Supplementary Material

Latency-dependent filtering and compact representation of the complete auditory pathway response

Angel de la Torre[1], Joaquin T. Valderrama[2,3,4], Jose C. Segura[1], and Isaac M. Alvarez[1]

[1]*Department of Signal Theory, Telematics and Communications, University of Granada, Granada, Spain.*
[2]*National Acoustic Laboratories, Sydney, Australia.*
[3]*Department of Linguistics, Macquarie University, Sydney, Australia.*
E-mail: atv@ugr.es (A. de la Torre); joaquin.valderrama@nal.gov.au (J.T. Valderrama); segura@ugr.es (J.C. Segura); isamaru@ugr.es (I.M. Alvarez)

Contents

1 Matrix implementation of low-pass filtering and down-sampling

This section describes in detail matrix formulation of low-pass filtering and down-sampling, including the equations and some examples. The information in this section could be useful for those readers not familiar with matrix description of signal processing operations.

1.1 Matrix formulation of filtering

Let $x(j)$ (with $j = 0, \ldots, J-1$) be a digital signal, and $h(j)$ the impulsive response of a digital filter. The filtered signal $x_h(j)$ is obtained as the convolution of the input signal and the impulsive response:

$$x_h(j) = h(j) * x(j) = \sum_{j'} h(j')x(j-j') \tag{1}$$

where the symbol '$*$' represents convolution. According to this equation, each sample of the filtered signal is obtained as a linear combination of several samples in the original signal, according to the values of the impulsive response. The samples of the signal $x(t)$ can be represented as a J-component column vector:

$$\mathbf{x} = \begin{pmatrix} x(0) \\ x(1) \\ \vdots \\ x(J-1) \end{pmatrix} \tag{2}$$

and filtering can be represented as the matrix product:

$$\mathbf{x}_h = H\mathbf{x} \tag{3}$$

where $\mathbf{x}_h$ is a J-component column vector representing the filtered signal, and H is the $J \times J$ convolution matrix, with elements $H(j_1, j_2) = h(j_1 - j_2)$:

$$\begin{pmatrix} x_h(0) \\ x_h(1) \\ \vdots \\ x_h(J-1) \end{pmatrix} = \begin{pmatrix} h_0 & h_{-1} & h_{-2} & h_{-3} & \ldots & h_{-(J-1)} \\ h_1 & h_0 & h_{-1} & h_{-2} & \ldots & h_{-(J-2)} \\ h_2 & h_1 & h_0 & h_{-1} & \ldots & h_{-(J-3)} \\ h_3 & h_2 & h_1 & h_0 & \ldots & h_{-(J-4)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ h_{J-1} & h_{J-2} & h_{J-3} & h_{J-4} & \ldots & h_0 \end{pmatrix} \begin{pmatrix} x(0) \\ x(1) \\ \vdots \\ x(J-1) \end{pmatrix} \tag{4}$$

Note that the sum in equation (1) can be limited to those terms j' with not null elements $h(j')$. Similarly, in the convolution matrix H, for those j for which $h(j)$ is null, the elements in the corresponding diagonal are null. If $x(j)$ contains J_x non null elements and $h(j)$ contains J_h non null elements, the filtered signal contains $J_x + J_h - 1$ non null elements, and therefore, in order to estimate the complete convolution in the matrix form, the signal has to be appropriately zero-padded with $J = J_x + J_h - 1$ (otherwise a truncated instead of a complete convolution is estimated).

1.2 Example of low pass filtering with convolution and matrix product

An example showing the equivalence of convolution with equation (1) and with matrix product in equation (3) has been implemented. For this example, a random signal $x(j)$ has been generated (460 samples of white noise low-pass filtered in the band $[0 - 0.1 f_s]$ being f_s the sampling rate; the signal has been zero padded up to 500 samples: $J = 500$). A low-pass filter has been prepared, and the signal has been filtered with a discrete time convolution according to equation (1).

Figure 1 shows the impulsive response $h(j)$ of the low-pass filter used in this example (left panel), as well as the original and the filtered signals, $x(j)$ and $x_h(j)$, respectively (right panel). The low-pass filter has been designed for a cut-off frequency $0.04 f_s$. For the matrix implementation of the filtering process, a convolution matrix has been prepared using the impulsive response of the low-pass filter $h(j)$, according to equation (4). The matrix is represented in figure 2 (the values of the coefficients are represented, for each row and column of the matrix, with a colormap). As can be seen, all the elements in each direct diagonal are identical (which corresponds to the definition of a Toeplitz matrix and is associated to the time invariance of the filter).

Figure 3 represents three rows of the convolution matrix in the left panel. As can be observed, the rows are identical except for the delay, which again is in accordance with the time invariance of the filter. The right panel compares the filtered signal obtained with the convolution (in red) and with the matrix product (in black). As expected, both filtered signals are identical, because both procedures are mathematically equivalent.

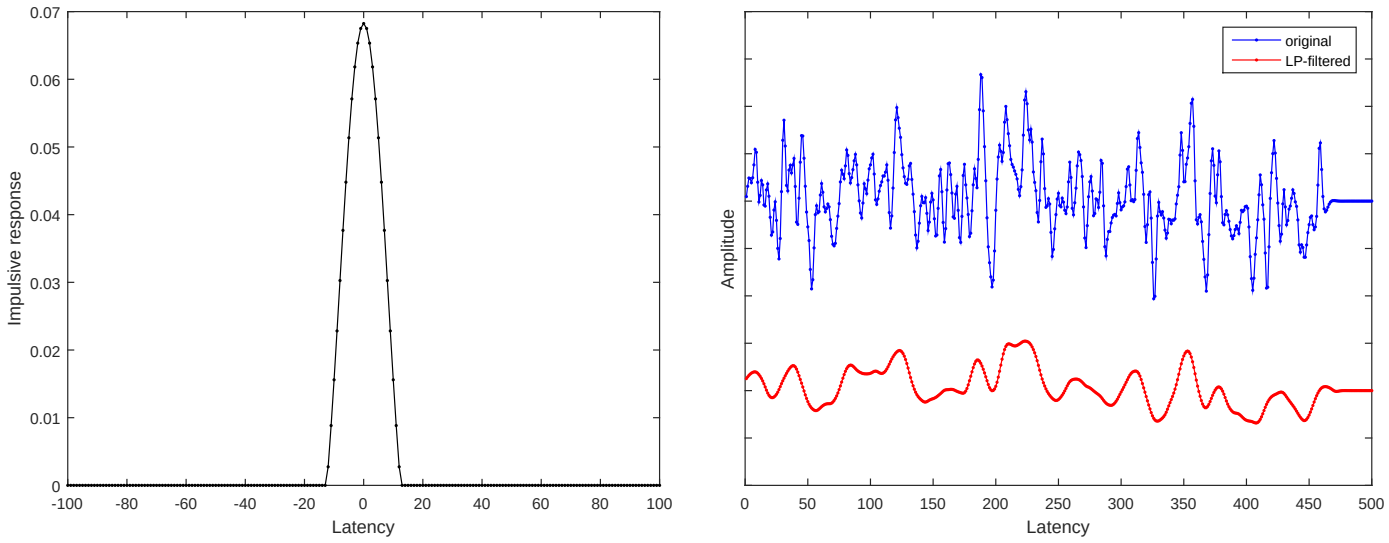

Figure 1: Left panel: impulsive response $h(j)$ of the low-pass filter used in the example. Right panel: original signal $x(j)$ (in blue) and low-pass filtered signal $x_h(j)$ (in red).

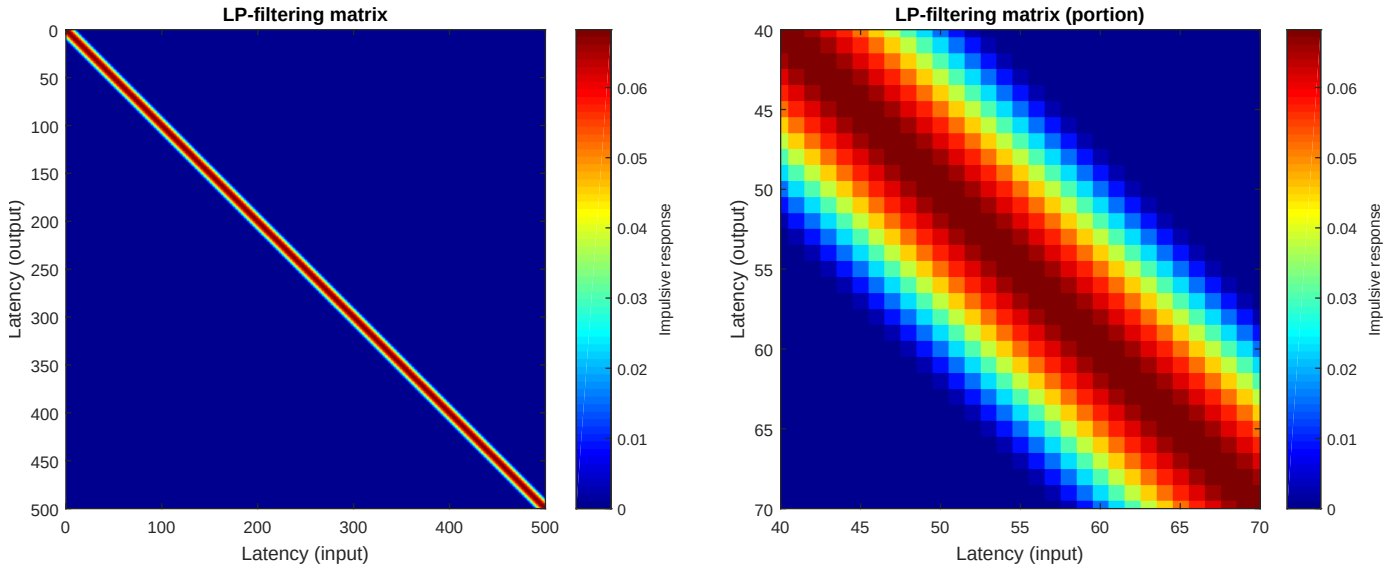

Figure 2: Matrix providing the low-pass filtering for the impulsive response $h(j)$. Left panel: the complete $J \times J$ matrix. Right panel: a portion of the matrix including the intersection of 30 rows and columns.

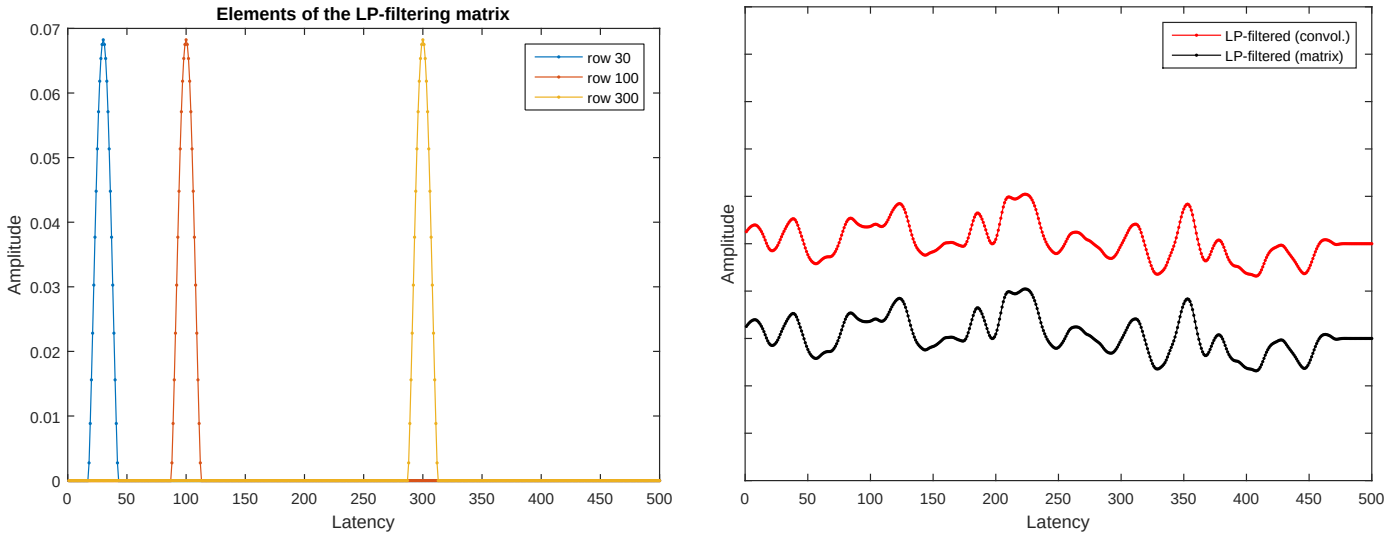

Figure 3: In the left panel, elements of the convolution matrix corresponding to the rows 30, 100 and 300. In the right panel, comparison of the filtered signals $x_h(j)$ obtained with convolution (red) and matrix product (black).

1.3 Matrix formulation of filtering and down-sampling

According to the sampling theorem, a band-limited signal can be down-sampled without information loss if the new sampling rate is at least twice the maximum frequency component. Down-sampling with a factor q is performed by keeping 1 sample and discarding $q-1$ samples from every q samples:

$$x_r(j_r) = x_h(j_r \cdot q) \tag{5}$$

where the down-sampled signal $x_r(j_r)$ includes J_r samples ($j_r = 0, \ldots, J_r - 1$), with $J_r = J/q$. The down-sampling procedure can be represented with a matrix operator. The down-sampling matrix for a factor q would be a $J_r \times J$ matrix obtained from an identity matrix where 1 row is preserved and $q-1$ are discarded from every q rows. For example, the down-sampling matrix for a down-sampling factor $q = 4$ would be:

$$H_{ds} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 1 \end{pmatrix} \tag{6}$$

And the matrix operation providing down-sampling would be:

$$\mathbf{x}_r = H_{ds}\mathbf{x}_h \tag{7}$$

Since both filtering and down-sampling are described as matrix operations, both procedures can be compacted in just one matrix operation:

$$\mathbf{x}_r = H_{ds}\mathbf{x}_h = H_{ds}H\mathbf{x} = H_r\mathbf{x} \qquad \text{where} \quad H_r = H_{ds}H \tag{8}$$

and taking into account the form of the down-sampling matrix, the filtering and down-sampling matrix H_r is obtained by preserving 1 row and discarding $q-1$ rows from every q rows in the filtering matrix H.

1.4 Example: low-pass filtering and down-sampling with a matrix product

The low-pass filtering and down-sampling implemented as a matrix operation has been applied to the previous example. In this example, the filtered signal has been down-sampled in a factor $q = 4$, and therefore the low-pass filtering and down-sampling matrix H_r has been obtained by removing 3 from every 4 rows in the matrix H. With this down-sampling factor, the length of the signal decreases from $J = 500$ to $J_r = 125$, and H_r is a 125×500 matrix.

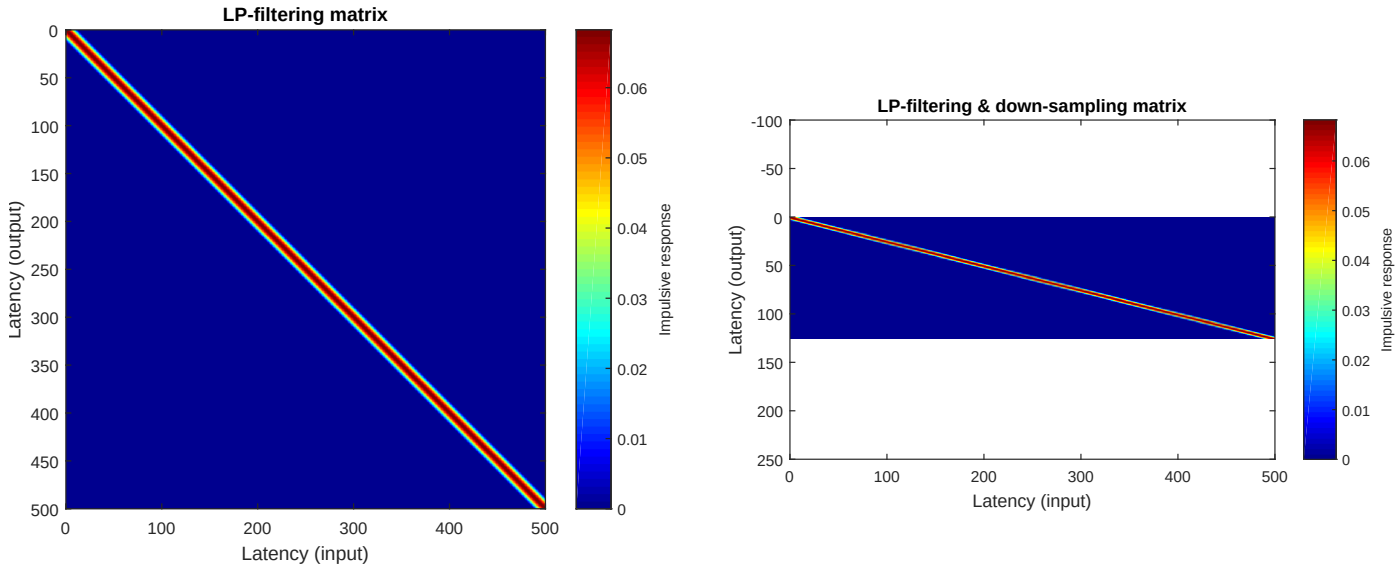

Figure 4: Left panel: matrix providing low-pass filtering. Right panel: matrix providing low-pass filtering & down-sampling.

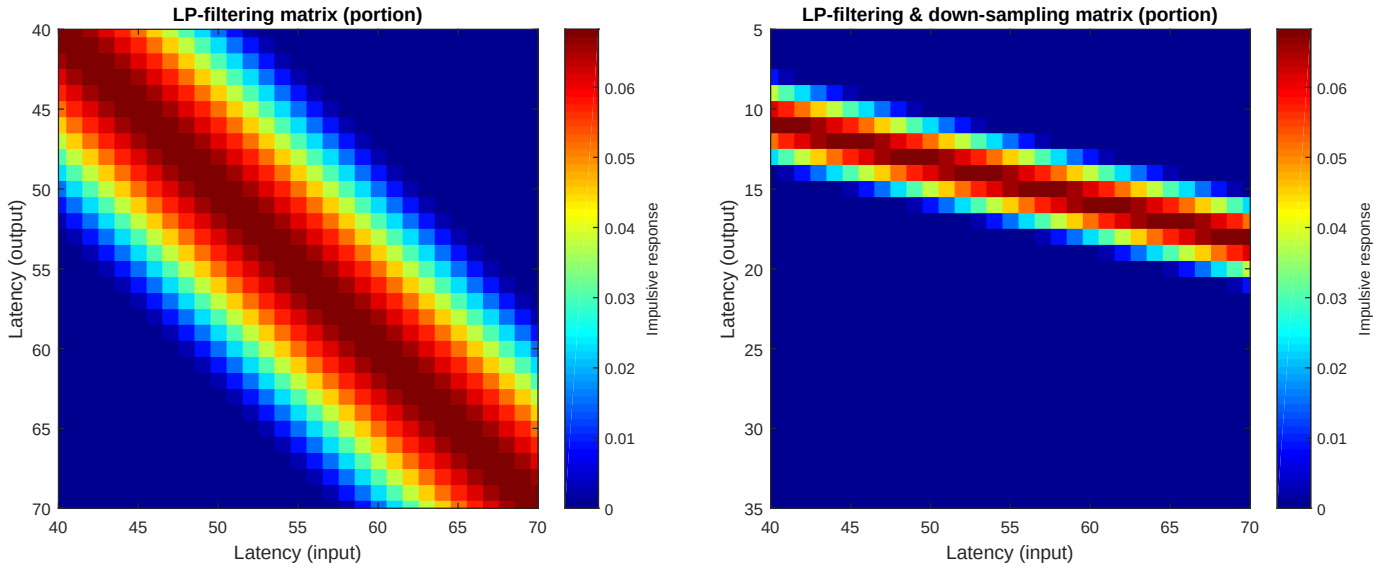

Figure 5: Left panel: matrix providing low-pass filtering. Right panel: matrix providing low-pass filtering & down-sampling. Detail for the intersection of 30 rows and columns.

