6.3.4

[153] 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. 6.3.3, 6.3.3

[154] A. James y B. Milner, "Towards improving the robustness of distributed speech recognition in packet loss," *Speech Communication*, vol. 48, págs. 1402–1421, 2006. 6.3.3, 6.6.3, A.2.1

[155] P. Moreno, "Speech recognition in noisy environments," Tesis doctoral, Carnegie Mellon University, 1996. 6.3.3, A.4.5

[156] A. Peinado, V. Sanchez, J. Perez-Cordoba, y A. Torre, "HMM-based channel error mitigation and its application to distributed speech recognition," *Speech Communication*, núm. 41, págs. 549–561, 2003. 6.3.4, 6.4.2, 6.4.3, A.4.2, A.4.2, A.4.2

[157] 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. 6.3.4, 7.4.4

[158] M. Skoglund y P. Hedelin, "Vector quantization over a noisy channel using soft decision," *in Proceedings of IEEE ICASSP*, 1994. 6.4.1

[159] 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. 6.4.1

[160] T. Fingscheidt y P. Vary, "Softbit speech decoding: a new approach to error concealment," *IEEE Transactions on Speech and Audio Processing*, vol. 9, págs. 240–251, 2001. 6.4.2

[161] A. M. Peinado, A. M. Gómez, V. Sánchez, y A. J. 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. 6.4.3, A.4.2

[162] V. Ion y R. Haeb-Umbach, "Uncertainty decoding for distributed speech recognition over error-prone networks," *Speech Communication*, vol. 48, págs. 1435–1446, 2006. 6.4.3, 6.4.6, A.4.3

[163] V. Ion y R. Haeb-Umbach, "A novel uncertainty decoding rule with applications to transmission error robust speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, págs. 1047–1060, 2008. 6.4.3, 6.4.6, 6.5.4, 7.5, A.5.3

[164] 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. 6.4.3

[165] R. Haeb-Umbach y V. Ion, "Soft features for improved distributed speech recognition over wireless networks," *in proceedings of INTERSPEECH-ICSLP*, 2004. 6.4.3, 6.5

[166] A. Morris, M. Cooke, y P. Green, "Some solutions to the missing feature problem in data classification, with application to noise robust ASR," *in proceedings of IEEE ICASSP*, vol. 2, págs. 737–740, 1998. 6.5.1

[167] A. Morris, J. Barker, y H. Boulard, "From missing data to maybe useful data: Soft data modelling for noise robust ASR," *in Proceedings of Workshop on Innovation in Speech Processing (WISP)*, 2001. 6.5.1

[168] J. Droppo, A. Acero, y L. Deng, "Uncertainty decoding with splice for noise robust speech recognition," *in Proceedings of IEEE ICASSP*, vol. I, págs. 57–60, 2002. 6.5.2

[169] A. M. Peinado, A. M. Gómez, V. Sánchez, J. L. Pérez-Cordoba, y A. J. Rubio, "An integrated solution for error concealment in dsr systems over wireless channels," *in proceedings of INTERSPEECH-ICSLP*, págs. 1093–1096, 2006. 6.5.3, A.4.3

[170] A. Potamianos y V. Weerackody, "Soft-feature decoding for speech recognition over wireless channels," *in Proceedings of IEEE ICASSP*, págs. 269–272, 2001. 6.6.2, 6.6.2, 6.6.3, 6.6.4, A.4.4

[171] 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. 6.6.2, 6.6.3

[172] T. Endo, S. Kuroiwa, y S.Ñakamura, "Missing feature theory applied to robust speech recognition over ip network," *in proceedings of INTERSPEECH-ICSLP*, págs. 3081–3084, 2003. 6.6.3

[173] A. Cardenal-Lopez, C. García-Mateo, y L. Docio-Fernandez, "Weighted viterbi decoding strategies for distributed speech recognition over ip networks," *Speech Communication*, vol. 48, págs. 1422–1434, 2006. 6.6.3, 7.5, A.5.3