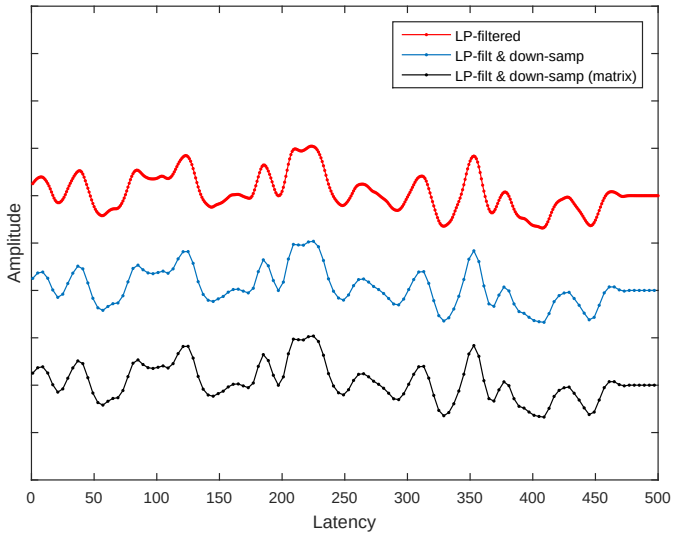

Figure 6: Comparison of the signals $x_h(j)$ (filtered) and $x_r(j_r)$ (filtered and down-sampled), where $x_r(j_r)$ has been estimated: (a) by down-sampling the filtered signal $x_h(j)$ (blue), and (b) by applying the matrix H_r to the original signal $x(j)$ (black).

Figure 4 provides a comparison of the matrices H and H_r. Figure 5 is a detail of the previous one for a 30×30 portion of the respective matrices. As can be seen, the matrix H_r includes 1 of every 4 rows from matrix H.

Figure 6 compares the low-pass filtered and down-sampled signals obtained by (a) down-sampling the filtered signal, and (b) the matrix operation $\mathbf{x}_r = H_r\mathbf{x}$. Again, as expected, the resulting low-pass filtered and down-sampled signals obtained with both procedures are identical.

2 Matrix implementation of the latency-dependent low-pass filtering and down-sampling

This section describes the matrix implementation of a latency-dependent low-pass filtering and down-sampling procedure, including a detailed description and some examples.

2.1 Latency-dependent filtering

A latency-dependent filtering implies that the impulsive response changes with the latency, and therefore, that the filter is not time-invariant (i.e. it is not a conventional filter). A latency-dependent filtered signal is obtained as a convolution where the impulsive response of the filter changes with the latency:

$$x_{ld}(j) = \sum_{j'} h_{ld}(j, j')x(j - j') \tag{9}$$

where $h_{ld}(j, j')$ is the impulsive response at latency j and the subindex ld stands for "latency-dependent".

A latency-dependent filtering can easily be implemented as a matrix operation. While in time-invariant filtering all the rows of the filtering matrix are identical except for the delay (the matrix is Toeplitz), a latency-dependent filtering can be implemented by using different impulsive responses in the different rows of the matrix:

$$H_{ld} = \begin{pmatrix} h_{ld}(0,0) & h_{ld}(0,-1) & h_{ld}(0,-2) & h_{ld}(0,-3) & \dots & h_{ld}(0,-(J-1)) \\ h_{ld}(1,1) & h_{ld}(1,0) & h_{ld}(1,-1) & h_{ld}(1,-2) & \dots & h_{ld}(1,-(J-2)) \\ h_{ld}(2,2) & h_{ld}(2,1) & h_{ld}(2,0) & h_{ld}(2,-1) & \dots & h_{ld}(2,-(J-3)) \\ h_{ld}(3,3) & h_{ld}(3,2) & h_{ld}(3,1) & h_{ld}(3,0) & \dots & h_{ld}(3,-(J-4)) \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ h_{ld}(J-1,J-1) & h_{ld}(J-1,J-2) & h_{ld}(J-1,J-3) & h_{ld}(J-1,J-4) & \dots & h_{ld}(J-1,0) \end{pmatrix} \tag{10}$$

and since each row corresponds to a specific latency of the filtered signal, the filtered response at latency j is obtained with the filter described with the specific impulsive response $h_{ld}(j, j')$ for this latency.

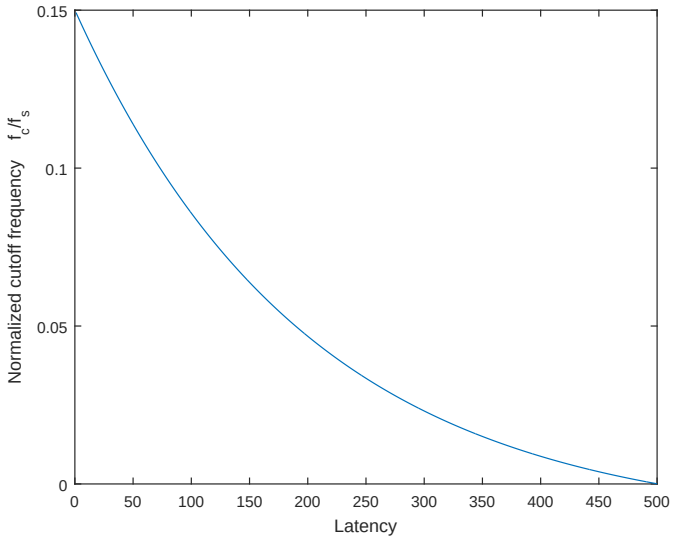

Figure 7: Latency-dependent cut-off frequency applied in the example.

2.2 Example of latency-dependent filtering

In this example, a latency-dependent low-pass filter has been designed using a cut-off frequency changing from $0.15f_s$ (at low latency) to $0f_s$ (at a latency of 500 samples). The evolution of the cut-off frequency with the latency proposed for this example has been represented in figure 7.

Figure 8 compares the low-pass filtering matrices for the time-invariant version (left panel) and the latency-dependent version (right panel). In the latency-dependent low-pass filter, the cut-off frequency decreases (and the response is wider) as the latency increases, according to the plot in figure 7. Additionally, due to the normalization of the response, as the latency increases the amplitude of the impulsive response also decreases.

Figure 9 shows a detail of the low-pass filtering matrices. Portions of the intersection of 30 rows and 30 columns are represented in the figure, in the left panels corresponding to the time-invariant low-pass filtering matrix, and in the right panels corresponding to the latency-dependent low-pass filtering matrix. The upper panels corresponds to small latency (range 40-70) and the lower panels to larger latency (range 250-280). While both panels look similar in the case of the time invariant filtering, in the case of the latency-dependent filtering the response is narrower at low latency and wider at large latency.

Figure 10 represents three rows of the latency-dependent low-pass filtering matrix in the left panel. The represented rows correspond to latency 30, 100 and 300 respectively. These plots illustrate how the impulsive response changes with latency in the proposed latency-dependent filtering. The right panel compares the original signal, the filtered signal and the latency-dependent filtered signal. Due to the evolution of the cut-off frequency in the latency-dependent filter, at low latency (below sample 200) the frequency content is greater in the latency-dependent filtered signal, similar at medium latency (around samples 200-250) and smaller at large latency (above sample 250), compared with the conventionally filtered signal.

2.3 Latency-dependent filtering and down-sampling

A latency-dependent low-pass filtering allows down-sampling without information loss if the sampling rate is at least twice the maximum frequency component, according to the sampling theorem. Since the bandwidth depends on the cut-off frequency and it depends on the latency, latency-dependent filtering suggest a latency-dependent down-sampling, where the sampling rate changes locally according to the local cut-off frequency.

The latency-dependent down-sampling can be implemented by selecting the new samples with a latency-dependent down-sampling factor q. Similarly, the latency-dependent down-sampling can also be implemented as a matrix operation by defining a reduced matrix H_r in which the rows are non-uniformly selected from the matrix H_{ld} according to the local cut-off frequency.

2.4 Example: matrix implementation of latency-dependent filtering and down-sampling

A latency-dependent down-sampling has been applied to the matrix H_{ld} in the previous example. The local sampling rate was at least 5 times the local cut-off frequency (the local down-sampling factor q was smaller than 0.2 times the inverse of the normalized

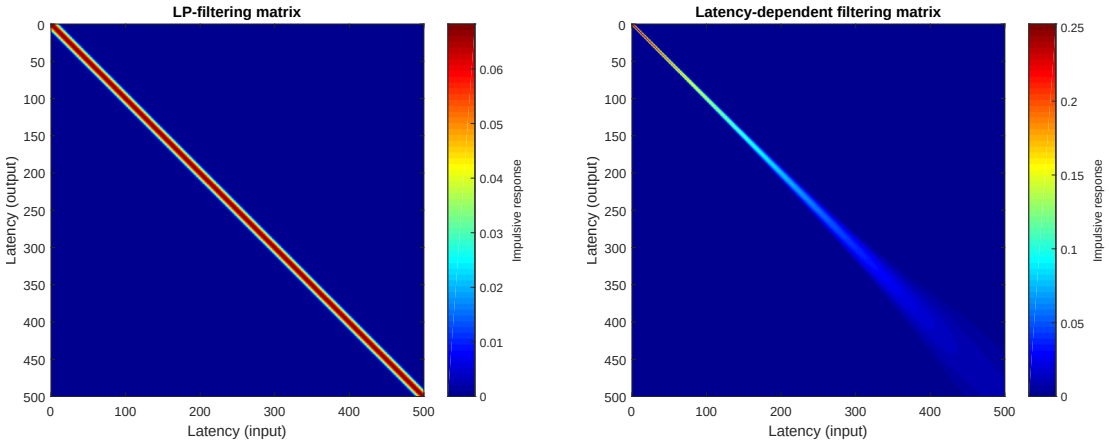

Figure 8: Matrix implementation of the latency-dependent low-pass filtering. Left side: standard filtering (time-invariant). Right side: latency-dependent filtering (not time-invariant). The cut-off frequency decreases with the latency in the latency-dependent filtering.

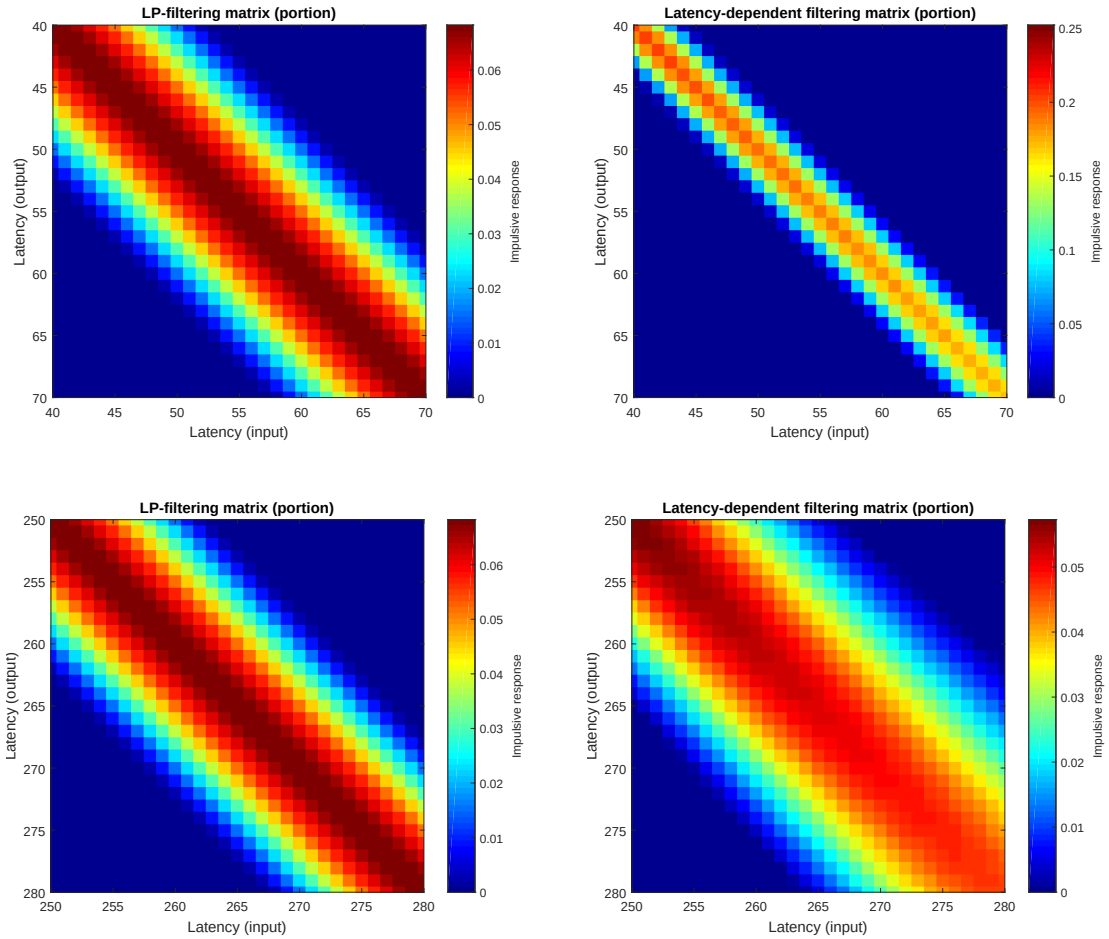

Figure 9: Detail of the matrices implementing the latency-dependent low-pass filtering. Left side: standard filtering (time-invariant). Right side: latency-dependent filtering (not time-invariant). The 30×30 portions of the matrices corresponds to small latency (upper panels) and large latency (lower panels). The cut-off frequency decreases with the latency in the latency-dependent filtering.

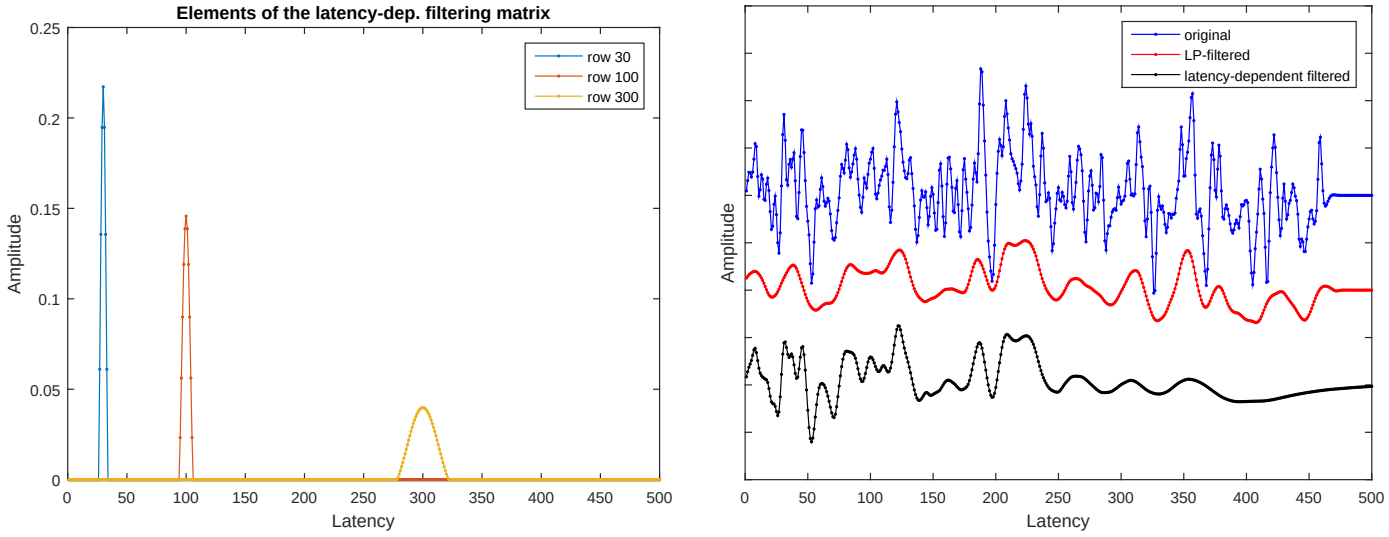

Figure 10: Left panel: elements of the latency-dependent low-pass filtering matrix, corresponding to the rows 30, 100 and 300. As can be observed, the cut-off frequency decreases as the latency increases. Right panel: comparison of the original signal, the filtered signal an the latency-dependent filtered signal. As can be observed, for the latency-dependent filtered signal, the cut-off frequency is greater at small latency (below 200), similar at medium latency (around 200-250), and smaller at large latency (above 250).

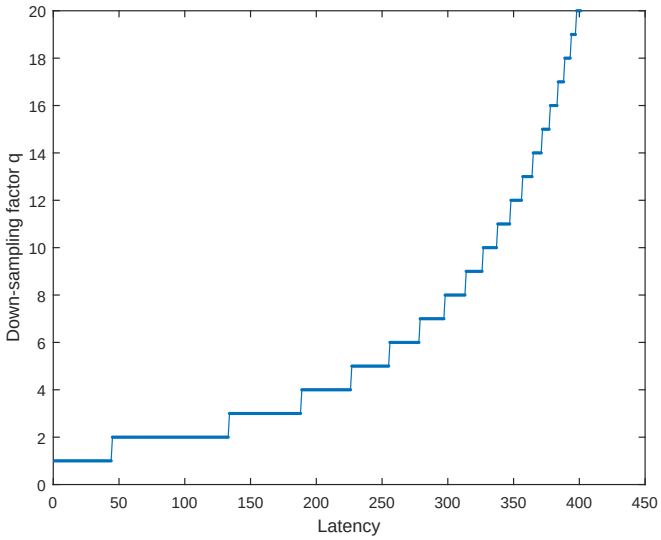

Figure 11: Down-sampling factor q as a function of the latency for the latency-dependent down-sampling.

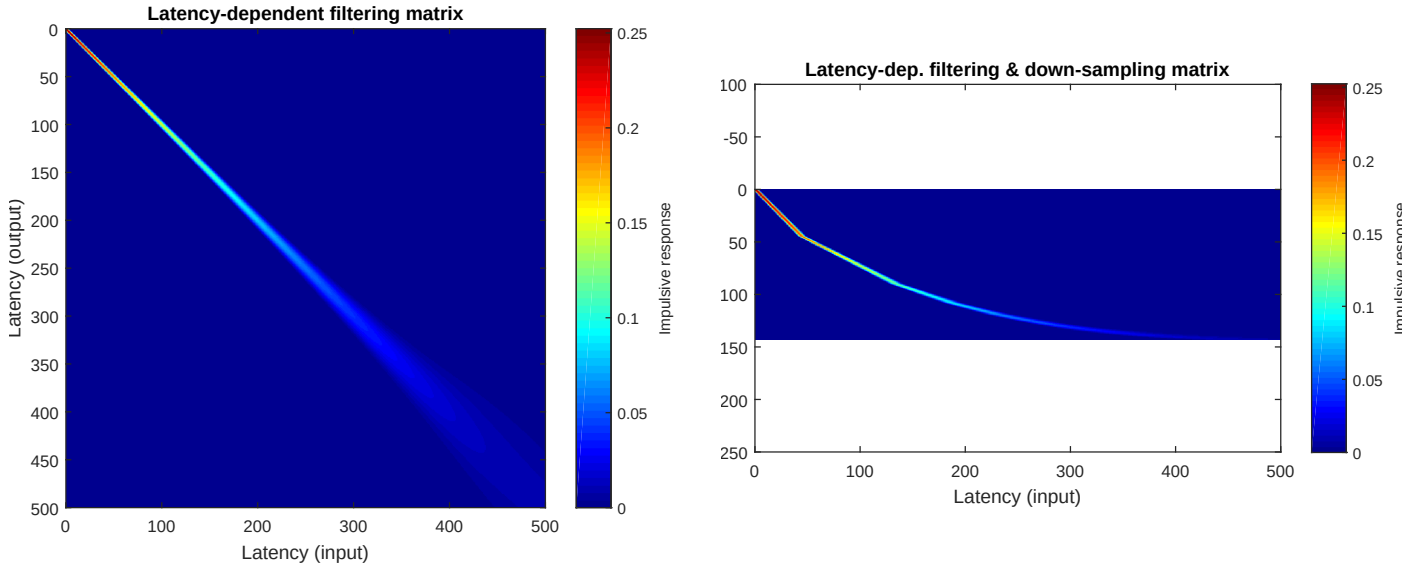

Figure 12: Matrix implementation of the latency-dependent low-pass filtering and down-sampling. Left panel: latency-dependent filtering matrix. Right panel: latency-dependent filtering and down-sampling matrix. Both the cut-off frequency and the sampling rate decreases as the latency increases.

cut-off frequency, with a minimum value of 1 sample). Figure 11 represents the local down-sampling factor q as a function of the latency, estimated with the previous criterion in order to prevent aliasing, according to the local cut-off frequency in figure 7. As can be observed, at small latency all the samples are preserved, and as the latency increases, the down-sampling factor q progressively increases (and therefore the sampling rate decreases).

Figure 12 shows the latency-dependent low-pass filtering matrix (left panel) and the latency-dependent low-pass filtering and down-sampling matrix (right panel). As can be observed, the down-sampling matrix is obtained by selecting specific rows of the initial matrix according to the latency-dependent down-sampling factor q in figure 11. Figure 13 shows a detail of the latency-dependent low-pass filtering and down-sampling matrix for small and large latency.

Finally, figure 14 compares the original signal, the latency-dependent low-pass filtered signal and the latency-dependent low-pass filtered and down-sampled signal. As can be observed, since the latency-dependent down-sampling has been performed in accordance with the sampling theorem, the down-sampled version preserves the information in the filtered signal (or equivalently, the filtered signal can be reconstructed from the down-sampled version).