[174] 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. 6.6.3

[175] 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. 6.6.3

[176] J. M. Huerta y R. M. Stern, "Speech recognition from GSM codec parameters," *in proceedings of INTERSPEECH-ICSLP*, vol. 4, págs. 1463–1466, 1998. 7.1

[177] J. M. Huerta, "Speech recognition in mobile environments," Tesis doctoral, Carnegie Mellon University, 2000. 7.1

[178] A. Gallardo-Antolin, F. D. de Maria, y F. Valverde-Albacete, "Recognition from GSM digital speech," *in proceedings of INTERSPEECH-ICSLP*, 1998. 7.1

[179] 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. 7.1

[180] C. Pelaez-Moreno, A. Gallardo-Antolin, y F. D. 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. 7.1

[181] C. Pelaez-Moreno, "Reconocimiento de habla mediante transparametrización: Una alternativa robusta para entornos móviles e IP," Tesis doctoral, Universidad Carlos III, 2002. 7.1

[182] A. Kataoka, T. Moriya, y S. Hayashi, "An 8-kb/s conjugate structure CELP (CS-CELP) speech coder," *IEEE Transactions on Speech and Audio Processing*, vol. 4, núm. 6, págs. 401–411, November 1996. 7.3.3

[183] W. B. Kleijn, "Enhancement of coded speech by constrained optimization," *IEEE 2002 Workshop on Speech Coding*, págs. 163–165, December 2002. 7.4

[184] J. L. Carmona, A. M. Peinado, J. L. Pérez-Córdoba, V. Sánchez, y A. M. Gómez, "Rendimiento perceptual y reconocimiento con codificadores voip sobre redes de paquetes," *Actas de las IV Jornadas en Tecnologías del Habla*, págs. 249–254, 2006. 8.2, 9.2

[185] J. L. Carmona, A. M. Peinado, J. L. Pérez-Córdoba, A. M. Gómez, y J. A. González, "A scalable coding scheme based on interframe dependency limitation," *in proceedings of IEEE ICASSP*, 2008. 8.2, 9.2

[186] A. M. Gómez, J. L. Carmona, A. M. Peinado, y V. Sánchez, "Intelligibility evaluation of ramsey-derived interleavers for internet voice streaming with the ilbc codec," *in proceedings of INTERSPEECH-ICSLP*, págs. 707–710, 2008. 8.2, 9.2

[187] A. M. Gómez, J. L. Carmona, A. M. Peinado, y V. Sánchez, "A multipulse-based forward error correction technique for robust CELP-coded speech transmission over erasure channels," *enviado a IEEE Transactions on Audio, Speech, and Language Processing*, 2009. 8.2, 9.2

[188] J. L. Carmona, A. M. Peinado, J. L. Pérez-Córdoba, y A. M. Gómez, "MMSE-based packet loss concealment for CELP-coded speech recognition," *en segunda revisión de IEEE Transactions on Audio, Speech, and Language Processing*, 2009. 8.2, 9.2

[189] A. M. Gómez, A. M. Peinado, V. Sánchez, J. L. Carmona, y A. J. Rubio, "Interleaving and MMSE estimation with VQ replicas for distributed speech recognition over lossy packet networks," *in proceedings of INTERSPEECH-ICSLP*, 2006. 8.2, 9.2

[190] A. M. Gómez, A. M. Peinado, V. Sánchez, y J. L. Carmona, "A robust scheme for distributed speech recognition over loss-prone packet channels," *Speech Communication*, vol. 51, págs. 390–400, 2009. 8.2, 9.2

[191] J. L. Carmona, A. M. Peinado, J. L. Pérez-Córdoba, V. Sánchez, y A. M. Gómez, "ilbc-based transparametrization: A real alternative to dsr for speech recognition over packet networks," *in proceedings of IEEE ICASSP*, vol. 4, págs. 961–964, 2007. 8.2, 9.2

[192] *Mandatory Speech Codec speech processing functions; AMR Speech Codec; Error concealment of lost frames*, 3GPP TS 26.091. A.4.1