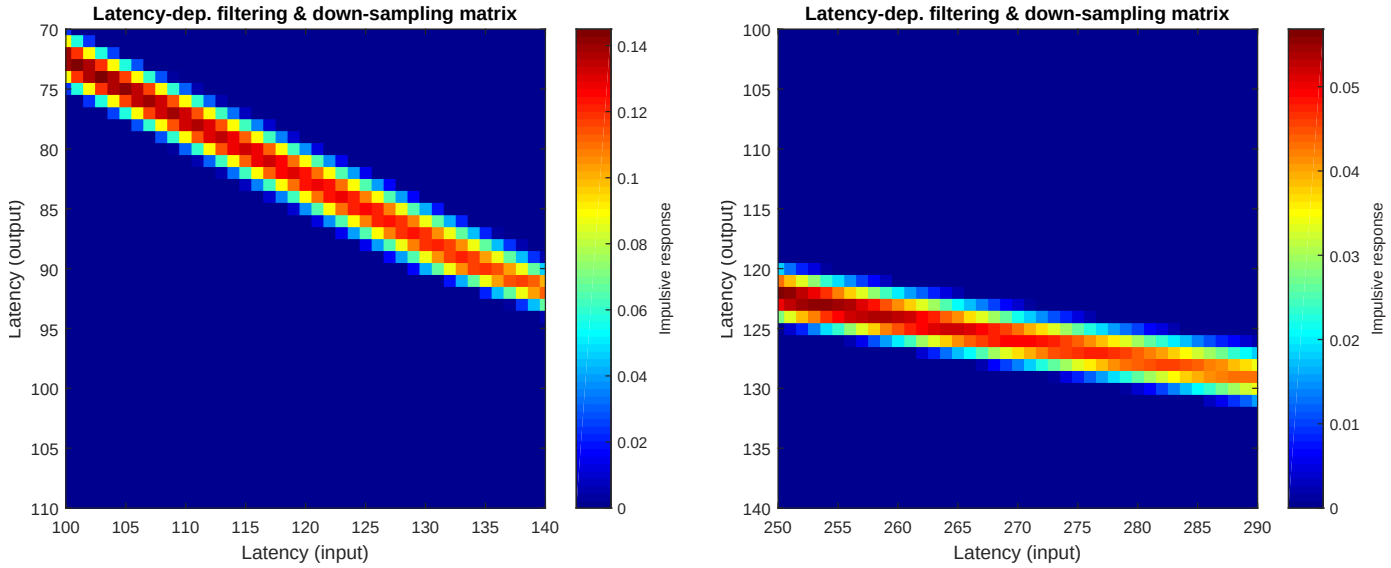

Figure 13: Detail of the latency-dependent low-pass filtering and down-sampling matrix for small and large latency.

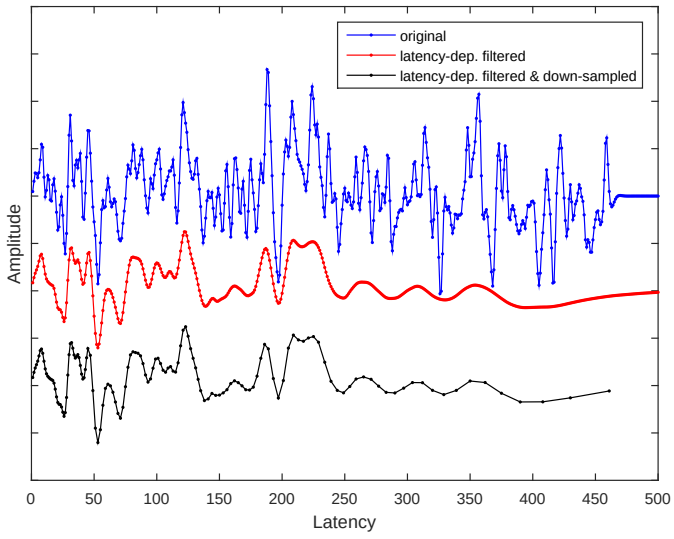

Figure 14: Comparison of the original signal, the latency-dependent low-pass filtered signal and the latency-dependent low-pass filtered and down-sampled signal.

3 Linear-logarithmic compression of the latency axis

This section describes the non-uniform compression of the latency axis providing a linear sampling (with a sampling rate similar to the original one) at small latency, and a logarithmic compression (with K_{dec} samples per decade) at large latency.

3.1 Uniform sampling at rate f_s

A uniform sampling with sampling rate f_s provides samples at uniform T_s intervals, being T_s the sampling period, with $T_s = 1/f_s$. The j^{th} sample is obtained at time t_j:

$$t_j = jT_s \tag{11}$$

From this equation, the sample index can be obtained as a function of the time:

$$j(t) = \frac{t}{T_s} \tag{12}$$

and the time instants corresponding to the samples are those providing an integer value of the index j in the previous equation.

3.2 Sampling in a logarithmically compressed latency axis

A non-uniform sampling of the latency axis is described as a non-linear relation between the latency t and the samples j. In particular, if the non-uniform sampling involves a logarithmic compression of the latency axis, the samples are described by the following equation:

$$j_r(t) = A \log_{10}\left(\frac{t}{T_s}\right) \tag{13}$$

The resolution of the logarithmically compressed sampling can be described in terms of the number of samples per decade. For a non-uniform sampling providing K_{dec} samples per decade, if a given latency t_0 corresponds to a sample $j_r(t_0)$ an increase in a factor 10 in the latency ($t_1 = 10 \cdot t_0$) should provide an increase of K_{dec} samples: $j_r(t_1) = j_r(t_0) + K_{dec}$. This definition of the resolution provides the equation of the logarithmically compressed sampling:

$$j_r(10 \cdot t_0) = j_r(t_0) + K_{dec} \tag{14}$$

$$j_r(10 \cdot t_0) = A \log_{10}\left(10\frac{t_0}{T_s}\right) = A \log_{10}\left(\frac{t_0}{T_s}\right) + A \log_{10}(10) = j_r(t_0) + A \tag{15}$$

and the identification of the last two equatoins provides the value of the constant A:

$$A = K_{dec} \tag{16}$$

and therefore, the relation between the latency and the sample index for a logarithmic compression of the latency axis for a resolution of K_{dec} samples per decade is given by the equation:

$$j_r(t) = K_{dec} \log_{10}\left(\frac{t}{T_s}\right) \tag{17}$$

Figure 15 represents the relation between the latency and the sample index for a logarithmic compression of the latency axis, with a resolution $K_{dec} = 10$ samples/decade. The solid line represents the function in equation (17), while the circles represent the latency values corresponding to each sample, i.e. those latency values providing an integer $j_r(t)$ in equation (17).

From this plot, it can easily be observed that the sampling period and the sampling rate are not constant. The sampling period increases (and the sampling rate decreases) as the latency increases. The local sampling frequency $f_s'(t)$ can be estimated as the partial $\partial j_r(t)/\partial(t)$ from equation (17):

$$f_s'(t) = \frac{K_{dec}}{\ln(10)}\frac{1}{t} \tag{18}$$

(where $\ln(\cdot)$ is natural logarithm) and the local sampling period as the inverse of the local sampling rate:

$$T_s'(t) = \frac{\ln(10)}{K_{dec}}t \tag{19}$$

As can be observed, the logarithmic compression of the latency axis requires a large number of samples at very small latency. The local sampling rate tends to infinite as the latency tends to zero, which makes no sense in practice, since the sampling rate after the compression should not be greater than the original sampling rate. This makes logarithmic compression unpractical at small latency.

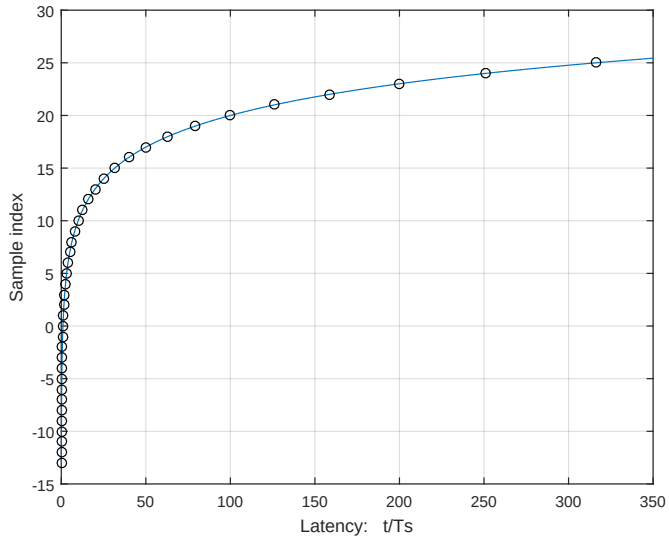

Figure 15: Non-uniform sampling with logarithmic compression.

3.3 Sampling with a linear-logarithmic compression

In order to avoid the singularity of the logarithmic compression at null latency, a linear-logarithmic compression can be proposed:

$$j_r(t) = A \log_{10}\left(B\frac{t}{T_s} + 1\right) \tag{20}$$

With this definition, at large latency a logarithmic compression is applied:

$$\lim_{t\to\infty} j_r(t) = A \log_{10}\left(B\frac{t}{T_s}\right) \tag{21}$$

and on the other hand, taking into account the following limit:

$$\lim_{x\to 0} \log_{10}(x+1) = \frac{x}{\ln(10)} \tag{22}$$

the relation between the latency and the sample index is linear at small latency:

$$\lim_{t\to 0} j_r(t) = \frac{AB}{\ln(10)}\frac{t}{T_s} \tag{23}$$

If a resolution of K_{dec} samples per decade is required at large latency, an increase in a factor 10 in the latency produces an increase of K_{dec} samples:

$$j_r(10\cdot t_0) \approx j_r(t_0) + K_{dec} \tag{24}$$

and expanding $j_r(10\cdot t_0)$ at large latency:

$$j_r(10\cdot t_0) \approx A \log_{10}\left(B\frac{10\cdot t_0}{T_s}\right) = A \log_{10}\left(B\frac{t_0}{T_s}\right) + A\log_{10}(10) = j_r(t_0) + A \tag{25}$$

and therefore, the identification of equations (24) and (25) provides the value of the constant A:

$$A = K_{dec} \tag{26}$$

On the other hand, at small latency, the local sampling rate should be equal to the original sampling rate:

$$\lim_{t\to 0} j_r(t) \approx \frac{t}{T_s} \tag{27}$$

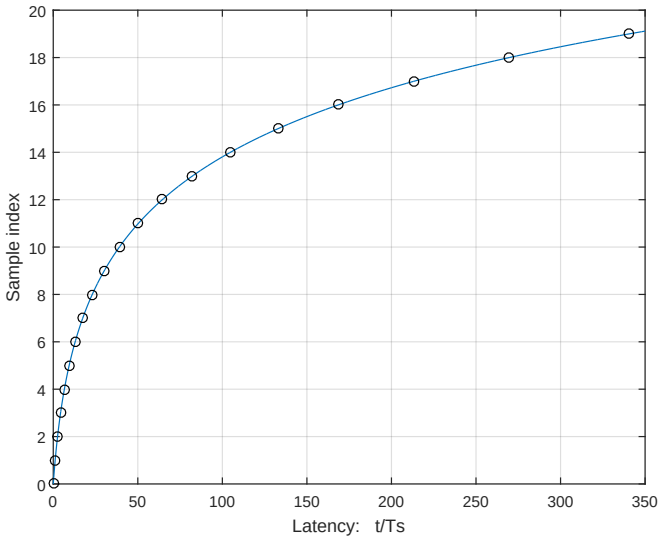

Figure 16: Non-uniform sampling with linear-logarithmic compression.

and taking into account the asymptotic behavior of $j_r(t)$ at low latency, we can write:

$$\lim_{t\to 0} j_r(t) \approx K_{dec} B \frac{t}{T_s} \frac{1}{\ln(10)} \tag{28}$$

The identification of the last two equations provides the value of the constant B:

$$\frac{K_{dec} B}{\ln(10)} = 1 \quad \Rightarrow \quad B = \frac{\ln(10)}{K_{dec}} \tag{29}$$

and therefore, the linear-logarithmic compression providing a resolution of K_{dec} samples per decade at large latency and a sampling period T_s at low latency is given by:

$$j_r(t) = K_{dec} \log_{10}\left(\frac{t}{T_s}\frac{\ln(10)}{K_{dec}} + 1\right) \tag{30}$$

Figure 16 represents the relation between the latency and the sample index for a linear-logarithmic compression of the latency axis with a resolution $K_{dec} = 10$ samples/decade. The solid line represent the function in equation (30), while the circles represent the latency values corresponding to each sample. A linear behavior is observed at small latency, with a slope 1 (corresponding to the original normalized sampling frequency), and the slope (corresponding to the local sampling frequency) decreases as the latency increases.

3.4 Local sampling rate and local sampling period

The local sampling rate is obtained as the partial $\partial j_r(t)/\partial t$ from equation (30):

$$f_s'(t) = \frac{j_r(t)}{\partial t} = \frac{K_{dec}}{\left(\frac{t}{T_s}\frac{\ln(10)}{K_{dec}} + 1\right)} \frac{1}{\ln(10)} \frac{1}{T_s} \frac{\ln(10)}{K_{dec}} \qquad f_s'(t) = \frac{1}{t\frac{\ln(10)}{K_{dec}} + T_s} \tag{31}$$

and the local sampling period is the inverse of the local sampling rate:

$$T_s'(t) = T_s + t\frac{\ln(10)}{K_{dec}} \tag{32}$$

Figure 17 shows the local sampling rate (left panel) and the local sampling period (right panel) as a function of the latency for a linear-logarithmic compression of the latency axis, with a resolution of 10 samples/decade. As can be observed, the local sampling rate is maximum at null latency, and is never greater than the original sampling rate. The local sampling rate decreases as the latency increases, and provides, at large latency, a resolution of K_{dec} samples per decade. Interestingly, at large latency, the local sampling rate and period depend on the latency and the resolution, but not on the original sampling rate.

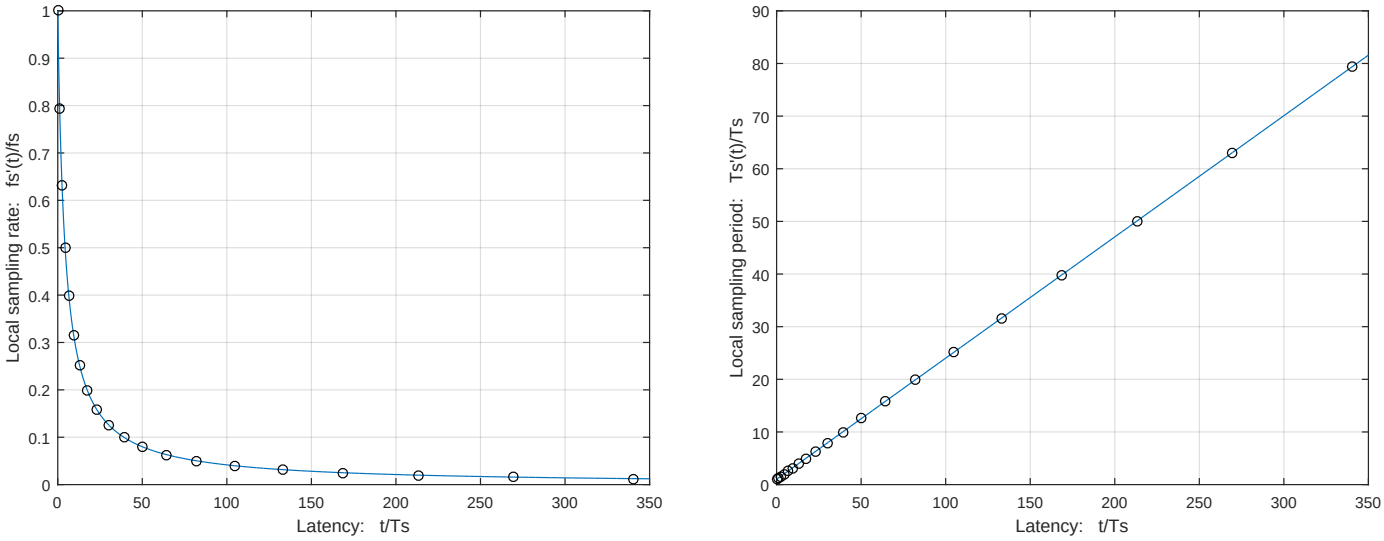

Figure 17: Local sampling rate (left panel) and local sampling period (right panel) for a linear-logarithmic compression with a resolution of 10 samples/decade.

4 Root-raised cosine filters

4.1 Raised cosine and root-raised cosine filters in digital communications

Root raised cosine filters (also known as square-root raised cosine filters) are filters providing a specific frequency response very useful in digital communications. When symbols are transmitted in a digital communication system, each symbol is represented as a signal in the transmitter, and the received signal is filtered with a matched filter in the receiver. The signals representing each symbol are generated by exciting the transmitter filter with impulses representing the symbols. The detection is performed in the receiver by filtering the received signal with the matched filter, sampling the output of the matched filter at the symbol rate, and comparing the sampled values with specific thresholds.

In order to avoid inter-symbol interference (ISI), the transmission channel (including the transmitter and receiver filters) must verify the Nyquist ISI criterion. Raised cosine (RC) filters are a family of filters verifying the Nyquist ISI criterion for the transmission channel. Since the transmission channel is separated into the transmitter and the receiver filters, the frequency response of the whole transmission channel is the product of the frequency responses of the transmitter and receiver filters. Since the receiver filter should match the transmitter filter, both frequency responses should be equal. For this reason, if a RC filter is proposed for the whole transmission channel, a root-raised cosine (RRC) filter (i.e. with a frequency response equal to the square root of that for the RC filter) is required for each, the transmitter filter and the receiver filter.

In this work we have selected the root raised cosine (RRC) filters for the latency-dependent filtering because of two reasons. On one hand, because the frequency response of the RRC filters is constant in a frequency range, and therefore these filters preserve without distortion the components within this frequency range. On the other hand because of the orthogonality of the RRC filters when they are delayed a integer number of symbols periods (which is a direct consequence of the Nyquist ISI criterion considered in their definition).

4.2 The Nyquist ISI criterion and the raised cosine filters

The Nyquist criterion for avoiding the inter-symbol interference in a digital communication (where symbols are transmitted at a symbol rate $1/T_0$) can be equivalently formulated either in the time domain or in the frequency domain. According to the time domain formulation, a filter with impulsive response $h(t)$ avoids ISI if the impulsive response is null at all the integer values of the symbol period except for $t = 0$, or equivalently if:

$$h(kT_0) = \begin{cases} 1 & \text{if} \quad k = 0 \\ 0 & \text{if} \quad k = \pm 1, \pm 2, \pm 3, \ldots \end{cases} \tag{33}$$

where T_0 is the symbol period. The Fourier transform of this criterion provides the formulation of Nyquist ISI criterion in the frequency domain, which requires that the frequency response convolved with the harmonics of the symbol rate (at k/T_0 with

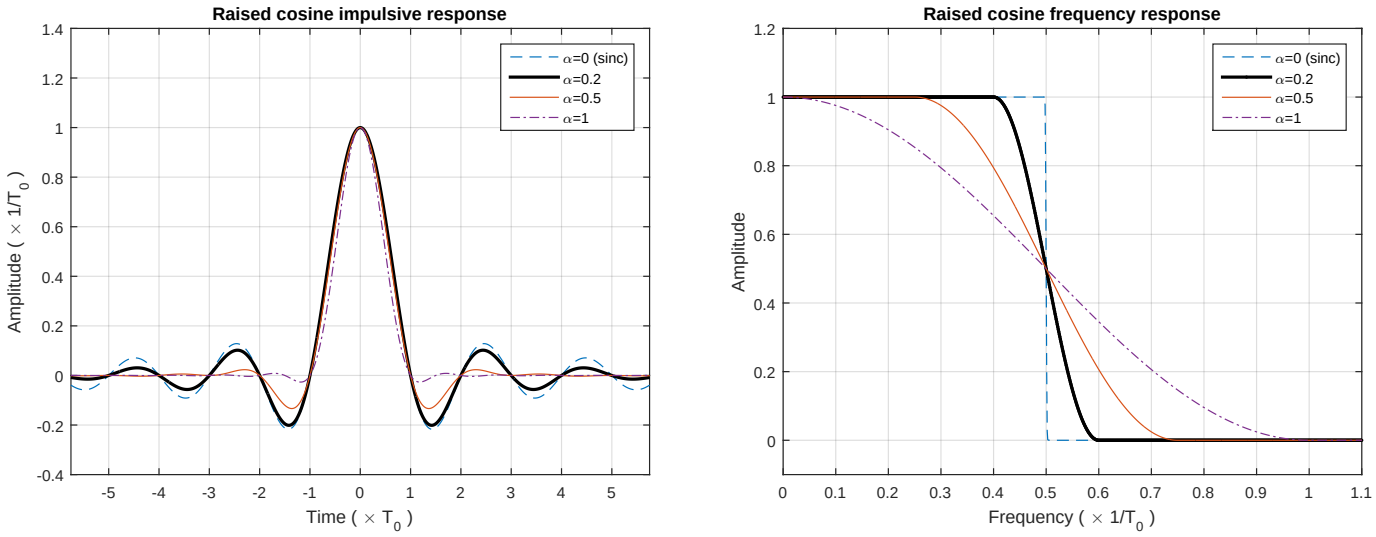

Figure 18: Left panel: impulsive response of raised cosine filters for symbol period T_0 (symbol rate $1/T_0$) and different values of the roll-off factor α (0, 0.2, 0.5, 1). Right panel: frequency response of the filters. Response normalized for unity gain at low frequency.

$k = 0, \pm 1, \pm 2, \ldots$) is constant:

$$\sum_{k=-\infty}^{+\infty} H\left(f - \frac{k}{T_0}\right) = \text{constant} \tag{34}$$

where $H(f)$ is the frequency response of the filter.

The raised cosine filters are a family of zero-phase filters verifying the Nyquist ISI criterion, specified by two parameters: the symbol period T_0 and the roll-off parameter α, and are defined from its frequency response as:

$$H_{RC}(f) = \begin{cases} 1 & \text{if} \quad |f| \leq \dfrac{1-\alpha}{2T_0} \\ \dfrac{1}{2}\left[1 + \cos\left(\dfrac{\pi T_0}{\alpha}\left(|f| - \dfrac{1-\alpha}{2T_0}\right)\right)\right] & \text{if} \quad \dfrac{1-\alpha}{2T_0} < |f| \leq \dfrac{1+\alpha}{2T_0} \\ 0 & \text{if} \quad |f| > \dfrac{1+\alpha}{2T_0} \end{cases} \tag{35}$$

As can be observed, their frequency response is constant between 0 and $(1-\alpha)/(2T_0)$, decreases between $(1-\alpha)/(2T_0)$ and $(1+\alpha)/(2T_0)$, and is null for frequency above $(1+\alpha)/(2T_0)$. It can also be observed that the frequency response verifies the Nyquist ISI criterion (in the frequency formulation). The impulsive response of the raised cosine filters is obtained as the inverse Fourier transform of their frequency response:

$$h_{RC}(t) = \begin{cases} \dfrac{1}{T_0} & \text{if} \quad t = 0 \\ \dfrac{\alpha}{2T_0}\sin\left(\dfrac{\pi}{2\alpha}\right) & \text{if} \quad t = \pm\dfrac{T_0}{2\alpha} \\ \dfrac{1}{\pi t}\sin\left(\dfrac{\pi t}{T_0}\right)\dfrac{\cos\left(\dfrac{\pi\alpha t}{T_0}\right)}{1-\left(\dfrac{2\alpha t}{T_0}\right)^2} & \text{otherwise} \end{cases} \tag{36}$$

The impulsive response and frequency response of the raised-cosine filters for different values of the roll-off factor α (0, 0.2, 0.5 and 1) are represented in figure 18.

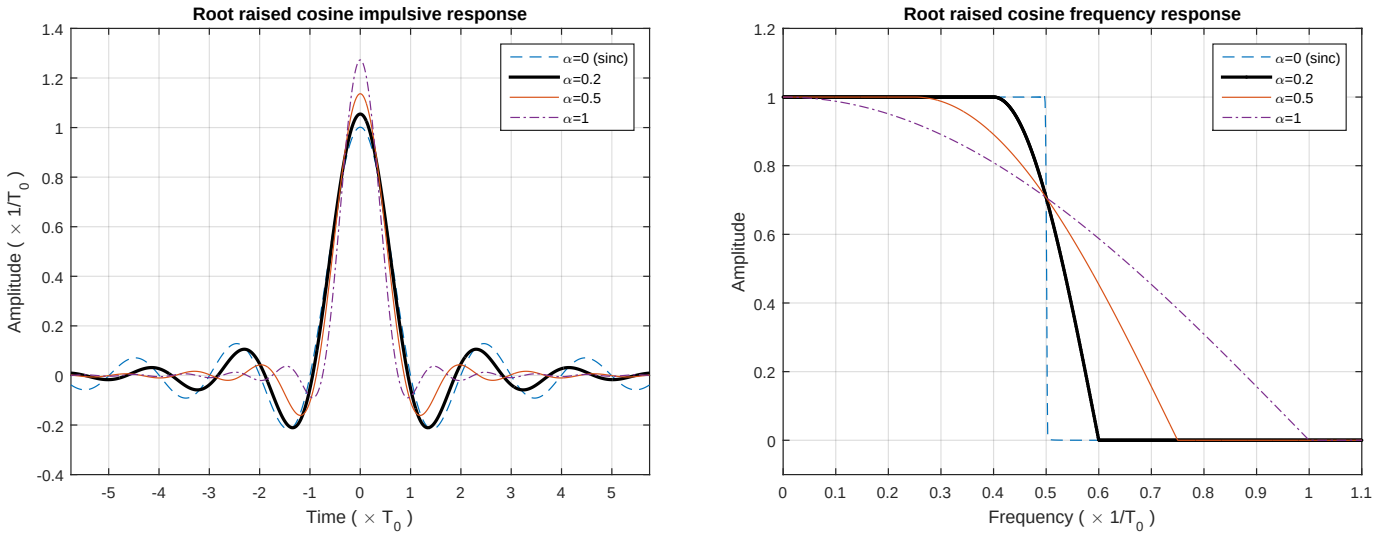

Figure 19: Left panel: impulsive response of root-raised cosine filters for symbol period T_0 (symbol rate $1/T_0$) and different values of the roll-off factor α (0, 0.2, 0.5, 1). Right panel: frequency response of the filters. Response normalized for unity gain at low frequency.

4.3 Impulsive response of the root-raised cosine filters

The root-raised cosine filters are derived from the frequency response of the raised cosine filters. Their impulsive response is defined as the inverse Fourier transform of the square root of the raised cosine frequency response. Therefore, the impulsive response of the RRC family of filters is obtained, for a roll-off factor α and a symbol period T_0, as:

$$h_{RRC}(t) = \begin{cases} \dfrac{1}{T_0} + \dfrac{\alpha}{T_0}\left(\dfrac{4}{\pi} - 1\right) & \text{if} \quad t = 0 \\ \dfrac{\alpha}{\sqrt{2}T_0}\left(\left(1 + \dfrac{2}{\pi}\right)\sin\left(\dfrac{\pi}{4\alpha}\right) + \left(1 - \dfrac{2}{\pi}\right)\cos\left(\dfrac{\pi}{4\alpha}\right)\right) & \text{if} \quad t = \pm\dfrac{T_0}{4\alpha} \\ \dfrac{\dfrac{1}{\pi t}\sin\left(\pi\dfrac{t}{T_0}(1-\alpha)\right) + \dfrac{4\alpha}{\pi T_0}\cos\left(\pi\dfrac{t}{T_0}(1+\alpha)\right)}{1 - \left(4\alpha\dfrac{t}{T_0}\right)^2} & \text{otherwise} \end{cases} \tag{37}$$

This definition can be found in texts related to digital communications, with different variants regarding the normalization (sometimes these equations are normalized with T_0 or with $\sqrt{T_0}$).

4.4 Properties of the root-raised cosine filters

The roll-off factor is a constant in the range [0-1] providing control of the bandwidth and the effective duration of the impulsive response. The minimum roll-off factor (null) provides a sinc response, with minimum bandwidth (equal to half of the symbol rate) and maximum effective duration of the impulsive response. The largest roll-off factor (1) provides an increase of the required bandwidth (up to the symbol rate) with minimum effective duration of the impulsive response. A roll-off factor α provides a frequency response constant in the interval $[0, (1-\alpha)]/(2T_0)$, decreasing in the interval $[(1-\alpha), (1+\alpha)]/(2T_0)$ and null for frequency above $(1+\alpha)/(2T_0)$.

Figure 19 represents the impulsive response and the frequency response of root-raised cosine filters designed with symbol period T_0 and several values of the roll-off factor (0, 0.2, 0.5 and 1). The amplitude attenuation in the impulsive response is faster as the roll-off factor increases. The frequency response shows an increasing bandwidth as the roll-off factor increases. Since the filters have been designed with symbol period T_0, the symbol rate is $1/T_0$. The filter for $\alpha = 0$ has a bandwidth equal to half of the symbol rate. All the filters present an attenuation of 3 dB at this frequency $(1/(2T_0))$. A constant response is observed in a portion of the

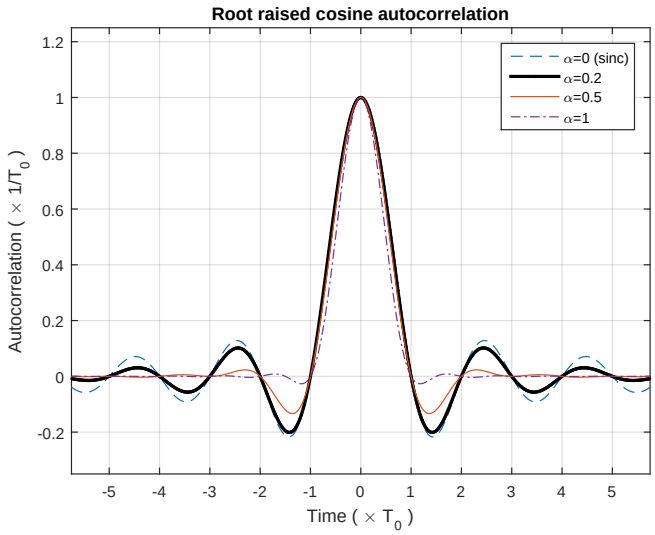

Figure 20: Autocorrelation function of the impulsive response of the root-raised cosine filters obtained with symbol period T_0 and different values of the roll-off factor α (0, 0.2, 0.5 and 1).

frequency response, then a decrease in the frequency response, and then a null response. In the case of $\alpha = 0.2$, the frequency response is constant between 0 and 40% of the symbol rate, decreasing between 40% and 60% of the symbol rate, and null for frequency above 60% of the symbol rate. The filters have been normalized in order to provide unity gain at low frequency.

Since the design of the RRC filters aims to verify the Nyquist ISI criterion for the cascade of two filters, the autocorrelation function of the impulsive response of the RRC filters is null for a delay of an integer number of symbol periods (except for null delay). This property provides the cancellation of the inter-symbol interference when RRC filters are used in digital communications. This property also provide orthogonality of the RRC functions delayed an integer number of symbol periods. Figure 20 shows the autocorrelation function of the impulsive response of the RRC filters with symbol period T_0 and roll-off factor 0, 0.2, 0.5 and 1. Figure 21 represents two RRC functions, obtained with $\alpha = 0.2$, delayed a symbol period T_0. The correlation of both functions (integral of the product of both) is null, showing both the orthogonality and the cancellation of the ISI interference.

The energy of the RRC functions in a given interval illustrates how fast is the attenuation and the potential inconvenience of a truncation. For $\alpha = 0$, the intervals $\pm 3T_0$ and $\pm 6T_0$ around the maximum provide 96.7714% and 98.4446% of the function energy, respectively. For $\alpha = 0.2$, these values increase up to 99.5740% and 99.9774% respectively, while for $\alpha = 0.5$, the energy for these intervals is 99.9632% and 99.9962%, respectively.

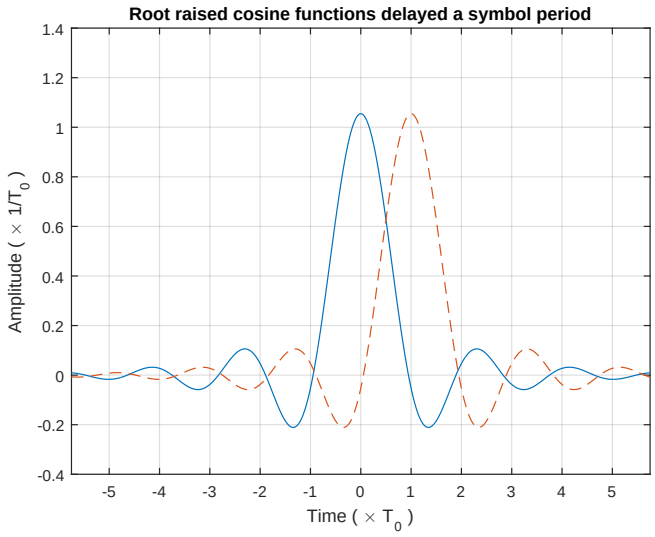

Figure 21: Two root-raised cosine functions (roll-off factor $\alpha = 0.2$) delayed a symbol period T_0.

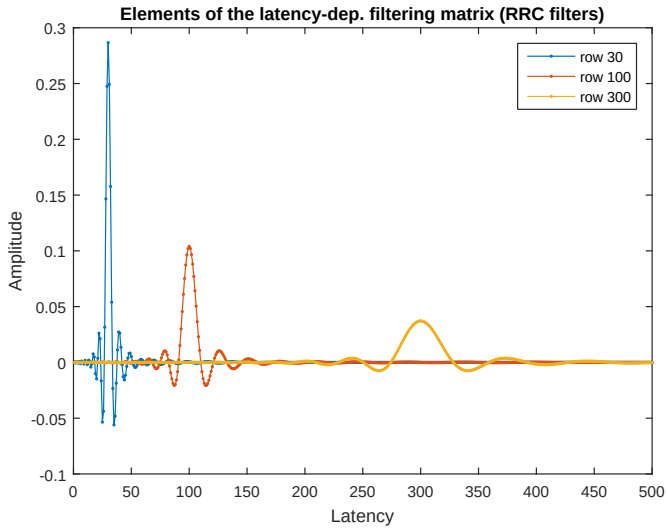

Figure 22: Elements of the latency-dependent filtering matrix implemented with root-raised cosine filters: rows 30, 100 and 300.

5 Latency-dependent low-pass filtering and down-sampling using RRC filters

5.1 Latency-dependent low-pass filtering matrix with RRC filters

In order to perform the latency-dependent low-pass filtering, a standard time-invariant filtering (with the root-raised cosine impulsive response) has been implemented in the compressed latency axis. The latency compression equation:

$$j_r(j) = K_{dec} \log_{10}\left(j\frac{\ln(10)}{K_{dec}} + 1\right) \tag{38}$$

(where $j = t/T_s$ is the sample index in the non-compressed latency axis) provides the sample index j_r in the compressed latency axis associated to each sample in the original latency axis. Since the latency-dependent filter is defined as a time-invariant filter in the compressed latency axis with the RRC impulsive response matching symbol period T_0 to the samplig period (i.e. T_0 equal to one sample in the compressed latency axis j_r), the impulsive response of the RRC filter can be expressed as a function of j_r as:

$$h(j_r) = \begin{cases} 1 + \alpha\left(\frac{4}{\pi} - 1\right) & \text{if} \quad j_r = 0 \\ \frac{\alpha}{\sqrt{2}}\left(\left(1 + \frac{2}{\pi}\right)\sin\left(\frac{\pi}{4\alpha}\right) + \left(1 - \frac{2}{\pi}\right)\cos\left(\frac{\pi}{4\alpha}\right)\right) & \text{if} \quad j_r = \pm\frac{1}{4\alpha} \\ \dfrac{\frac{1}{j_r\pi}\sin\left(\pi j_r(1-\alpha)\right) + \frac{4\alpha}{\pi}\cos\left(\pi j_r(1+\alpha)\right)}{1 - (4\alpha j_r)^2} & \text{otherwise} \end{cases} \tag{39}$$

Taking into account the latency compression equation and using the definition of $h(j_r)$, the elements of the matrix providing the latency-dependent low-pass filtering are obtained as:

$$H(j_1, j_2) = h(j_r(j_1) - j_r(j_2)) \tag{40}$$

with j_1 and j_2 in the range $(0, 1, \ldots, J-1)$.

Figure 22 represents 3 rows of the matrix implementing a latency-dependent low-pass filtering using root-raised cosine filters with $\alpha = 0.2$. The matrix have been prepared for $J = 500$ samples and the resolution was set to $K_{dec} = 25$ samples/decade. The impulsive responses have been normalized in order to provide gain unity at low frequency (the sum of the matrix coefficients in each row is 1 after normalization). The complete matrix is represented as an image in figure 23, and a detail of portions at early and late latency are represented in figure 24. In these figures, it can be observed how the impulsive response is wider as the latency increases (and the amplitude of the response decreases due to the normalization). Similarly, the bandwidth decreases with the latency.

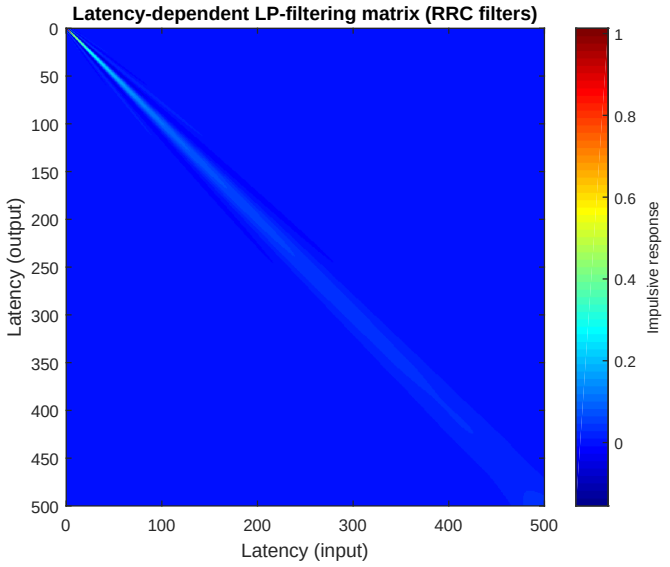

Figure 23: Matrix providing the latency-dependent low-pass filtering with RRC filters, α=0.2, J=500, K_{dec}=25 samples/decade.

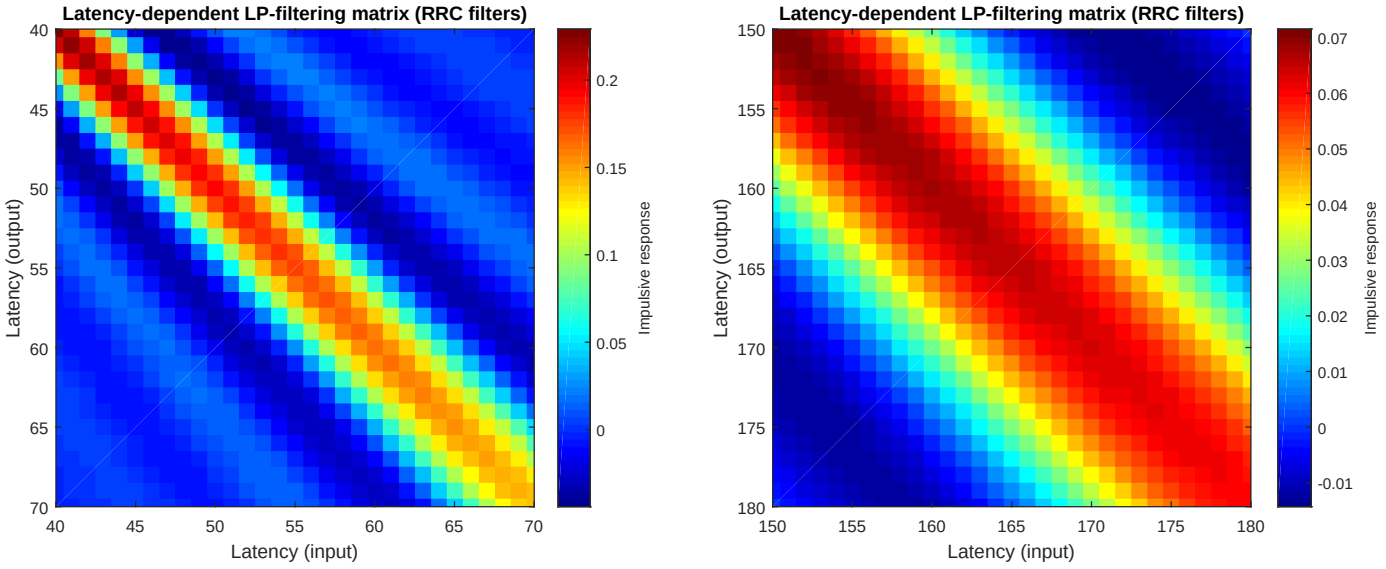

Figure 24: Portions of the RRC latency-dependent low-pass filtering matrix at small latency (left panel) and large latency (right panel).

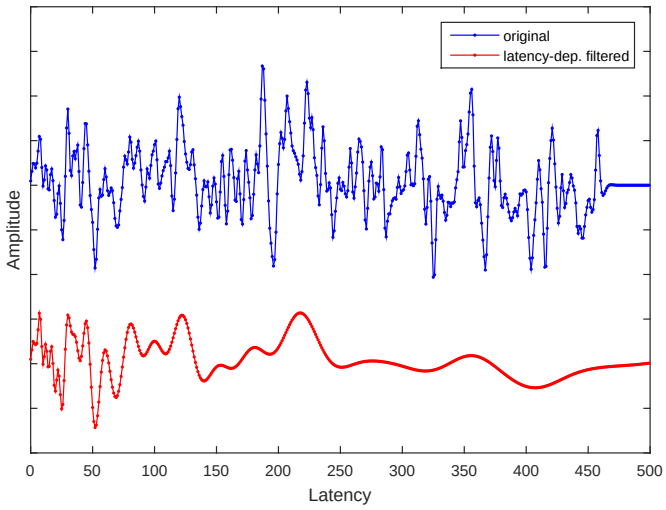

Figure 25: Comparison of the original and filtered signals. Signal filtered with the RRC latency-dependent low-pass filtering matrix with K_{dec}=25 samples/decade.

Figure 25 shows a signal (that one used in the previous examples) before and after the filtering, using the RRC latency-dependent low-pass filtering with K_{dec}=25 samples/decade. As can be observed, the frequency content of the filtered signal decreases with the latency.

In order to show the effect of the resolution parameter, a latency-dependent filtering matrix has been obtained with K_{dec}=60 samples/decade. Figure 26 compares three rows of the filtering matrices obtained with K_{dec}=25 and K_{dec}=60 samples/decade. As can be observed, the impulsive responses are narrower in the second case (and the amplitudes larger because of the normalization). Figure 27 shows the original signal and the signals filtered with both resolutions. In both cases the frequency content decreases as the latency increases, but when the resolution is set to 60 samples/decade, the bandwidth is greater than that for 25 samples/decade.

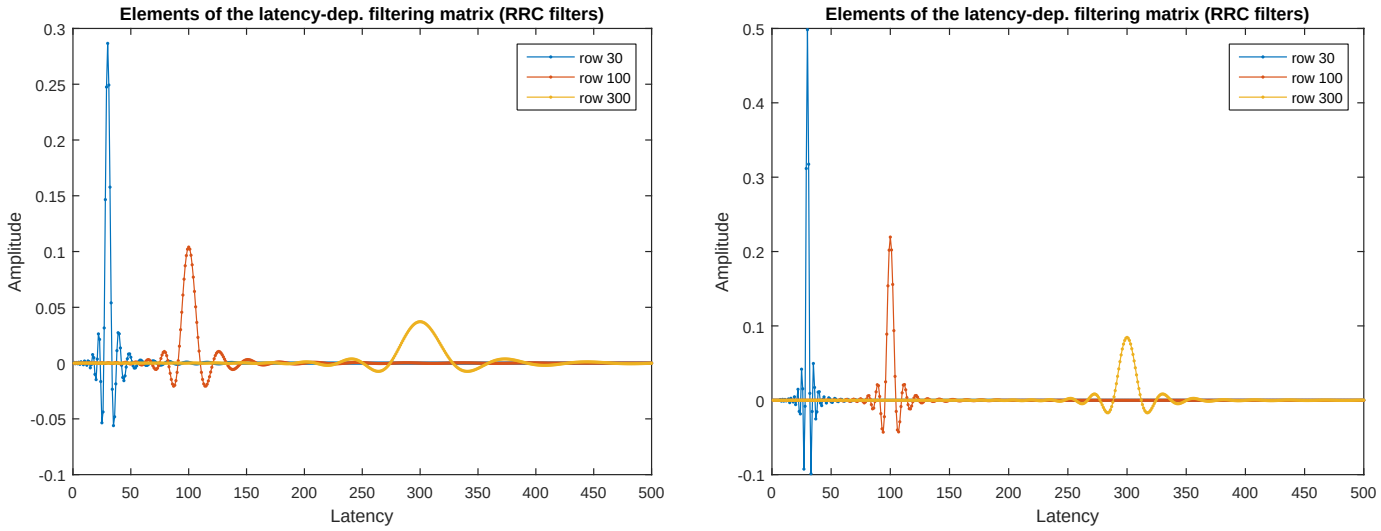

Figure 26: Three rows of the RRC latency-dependent low-pass filtering matrices with K_{dec}=25 (left panel) and K_{dec}=60 samples/decade (right panel).

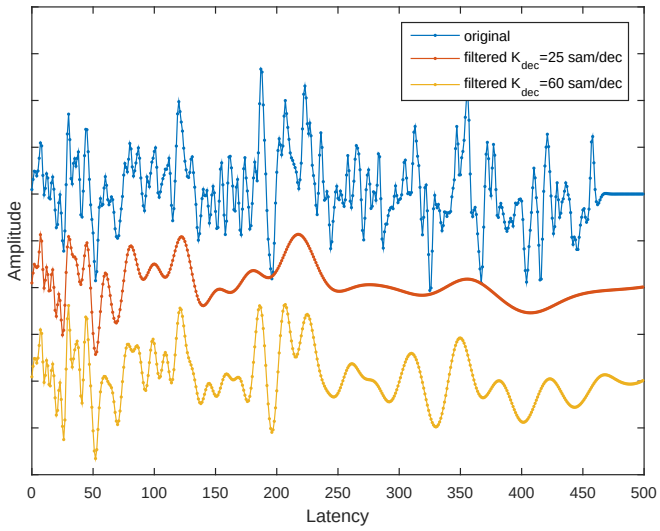

Figure 27: Comparison of the original and filtered signals. Signal filtered with the RRC latency-dependent low-pass filtering matrix with K_{dec}=25 and K_{dec}=60 samples/decade.

5.2 Latency-dependent low-pass filtering & down-sampling matrix with RRC filters

As previously described, the filtering and down-sampling matrix can be obtained by selecting specific rows in the filtering matrix. The rows to be selected are given by the integer values of the latency compression function $j_r(j)$. However, since the latency compression function does not provide integer values j_r for integer values of j in general, instead of selecting specific rows from the filtering matrix, specific rows at integer j_r values are estimated. The filtering and down-sampling matrix H_r is a $J_r \times J$ matrix, with elements $H_r(j_{r_1}, j_2)$ given by:

$$H_r(j_{r_1}, j_2) = h(j_{r_1} - j_r(j_2)) \tag{41}$$

where j_{r_1} and j_2 are in the ranges $(0, 1, \ldots, J_r - 1)$ and $(0, 1, \ldots, J - 1)$, respectively, and $J_r = j_r(J)$. According to the latency compression equation, the number of rows J_r (i.e. the final number of samples) depends on the initial number of samples J and the resolution K_{dec}.

Figure 28 represents the latency-dependent low-pass filtering and down-sampling matrices obtained with RRC filters for resolutions K_{dec}=25 and K_{dec}=60 samples/decade, in the left and right panels, respectively. The complete matrices are in the top panels, and portions at different latency ranges are shown at the middle and bottom panels. The number of rows in the matrices are J_r=41 and J_r=78 for K_{dec}=25 and K_{dec}=60, respectively. The elements in the rows are represented in the figure 29. As expected, in the case of 60 samples/decade (higher resolution) there is a higher density of functions (more density of samples after down-sampling) and the impulsive responses are narrower (wider bandwidth).

Finally, figure 30 compares the original signal, the filtered signal and the filtered and down-sampled signal for resolutions K_{dec}=25 and K_{dec}=60 samples/decade. The down-sampled signal contains 41 samples in the case of 25 samples/decade and 78 samples when K_{dec}=60 samples/decade.

5.3 Bandwidth preserved for each latency

As previously described, the latency-dependent filtering and down-sampling is performed using RRC filters with symbol period T_0 matching the latency-dependent sampling period $T'_s(t)$, and a roll-off α=0.2. The latency-dependent sampling period is given by the equation:

$$T'_s(t) = T_s + t\frac{\ln(10)}{K_{dec}} \tag{42}$$

where T_s is the original sampling period, t is the latency and K_{dec} the resolution. The RRC filter response is constant in the frequency range $[0, (1-\alpha)/(2T_0)]$, and therefore, for α=0.2, the latency-dependent filtering and down-sampling procedure preserves the frequency range $[0, 0.4/T'_s(t)] = [0, 0.4 \cdot f'_s(t)]$, where $f'_s(t) = 1/T'_s(t)$ is the latency-dependent sampling rate. The preserved bandwidth for $\alpha = 0.2$ is, therefore:

$$B(t) = 0.4\, f'_s(t) = 0.4\, \frac{f_s}{\frac{\ln(10)}{K_{dec}} f_s\, t + 1} \tag{43}$$

where f_s is the original sampling rate. As can be observed from this equation, the preserved bandwidth depends on the original sampling rate f_s, the latency t and the resolution K_{dec}. For a latency significantly greater than the original sampling period (i.e. for $t \gg T_s$), the unit in the denominator can be neglected, and the preserved bandwidth can be approximated as:

$$\lim_{t\to\infty} B(t) = 0.4\, \frac{K_{dec}}{\ln(10)\, t} \tag{44}$$

which does not depend on the original sampling frequency.

Figure 31 represents the bandwidth preserved by the latency-dependent filtering and down-sampling procedure using the proposed RRC filters, for different resolutions (K_{dec} between 10 and 200 samples/decade) and for original sampling rates f_s=14 700 Hz and f_s=25 000 Hz. At low latency, the preserved bandwidth tends to 0.4 f_s, while it does not depend on the original sampling rate at large latency.

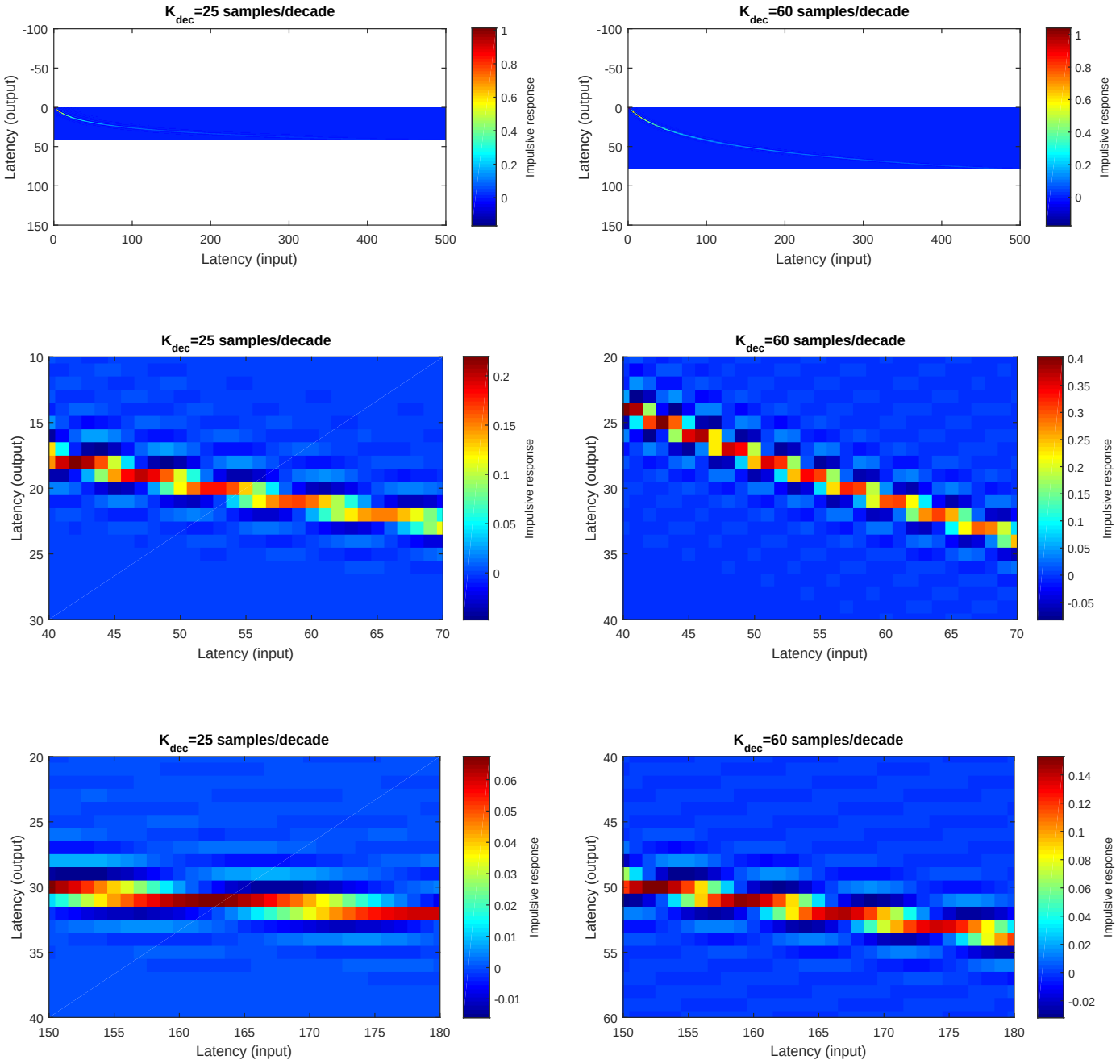

Figure 28: Matrices providing latency-dependent filtering and down-sampling for resolution K_{dec}=25 (left panels) and K_{dec}=60 samples/decade (right panels). From top to bottom, the complete matrix and portions corresponding to small and large latency.

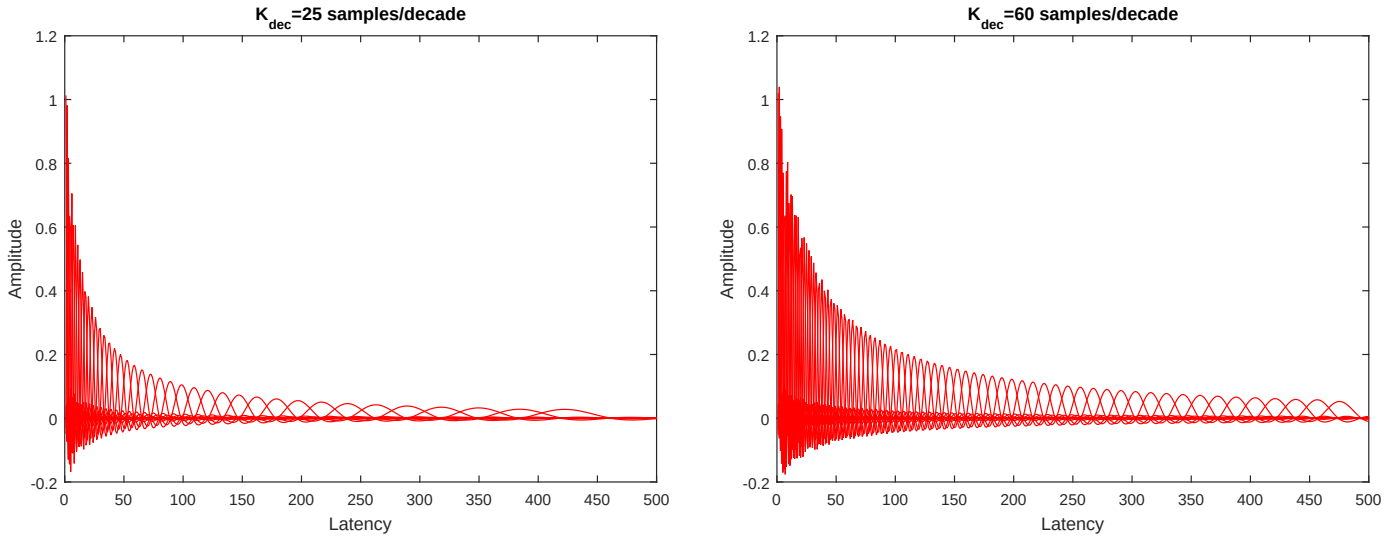

Figure 29: Elements in each row of the latency-dependent filtering and down-sampling matrices H_r, for K_{dec}=25 (left panel) and K_{dec}=60 samples/decade (right panel).

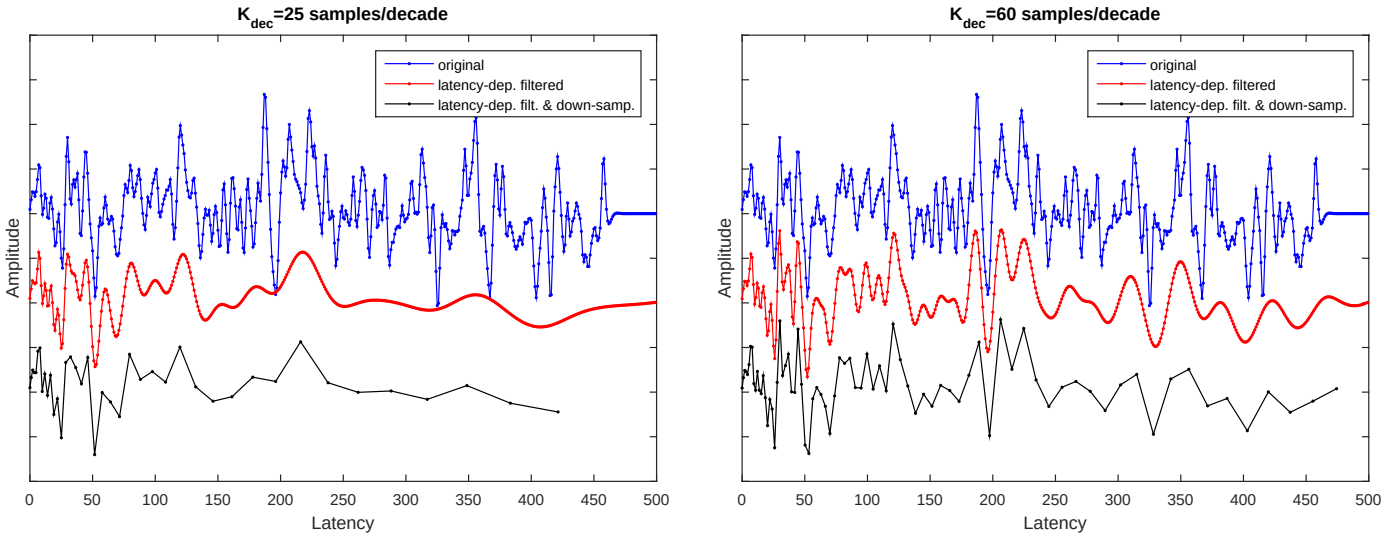

Figure 30: Comparison of the original signal, the filtered signal and the filtered and down-sampled signal for resolutions K_{dec}=25 (left panel) and K_{dec}=60 samples/decade (right panel).

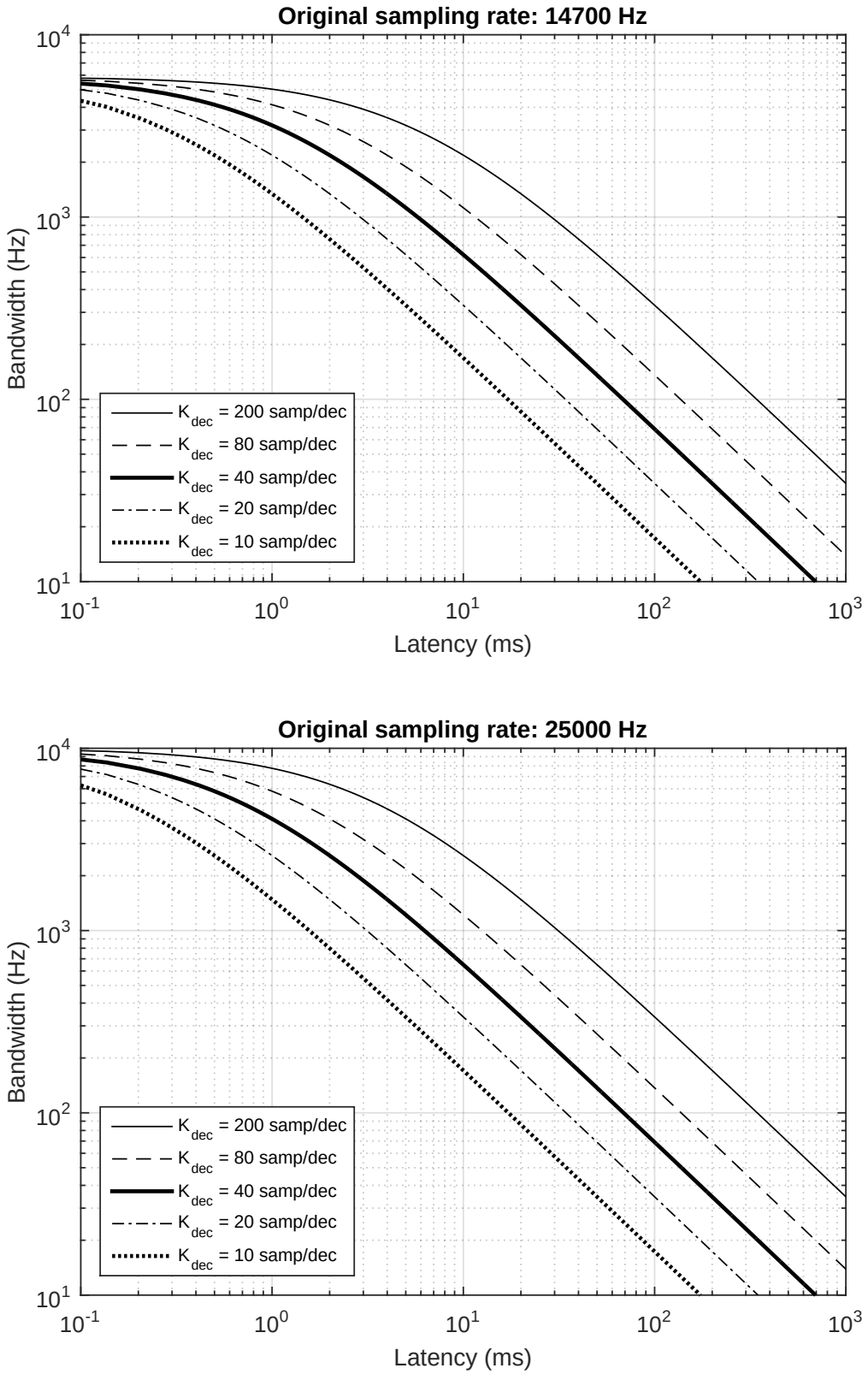

Figure 31: Bandwidth preserved by the latency dependent filtering as a function of the latency (using RRC filters with roll-off factor α=2). The bandwidth is estimated as $0.4 \cdot f_s'(t)$ for different resolutions between 10 and 200 samples/decade. Top and bottom panels: original sampling rate 14 000 Hz and 25 000 Hz, respectively.

6 Code to generate the matrix providing latency-dependent low-pass filtering and down-sampling

The following MatLab / Octave code generates a matrix providing latency-dependent low-pass filtering and down-sampling for resolution of K_{dec} samples/decade, to be applied to digital signals with J samples. The matrix is orthonormalized, so the rows of matrix V generated are an orthonormal basis of the subspace of dimension J_r expanded by the latency-dependent band-limited signals. The filters are generated from a root-raised cosine function with roll-off factor $\alpha = 0.2$. The latency-dependent bandwidth and down-sampling factor are defined by a lin-log compression of the latency axis.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% function [V] = Basis_LinLog_RRC(J,Kdec)     (revised Nov-2019)
%
% Basis_LinLog_RRC() constructs an orthonormal basis of functions (V),
% uniformly distributed in the lin-log-scaled time, with one function
% per sample at small latency and Kdec functions per decade at large
% latency.
% Root-raised-cosine (RRC) function is used for each element of the basis.
% This function is used in digital communications because it provides
% an appropriate limitation of the bandwidth with a relatively short
% duration in the impulsive response.
% The resulting functions are orthonormalized with Gram-Schmidt.
% The resulting basis is contained in a [J_red,J] matrix, with J_red<J.
% The matrix V contains J_red rows, each one with a vector of J components.
% The application of the basis V to a function x (V*x) provides a
% representation in a reduced representation space. The application of the
% transpose V' to the reduced representation (V'*(V*x)) provides a
% reconstruction from the reduced representation into the original time
% representation that is equivalent to a latency-dependent low-pass
% filtering of the response.
%
% INPUT:  [J]      Number of samples in the original representation
%         [Kdec]   Number of functions per decade
% OUTPUT: [V]      Orthonormal basis of functions provided as a matrix
%
% Example: V = Basis_LinLog_RRC(500,25);
%
% Angel de la Torre, Jose Carlos Segura, Joaquin Valderrama (2019)
%     University of Granada (Spain)
%     National Acoustic Laboratories, Macquarie University (Australia)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [V] = Basis_LinLog_RRC(J,Kdec)
%%%% Check for correct input of data
if(J<40||J<Kdec||Kdec<5||Kdec>500)
    error('Error: V=Basis_LinLog_RRC(J,Kdec); (J>=40,J>=Kdec,5<=Kdec<=500)')
end
%%%% Initialization of variables
j=0:(J-1);           % Time-axis, linear scale (in samples)
jr_samp=Kdec*log10(j*log(10)/Kdec+1);  % Time-axis, compressed scale
jr=0:(jr_samp(end)-1); % Samples in compressed scale (functions in basis)
K=length(jr);        % K: Number of functions of the base
%%%% Function template: raised cosine function
Npt_sym=40;           % number of points per symbol period in raised-cosine
N_per=14.6;           % number of periods to each side of the raised-cosine
alpha=0.20;           % roll-off factor (low-pass filtering effect)
[h,tau]=sr_rcos(Npt_sym,N_per,alpha);
%%%% Set of functions of the base before amplitude normalization (V0)
V0 = zeros(length(jr),length(j)); % V0: K functions with J*5 samples
for k=1:K
    % Functions are placed at jr(k) latency (linearly distr in compr. scale)
    V0(k,:)=interp1(tau,h,jr(k)-jr_samp,'linear',0);
end
%%%% Gram-Schmidt orthonormalization
V=OrthoNorm_Gauss_fast(V0,K);
return;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% function [h,tau] = sr_rcos(Npt_sym,n_per,alpha)
%
```

```
% This function prepares the samples of a square root raised cosine filter
%  See Proakis, J. (1995). Digital Communications. McGraw-Hill Inc. or
%  https://en.wikipedia.org/wiki/Root-raised-cosine_filter)
% Proakis and wikipedia formulas are identical except for a factor Ts
% In this implementation Ts=1;
% Input parameters:
%   Npt_sym samples per symbol (or symbol period expressed in samples)
%   n_per   number of periods to the left and to the right of maximum
%   alpha   roll off factor (between 0 and 1)
% output:
%   h       impulsive response
%   tau     normalized time, t/Ts
% example:
%   [h,tau]=sr_rcos(20,6,0.35)
function [h,tau] = sr_rcos(Npt_sym,N_per,alpha)
if alpha == 0, alpha = realmin; end
tau = (-N_per:1/Npt_sym:N_per); h=zeros(size(tau));
% (A) response for t=0:
cond1=tau==0;
h(cond1)=1+alpha*(4/pi-1);
% (B) response for denominator=0:
denom0=1-(4.*alpha.*tau).^2;
cond2=abs(denom0) < sqrt(eps);
phi=pi/(4*alpha);
h(cond2)=alpha/(sqrt(2))*( (1+2/pi)*sin(phi) + (1-2/pi)*cos(phi) );
% (C) response for all the other samples:
cond3=~(cond2|cond1);
t1=tau(cond3);
phi1=pi*t1*(1-alpha); phi2=pi*t1*(1+alpha);
denom=pi*t1.*denom0(cond3);
h(cond3)=(sin(phi1) + 4*alpha*t1.*cos(phi2))./(denom);
return;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% V=OrthoNorm_Gauss_fast(V0,K)
% Orthonormalization, Gauss Method (fast implementation)
% Input parameters:
%    V0  matrix with vectors to be orthonormalized
%    K   number of vectors
% Output paramter:
%    V   matrix with orthonormalized vectors
function V=OrthoNorm_Gauss_fast(V0,K)
% prenormalization
for k=1:K,  v=V0(k,:); V0(k,:)=v/sqrt(dot(v,v)); end;
P=V0*V0';       % Gaussian elimination of [V0*V0' | V] = [P | V0]
M_all=eye(K);   % Matrix operations to be applied to P for Gaussian elimination
for j1=1:K-1
    M_tmp=eye(K);
    for j2=(j1+1):K
        M_tmp(j2,j1)=-P(j2,j1); M_tmp(j2,j2)=P(j1,j1);
        P(j2,:)=P(j2,:)*P(j1,j1)-P(j1,:)*P(j2,j1);
    end
    M_all=M_tmp*M_all;
    M_tmp=eye(K);
    M_tmp(j1+1,j1+1)=1/P(j1+1,j1+1);
    M_all=M_tmp*M_all;
    P(j1+1,:)=P(j1+1,:)/P(j1+1,j1+1);
end
V=M_all*V0;    % Matrix operations applied to V0
% Final normalization
for k=1:K, v=V(k,:);  v=v/sqrt(dot(v,v));  V(k,:)=v;  end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

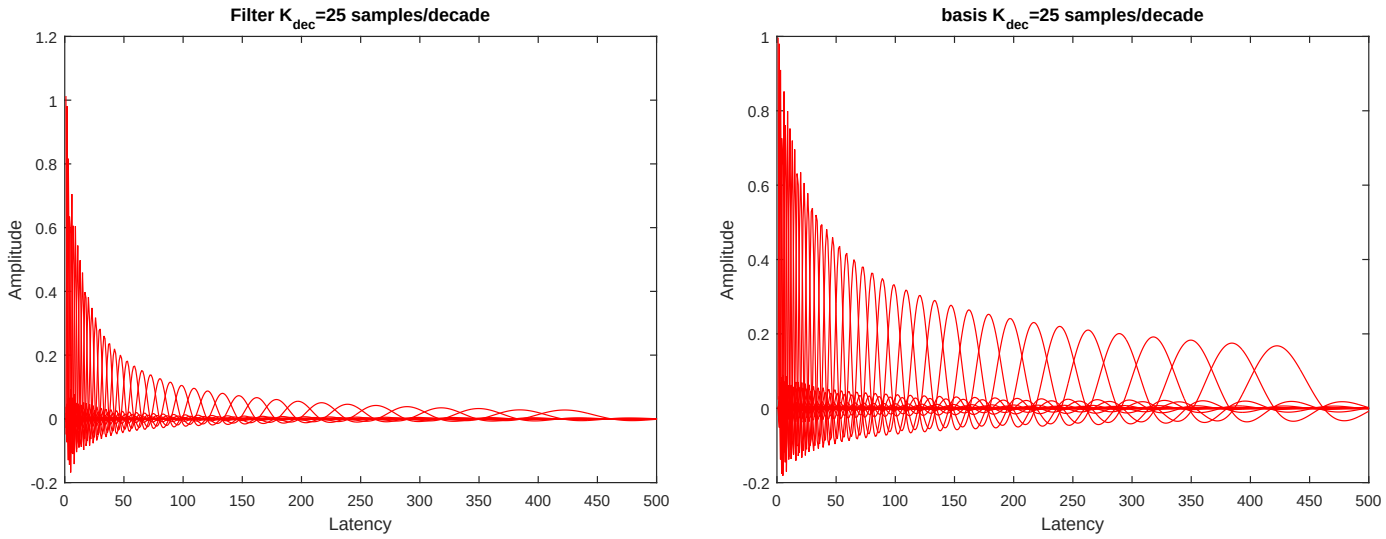

Figure 32: Left panel: rows of the matrix providing latency-dependent filtering and down-sampling with K_{dec}=25 samples/decade. Right panel: rows of the matrix providing the corresponding orthonormal basis.

7 The basis of functions for the reduced representation space

After the orthonormalization of the filtering and down-sampling matrix H_r, the resulting orthonormalized matrix V_r constitutes a basis of functions describing the subspace of dimension J_r of the latency-dependent band limited signals. The matrix product $\mathbf{x}_r = V_r \cdot \mathbf{x}$ provides the representation of the projection of the signal $\mathbf{x}$ over the subspace of functions expanded by the functions of the basis V_r, and therefore the J_r components of the vector $\mathbf{x}_r$ constitute a compact representation of the signal $\mathbf{x}$ after the latency dependent low-pass filtering.

Figure 32 shows the rows of the matrix implementing the latency-dependent filtering and down-sampling before (left) and after the orthonormalization (right). Since the rows of the matrix are quasi-orthogonal (they are orthogonal in the compressed latency axis but the non-linear compression produces a slight deviation from orthogonality) the functions before and after the orthonormalization are almost identical except for the amplitude. In the filter, normalization is applied to provide gain unity at low frequency:

$$\sum_{j=0}^{J-1} H_r(j_r, j) = 1 \quad \text{for all} \quad j_r \in \{0, 1, \ldots, J_r - 1\} \tag{45}$$

while in the basis, normalization is applied to provide an orthonormal basis:

$$\sum_{j=0}^{J-1} V_r(j_{r_1}, j) V_r(j_{r_2}, j) = \begin{cases} 1 & \text{if} \quad j_{r_1} = j_{r_2} \\ 0 & \text{if} \quad j_{r_1} \neq j_{r_2} \end{cases} \tag{46}$$

(or equivalently, $V_r \cdot V_r^T = I_{J_r}$, where I_{J_r} is the $J_r \times J_r$ identity matrix) and for this reason, the amplitude reduction with latency is faster in the filter than in the basis.

8 Reconstruction of the signal in the original latency axis

8.1 Matrix product $V_r \cdot \mathbf{x}$ as a decomposition; reconstruction of the signal from the decomposition

The J_r components of the vector $\mathbf{x}_r$ provide the decomposition of the signal $\mathbf{x}$ in the components defined by the basis. Each component $x_r(j_r)$ is the coefficient to be multiplied with the corresponding vector in the basis $H(j_r, j)$ to recover the filtered signal in the original representation. Therefore, in order to recover the signal in the original representation, the contribution from the different vectors in the basis should be added:

$$x_{lp}(j) = \sum_{j_r=0}^{J_r-1} x_r(j_r) V_r(j_r, j) \tag{47}$$

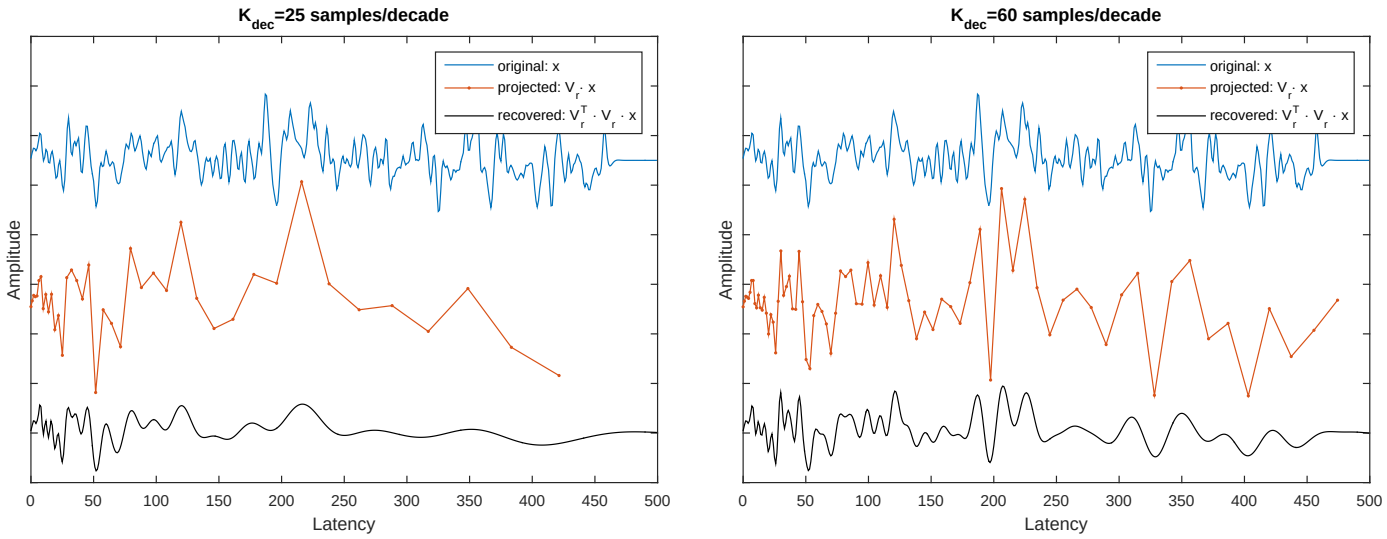

Figure 33: Recovered signals for 25 (left panel) and 60 samples/decade (right panel). The plots are the original signal (top) the projected signal $\mathbf{x}_r = V_r \cdot x$ (center) and the recovered signal $\mathbf{x}_{lp} = V_r^T \cdot V_r \cdot \mathbf{x} = V_r^T \cdot \mathbf{x}_r$ (bottom).

This equation can be rewritten in matrix form:

$$\mathbf{x}_{lp} = V_r^T \cdot \mathbf{x}_r \tag{48}$$

where V_r^T is the transposed of V_r. Since V_r is an incomplete basis (the number of vectors in V_r is $J_r < J$) the product $\mathbf{x}_r = V_r \cdot \mathbf{x}$ is a projection to a subspace, and the product $V_r^T \cdot \mathbf{x}_r = V_r^T \cdot V_r \cdot \mathbf{x}$ does not recover the original signal, but the projected signal transformed from the compact representation in the subspace to the original representation. In fact, since the matrix V_r implements a latency-dependent filtering, the recovered signal $V_r^T \cdot \mathbf{x_r}$ is a latency-dependent low-pass filtered version of the original signal, transformed to the original representation.

Figure 33 represents the original signal $\mathbf{x}$, the projected signal $\mathbf{x}_r = V_r \cdot \mathbf{x}$ and the recovered signal $\mathbf{x}_{lp} = V_r^T \cdot \mathbf{x}_r$, for resolutions K_{dec}=25 and K_{dec}=60 samples/decade. The components of the vector $\mathbf{x}_r$ have been represented at the latency corresponding to the not compressed representation, and the amplitudes increase with respect to the amplitude of the signal as the latency increases due to the orthonormalization of the elements in V_r. As can be observed, the recovered signal $\mathbf{x}_{lp}$ is a latency-dependent low-pass filtered version of the original signal.

8.2 Equivalence of recovering with $V_r^T \cdot V_r$ and latency-dependent low-pass filtering

Since the matrix V_r provides latency-dependent low-pass filtering and down-sampling and the matrix V_r^T provides recovering of the down-sampled signal to the original representation (i.e. with the non-compressed latency axis), the composition of both matrices provide a simple latency-dependent low-pass filtering, and therefore, the application of the matrix H_{lp} (providing latency-dependent filtering) and $V_r^T \cdot V_r$ should be equivalent.

In order to verify the equivalence of both, we have compared the latency-dependent low-pass filtered signals $\mathbf{x}_{lp1} = H_{lp} \cdot \mathbf{x}$ and $\mathbf{x}_{lp2} = V_r^T \cdot V_r \cdot \mathbf{x}$. Figure 34 represents the original signals, both filtered signals and the difference between them. This comparison has been performed using the original signal in the previous example with J=500 samples (left panel), and for J=5000 samples (right panel). As can be observed in this figure, the filtered signals are similar, but they are not identical, and the difference is small but significant.

In order to evaluate the similarity of both filtered signals, some spectrograms have been presented in figure 35. These spectrograms have been estimated with a logarithmic compression of the latency axis by resampling the signal with 200 samples/decade (after appropriate low-pass filtering). The spectrograms show the power spectral density (in dB) with a colormap, as a function of the latency (horizontal axis, log-scaled) and the frequency (in oscillations/decade, in the vertical axis). The spectrograms are represented for the original signal, the signal filtered with H_{lp}, the signal recovered with $V_r^T \cdot V_r$ and the difference between both filtered signals.

Regarding the original signal, since it is white noise band-limited to $0.2 f_s$ being f_s the sampling rate, at low latency the resolution of 200 oscillations/decade implies an oversampling, and there is no power for the number of oscillations/decade corresponding to a frequency greater than $0.2 f_s$. At large latency, (greater than 200 samples) the resolution of 200 samples/decade implies a down-sampling and the power spectral density covers all the frequency range.

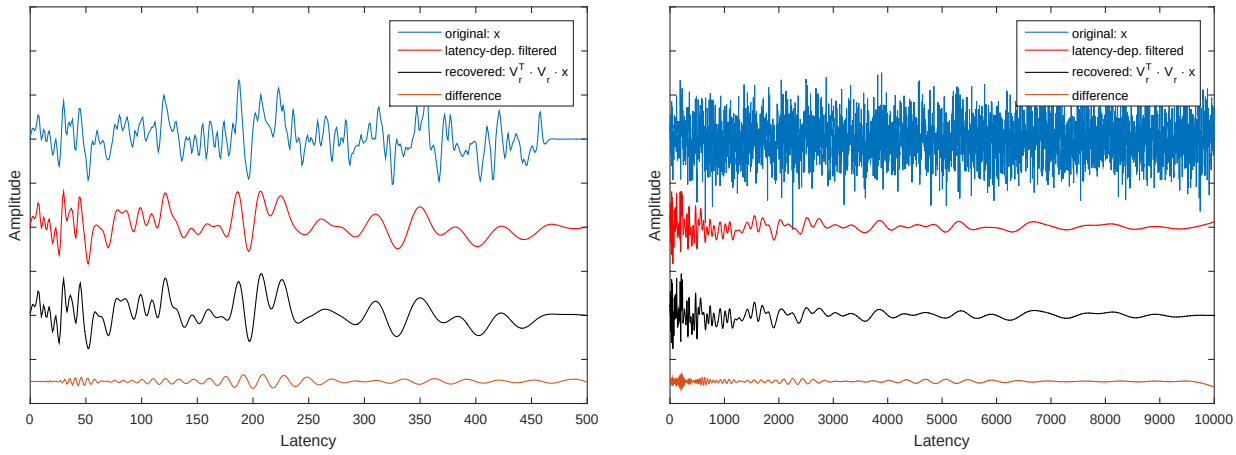

Figure 34: Comparison of the latency-dependent low-pass filtered signals obtained with the filtering matrix H_{lp} and with the projection and recovering procedure using the basis $V_r^T \cdot V_r$, for K_{dec}=60 samples/decade. Signals from top to bottom: original; filtered ($H_{lp} \cdot \mathbf{x}$); recovered ($V_r^T \cdot V_r \cdot \mathbf{x}$); and difference between the last two signals. Two signal lengths are considered: J=500 (left) and J=5000 samples (right).

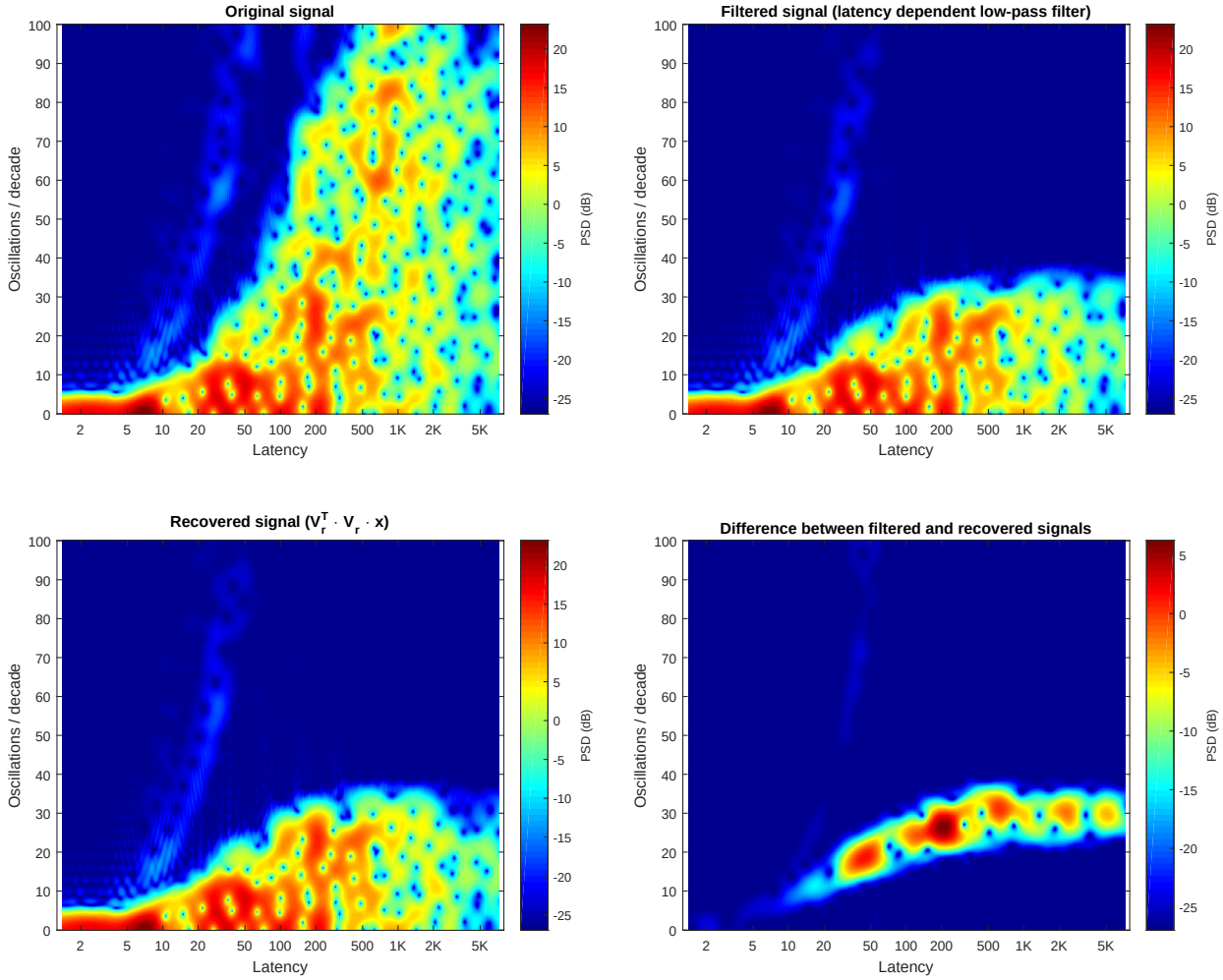

Figure 35: Spectrograms over a log-scaled latency axis. The spectrograms show the power spectral density (in dB) with the colormap, as a function of the latency (horizontal axis, log-scaled) and the frequency (in oscillations/decade, in the vertical axis). Spectrograms shown for the original, filtered and recovered signals, and the difference between the filtered and recovered signals.

The spectrograms of both filtered signals are identical at low frequency, and they are also identical to the low frequency portion of the original signal, as expected, since both latency-dependent low-pass filtering procedures are expected to be equivalent and provide a low-pass filtered version of the original signal. The differences are clear in the last spectrogram (the spectrogram of the difference). At large latency (between 500 and 5000 samples), the spectrograms are identical below 24 oscillations/decade and are different between 24 and 36 oscillations/decade. These differences are consequence of the aliasing of the root-raised cosine filters with α=0.2, sampled at the symbol period and using K_{dec}=60 samples/decade. The RRC filter with α=0.2 preserved unaltered the frequency range between 0 and 0.4/T_0 (being T_0 the inverse of the symbol period, and in this case equal to the sampling rate, i.e. 60 samples/decade), or in other words, provides gain unity in the range from 0 to 24 samples/decade. The RRC filter also removes the components beyond 36 samples/decade. However, those components between 24 and 36 samples/decade are attenuated (but nor removed) and since half of the sampling rate is 30 samples/decade, aliasing problems are expected in the band between 24 and 36 samples/decade. Therefore, the aliasing is the origin of the difference between the filtering process and the recovering process (including filtering, down-sampling and restoration).

The aliasing should be taken into account when a signal is latency-dependent low-pass filtered and down-sampled. If the signal contains no components beyond 40% of the sampling rate and the power spectral density of the noise is not very important in the range 40%-60% of the sampling rate, aliasing is not a problem because it only affects the noise components. Otherwise, an additional latency-dependent low-pass filtering should be applied.

9 Code for signal reconstruction from the compact representation; examples

9.1 Code generating the reconstruction matrix

The following MatLab / Octave code generates a matrix V_{rec} providing reconstruction of the signal, from the reduced representation $\mathbf{x}_r$, to the original representation, evaluated at a specific set of latency values $\{t_{rec}\}$. The vector $\mathbf{x}_{rec}$ resulting from the product $V_{rec} \cdot \mathbf{x}_r$ is the reconstructed signal $x_{lp}(t)$ evaluated at the specified latency values $\{t_{rec}\}$.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% function Vr = Basis_reconstruction(V,fs,t_rec)      (revised Nov-2019)
% This function provides a transformation Vr providing the reconstruction
% of a function x(t) evaluated at times t_rec after a projection with a
% transformation providing a representation in the reduced space V:
%  * Vr*(V*x) provides V'*(V*x) evaluated at times given by t_rec
%  * It is equivalent to:   interp1(t,V'*(V*x),t_rec) (but more efficient)
% Example
%    x          original function at t=(0:(J-1))/fs
%    V = Basis_LogScale(J,K_dec);   provides the transformation V
%    V*x;       representation at reduced space
%    V'*(V*x);  projection into original representation space
%    Vr*(V*x);  reconstruction of projection for time instants t_rec
%    semilogx(t,x,t,V'*(V*x),t_rec,Vr*(V*x),t_rec,interp1(t,V'*(V*x),t_rec))
% Syntax:
%    Vr = Basis_reconstruction(V,fs,t_rec)
% Input parameters:
%    V: basis of functions providing representation in reduced space
%    fs: sampling frequency of original representation
%    t_rec: time values in which the function is being reconstructed
% Output parameters:
%    Vr: transformation providing reconstruction at t_rec from reduced
%    representation
% Angel de la Torre, Jose Carlos Segura, Joaquin Valderrama (2019)
%     University of Granada (Spain)
%     National Acoustic Laboratories, Macquarie University (Australia)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function Vr = Basis_reconstruction(V,fs,t_rec)
[K,J]=size(V);
t=(0:(J-1))/fs;
Vr=zeros(length(t_rec),K);
for k=1:K
    Vr(:,k)=interp1(t,V(k,:),t_rec,'linear',0)';
end
return
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

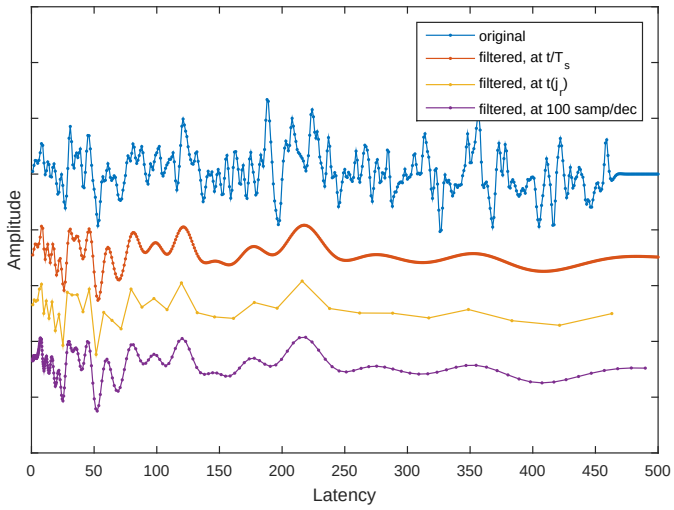

Figure 36: Reconstruction of the filtered signal from the compact representation. From top to bottom, (a) the original signal $x(j)$, (b) the filtered signal $x_{lp}(j)$ estimated as $\mathbf{x}_{lp} = V_r^T \cdot \mathbf{x}_r$, and the filtered signal reconstructed (c) at $t(j_r)$ and (d) with a resolution of 100 samples/decade, respectively, using $x_{rec}(t_{rec}) = \mathbf{x}_{rec} = V_{rec} \cdot \mathbf{x}_r$. Latency dependent filtering implemented with K_{dec}=25 samples/decade.

9.2 Examples of signals reconstructed at specific values of latency

Figure 36 shows the latency-dependent filtered signals obtained with the reconstruction matrix at different sets of latency values $\{t_{rec}\}$. The signals correspond to the example with 500 samples of filtered noise, for which a latency-dependent filtering and down-sampling has been applied with 25 samples/decade. In order to recover the filtered signal at the original representation (at the samples $j = t/T_s$), the transposed of the matrix V_r has been applied to the reduced representation: $\mathbf{x}_{lp} = V_r^T \cdot \mathbf{x}_r$. In order to reconstruct the filtered signal at specific latency values $\{t_{rec}\}$, the corresponding reconstruction matrix V_{rec} has been applied to the reduced representation: $\mathbf{x}_{rec} = V_{rec} \cdot \mathbf{x}_r$. The reconstruction has been obtained for those latency values corresponding to the compressed samples ($\{t_{rec}\} = t(j_r)$), and also for latency values providing a resolution of 100 samples/decade. Figure 36 includes (a) the original signal; (b) the filtered signal (reconstructed at the original samples); (c) the filtered signal reconstructed at the samples in the compressed latency axis; and (d) the filtered signal reconstructed with a resolution of 100 samples/decade.

In this example, the representation of the latency-dependent filtered signal requires 500 samples in the original representation (J=500), 41 samples in the compact representation ($J_r = 41$), and 270 samples for a compact representation with 100 samples/decade (because covering the latency range of the example with 100 samples/decade requires 270 samples).

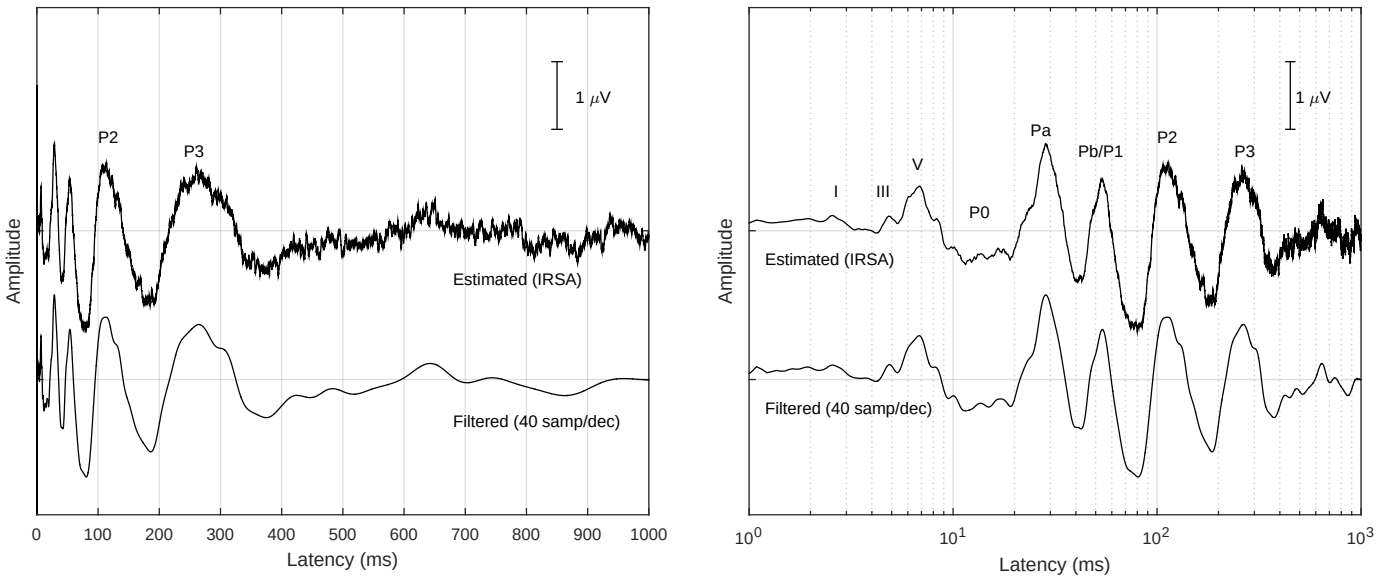

Figure 37: Response used for the simulations (grand average AEP response to 74 dB clicks presented at ISI 480-960 ms, estimated with IRSA (de la Torre et al., J. Acoust. Soc. Am. 146 (6), Dec. 2019, 4545-4556), before filtering (top) and filtered using a latency-dependent low-pass filter with 40 samples/decade (bottom). Left panel: response as a function of the linearly-scaled latency; right panel: response as a function of the logarithmically scaled latency. The filtered response has been used as reference for the simulations.

10 AEP Response used as reference in the simulations

This section describes the AEP response used as reference in the simulations. The selected response corresponds to the grand average of the estimated AEP response to clicks presented with an inter-stimulus interval (ISI) randomized with uniform distribution between 480 ms and 960 ms, with a stimulation level of 74 dB (Hearing Level). The grand average was estimated from 4 subjects by applying the IRSA algorithm to the recorded EEGs. This response was published in (de la Torre et al., J. Acoust. Soc. Am. 146 (6), Dec. 2019, 4545-4556). Detailed description about the experimental procedure for estimating this response can be found in this reference.

Figure 37 shows the AEP response, as a function of the linearly scaled latency (left panel) and the logarithmically scaled latency (right panel). The main waves of ABR, MLR and CAEP responses can be identified and are marked in the plot of logarithmically scaled latency. Even though the grand average response is less affected by noise than the individual responses, some noise is observed in the response (due to the noise affecting the EEG acquisition), particularly evident for the late portion in the plot with logarithmically scaled latency.

In order to obtain a clean response for the simulations, and taking into account the frequency content expected in the AEP responses at the different latency values, a latency-dependent low-pass filter has been applied with a resolution K_{dec}=40 samples/decade. The upper plots in the figure correspond to the original grand average response $\mathbf{x}_0$ (as provided by the IRSA algorithm), while the plots in the bottom correspond to the latency-dependent filtered response $\mathbf{x}_{ref}$ used as reference in the simulations. The orthonormalized filtering matrix V_r has been prepared for J=14700 (response length 1 s, sampling rate f_s=14.7 kHz) and K_{dec}=40 samples/decade. From the original response $\mathbf{x}_0$, the filtered response used as reference has been estimated as $\mathbf{x}_{ref} = V_r^T \cdot \mathbf{x}_r = V_r^T \cdot (V_r \cdot \mathbf{x}_0)$.

11 Detailed experimental results with simulations

A simulated EEG has been synthesized with a random ISI in the range 480-960 ms, using the reference response $\mathbf{x}_{ref}$. Noise has been added to the synthetic EEG, with a reasonable noise level (providing a SNR around 10 dB in the estimated response). Different types of noise have been considered in the simulations: pink noise (power spectral density decreasing with frequency with a slope of -3dB/octave), white noise (flat power spectral density) and real EEG noise (extracted from an EEG recording session).

The contaminated EEG has been processed with the IRSA algorithm in order to estimate the AEP response $\mathbf{x}$. The estimated AEP response has been latency-dependent filtered using an orthonormal matrix: $\mathbf{x}_{lp} = V_r^T \cdot V_r \cdot \mathbf{x}$. Different resolutions K_{dec} between 10

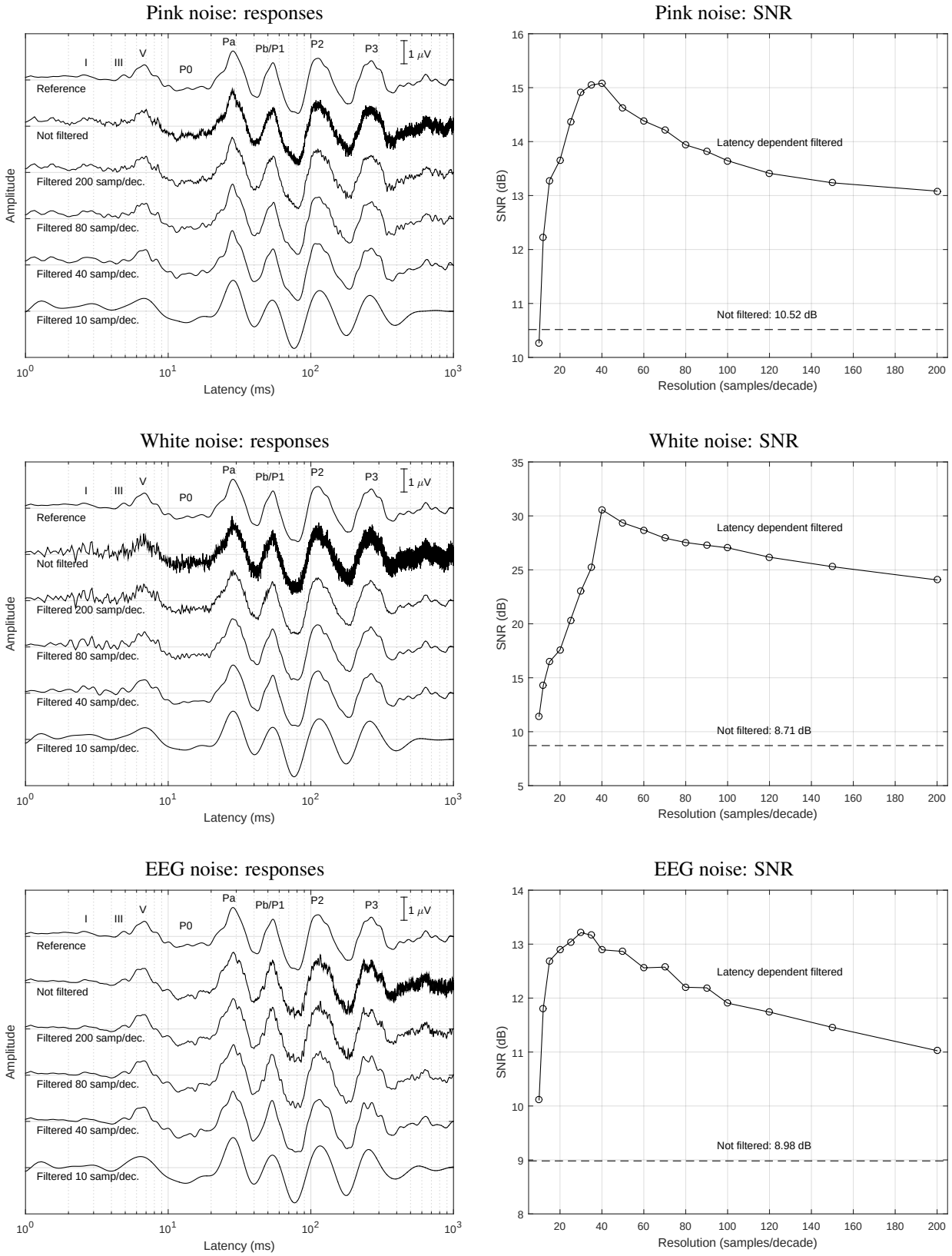

Figure 38: Effect of the latency-dependent low-pass filtering on the response. Simulation using a synthetic EEG (generated with the reference AEP response) contaminated with noise. Plots in the left side: responses (reference $\mathbf{x}_{ref}$, not filtered $\mathbf{x}$ and filtered with different resolutions $\mathbf{x}_{lp} = V_r^T \cdot V_r \cdot \mathbf{x}$). Plots in the right side: SNR with respect to the reference response as a function of the resolution in the latency-dependent filtering. From top to bottom: responses estimated from a synthetic EEG contaminated with pink noise, white noise and real EEG noise.

NOT FILTERED								
Resolution	Bandwidth	Number	SNR (dB)			SNR improvement (dB)		
(samp./second)	(Hz)	of samples	pink	white	EEG	pink	white	EEG
14700	0-7350	14700	10.52	8.71	8.98	0	0	0
LATENCY-DEPENDENT LOW-PASS FILTERED								
Resolution	Bandwidth	Number	SNR (dB)			SNR improvement (dB)		
(samp./decade)	(oscilations/dec.)	of samples	pink	white	EEG	pink	white	EEG
200	0-80	446	13.08	24.07	11.03	2.57	15.35	2.04
150	0-60	353	13.24	25.29	11.46	2.72	16.58	2.47
120	0-48	294	13.41	26.17	11.74	2.90	17.46	2.76
100	0-40	253	13.65	27.04	11.91	3.13	18.33	2.92
90	0-36	231	13.82	27.28	12.19	3.30	18.56	3.21
80	0-32	210	13.94	27.53	12.20	3.42	18.82	3.22
70	0-28	187	14.22	27.94	12.57	3.70	19.23	3.59
60	0-24	165	14.38	28.69	12.56	3.87	19.98	3.58
50	0-20	141	14.63	29.36	12.86	4.11	20.64	3.88
40	0-16	117	15.08	30.58	12.90	4.57	21.86	3.91
35	0-14	104	15.05	25.20	13.17	4.53	16.49	4.18
30	0-12	91	14.91	23.04	13.21	4.40	14.33	4.23
25	0-10	78	14.37	20.34	13.04	3.85	11.63	4.06
20	0-8	64	13.66	17.58	12.90	3.14	8.86	3.92
15	0-6	50	13.28	16.48	12.69	2.76	7.77	3.71
12	0-4.8	41	12.22	14.27	11.81	1.70	5.56	2.83
10	0-4	35	10.27	11.41	10.11	-0.24	2.70	1.13

Table 1: Comparison of the different representations of the AEP responses (not filtered and latency-dependent filtered with different resolutions). For each resolution, the corresponding representations can be compared in terms of the bandwidth, the required number of samples and the SNR when the EEG is contaminated in the simulations with pink, white or real EEG noise. The SNR is estimated with respect to the AEP response used as reference in the simulations. The SNR improvement is referred to the not filtered response.

and 200 samples/decade have been considered. Since the reference response $\mathbf{x}_{ref}$ used in the simulations is available, the quality of the estimated responses (before and after the latency-dependent filtering) can be evaluated in terms of the SNR.

Figure 38 shows the effect of the latency-dependent low-pass filtering in the experiments involving simulations. The plots in the left side represent the AEP responses (reference $\mathbf{x}_{ref}$, not filtered $\mathbf{x}$, and latency-dependent filtered $\mathbf{x}_{lp}$ with resolutions of 200, 80, 40 and 10 samples/decade) when the EEG is contaminated with pink, white and EEG noise (plots in the top, center and bottom, respectively). The plots in the right side represent the SNR (with respect to the AEP response $\mathbf{x}_{ref}$ used as reference). The solid line with circles corresponds to the latency-dependent filtered response, while the dashed line is the SNR of the not filtered response. As can be observed, the latency-dependent filtering improves the quality of the responses by appropriately removing the high frequency noise. As expected, the resolution providing optimal results is around 40 samples/decade (which is the resolution used for preparing the reference response). In the case of the pink and EEG noise models, the latency-dependent filtering provides a SNR improvement close to 4 dB in a wide range of resolutions (25-60 samples/decade for the pink noise; 20-50 samples/decade for the EEG noise). The improvement is more important (with a narrower range around 40 samples/decade) in the case of the white noise (however, the white noise seems to be an inappropriate model of noise for AEP recordings).

In addition to the SNR improvement, the latency-dependent filtering provides a compact representation, requiring a small number of samples for the AEP responses. Table 1 compares the representation of the not filtered response and the latency-dependent filtered responses, in terms of the preserved bandwidth, the required number of samples and the SNR. While the conventional representation of the AEP response requires 14700 samples, the compact representation with 40 samples/decade provided by the corresponding orthonormal matrix requires 117 samples, which implies less than 1% of the storage requirements. As illustrated in figure 39, the responses in the reduced representation space $\mathbf{x}_r = V_r \cdot \mathbf{x}$ can be converted to the standard representation by the matrix product $\mathbf{x}_{lp} = V_r^T \cdot \mathbf{x}_r$. This figure compares the latency-dependent filtered responses estimated with different resolutions for the synthetic EEG contaminated with pink noise.

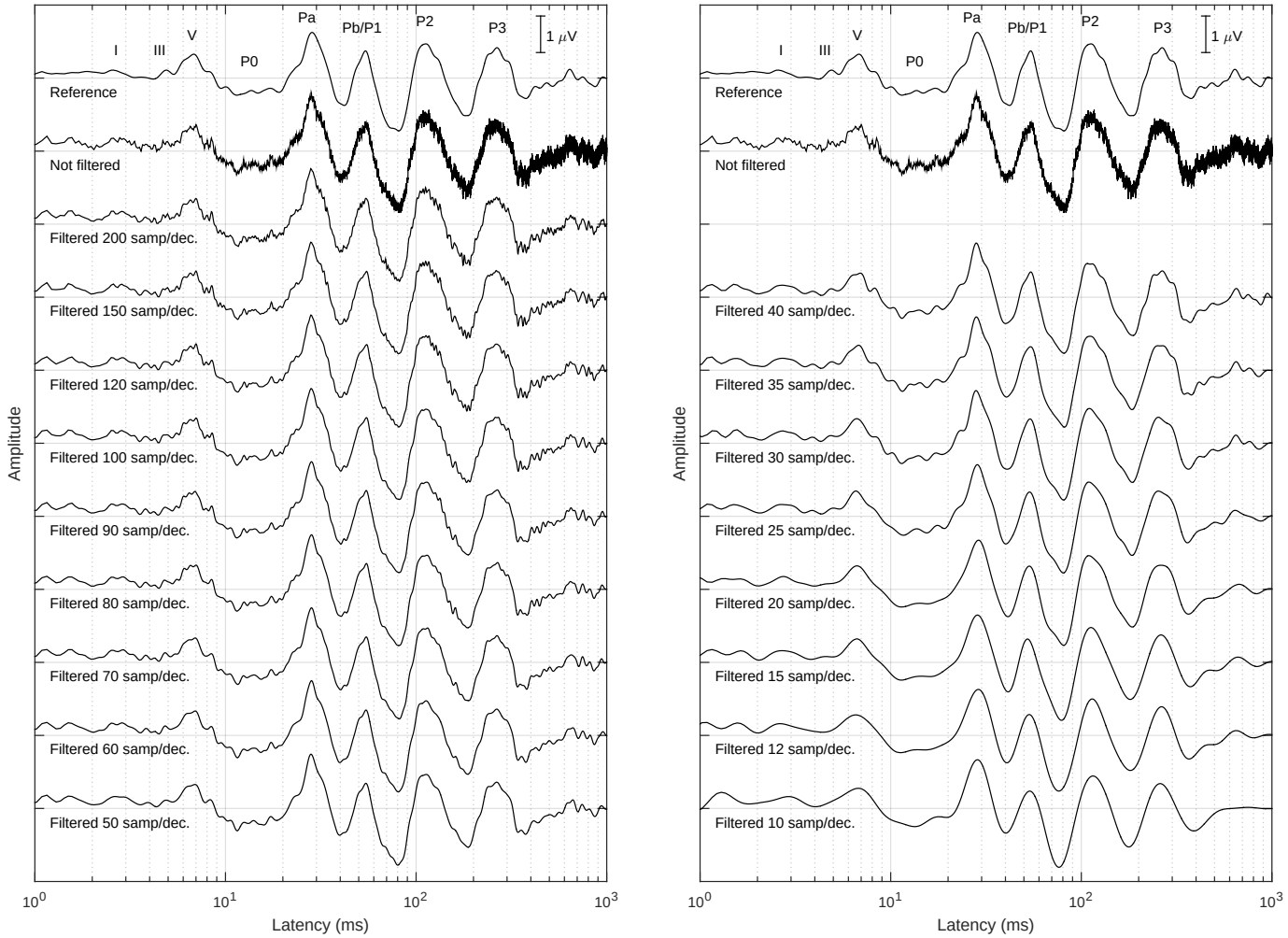

Figure 39: Comparison of the AEP responses filtered with different resolutions. Responses estimated from the synthetic EEG contaminated with pink noise.

12 Effect of the latency-dependent filtering with real responses

The effect of the latency-dependent filtering has been evaluated with real AEP responses. The AEP recording experiment involved 8 subjects. The EEGs have been recorded with 6 different stimulation rates between 1.39 Hz and 44.44 Hz. The inter-stimulus interval (ISI) of the stimulation signal followed a uniform distribution between 15-30 ms (for the average stimulation rate of 44.44 Hz) and 480-960 ms (for 1.39 Hz). The stimulation consisted in a sequence of rarefaction clicks presented at the instants defined by the stimulation sequence according to each ISI interval. The pulses had a duration of 0.1 ms and were presented at 74 dB (Hearing Level).

The EEGs were recorded with electrodes located at the upper forehead (Fz, active), the right mastoid (Tp10, reference) and the middle forehead (Fpz, ground). An instrumentation preamplifier was used for acquiring the EEGs. The amplified signal was digitized at a sampling rate of 44100 Hz with 16 bits/sample. The digitized EEG was low-pass filtered and down-sampled to 14700 Hz. The EEGs were recorded for each ISI condition during 684 seconds. Each EEG recording was organized in three blocks of 228 seconds. The estimation of the AEP responses was carried out with the iterative randomized stimulation and averaging (IRSA) procedure.

12.1 Responses for each subject before and after latency-dependent filtering

Figures 40, 41 and 42 represent the AEP responses obtained for each subject under the different ISI conditions. For each stimulation condition, the responses estimated from each portion of the EEG are plotted separately in order to provide information about the consistency of the waves. The panels in the left show the not-filtered AEP responses, while panels in the right show the filtered AEP responses. The latency-dependent filtering has been carried out with a resolution of 40 samples per decade. The orthonormal filtering matrix V_r has been prepared for a response length of 1 second (i.e. 14700 samples), and it is a 117×14700 matrix for the selected resolution. The compact representation for each AEP responses $\mathbf{x}_r = V_r \cdot \mathbf{x}$ contains 117 samples, and is transformed to the standard representation (14700 samples) with the matrix product $\mathbf{x}_{lp} = V_r^T \cdot \mathbf{x}_r$. The waves of the AEP response are labeled for the subject number 1.

As can be observed in these figures, the latency-dependent filtering significantly reduces the energy of the noise (the noise reduction is particularly evident at late latency) and preserves the waves of the AEP response.

12.2 Effect of the resolution in the latency-dependent filtering

The resolution of 40 samples/decade has been selected taking into account the standard configurations and protocols for recording ABR, MLR and CAEP responses. Obviously a different resolution affects to the number of samples required for the compact representation of the response $\mathbf{x}_r$, and also affects the preservation of the waves and the noise reduction. In order to analyze the effect of the resolution, the AEP responses of subject 1 have been latency-dependent filtered with resolutions between 200 samples/decade and 10 samples/decade.

Figures 43 and 44 show the AEP responses of subject 1 without filtering and latency-dependent filtered at the different resolutions. When high resolution is applied in the latency-dependent filtering, a better preservation of the waves is observed, but noise reduction is less efficient. As the resolution decreases, noise reduction is more effective but at very low resolution (20 and 10 samples/decade) the AEP morphology is affected by the low-pass filtering. As observed in these plots, the resolution of 30 samples/decade preserves without distortion all the AEP waves. A resolution of 20 samples/decade is acceptable (all the waves are identifiable, even though some of them looks a little wider at this resolution). The resolution of 10 samples/decade strongly affects the waves morphology. However, this resolution could be useful for some applications, since it provides the most effective noise reduction and preserves the main waves of the response.

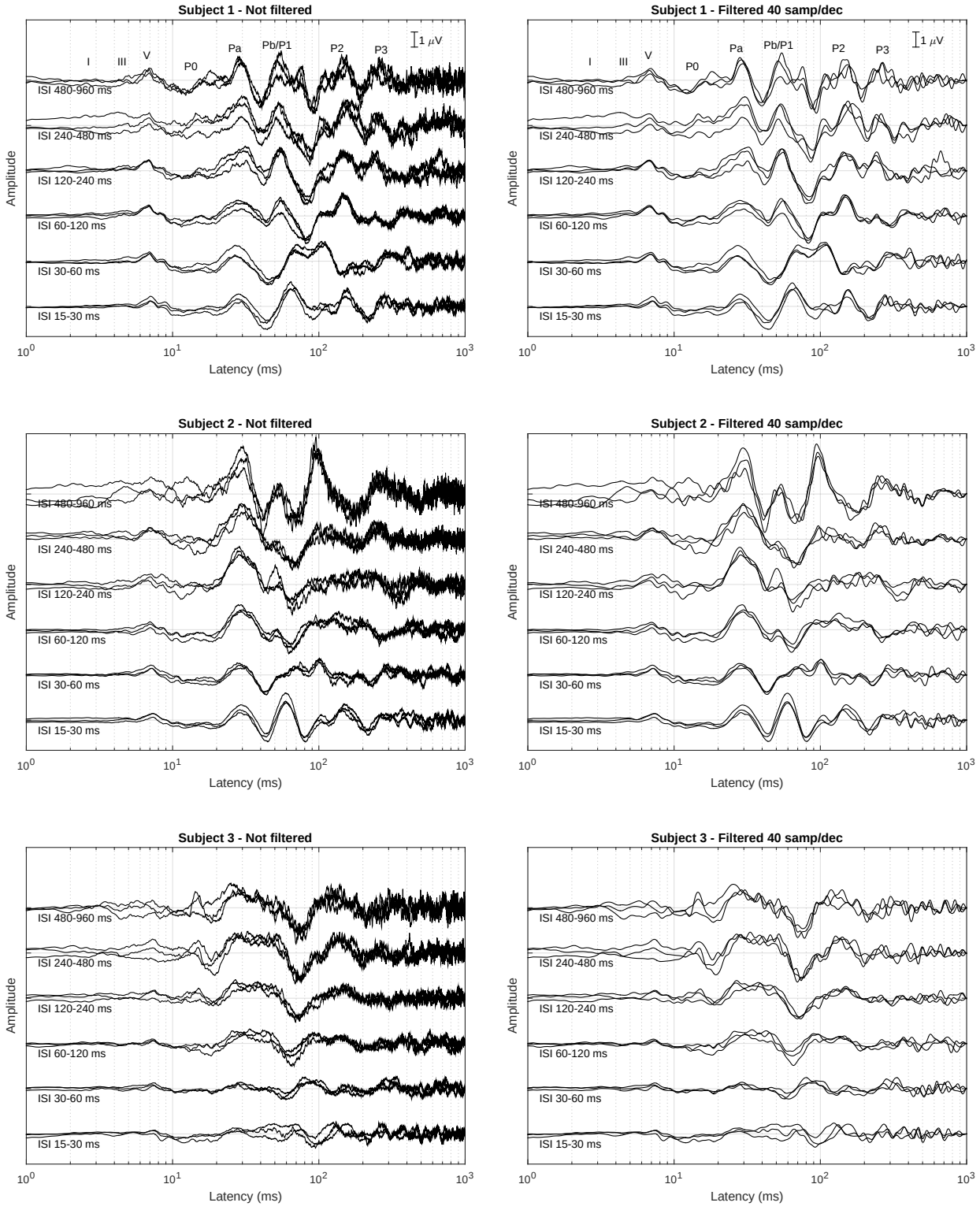

Figure 40: AEP responses, for subjects 1, 2 and 3, before (left) and after (right) the latency-dependent filtering with a resolution of 40 samples/decade.

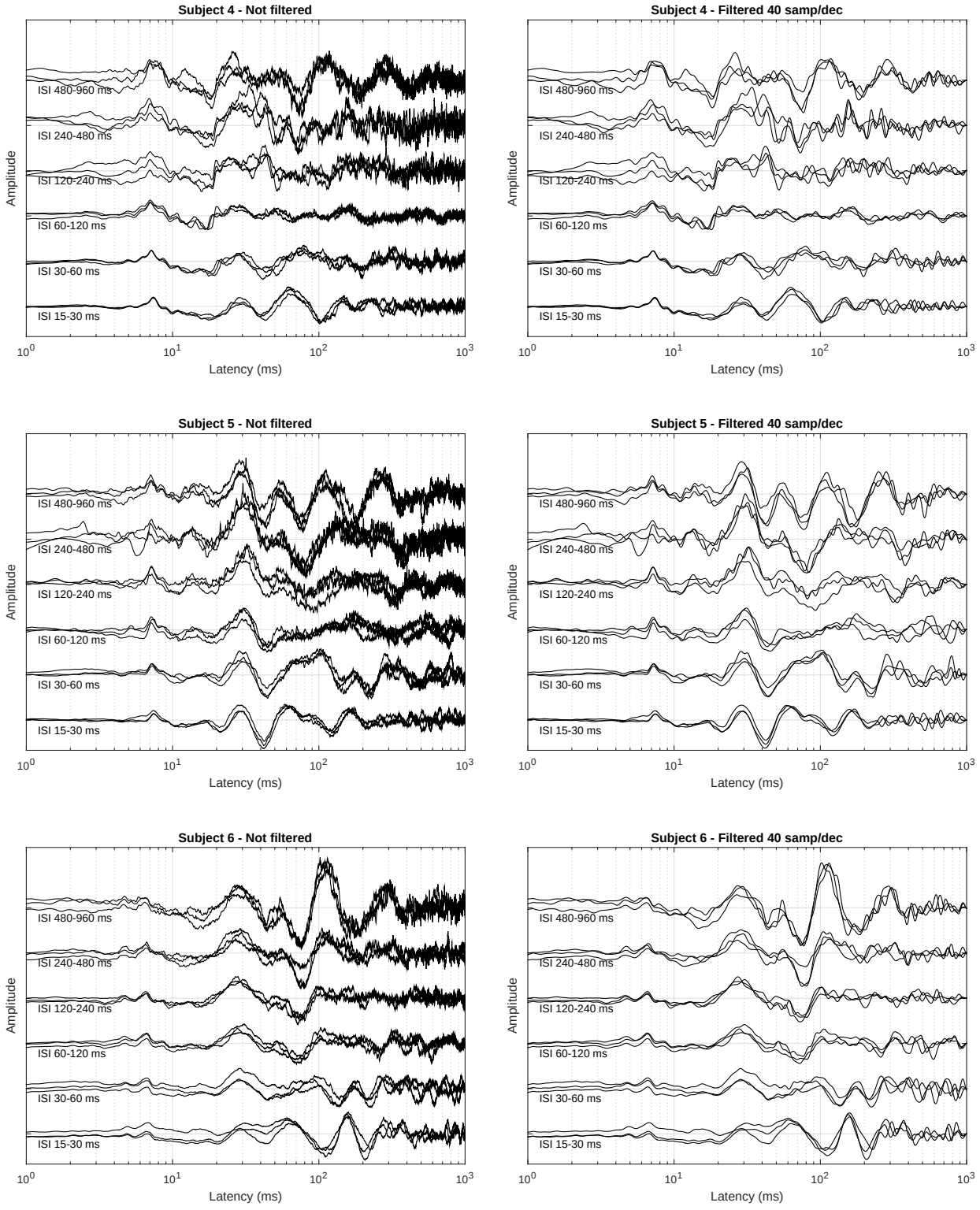

Figure 41: AEP responses, for subjects 4, 5 and 6, before (left) and after (right) the latency-dependent filtering with a resolution of 40 samples/decade.

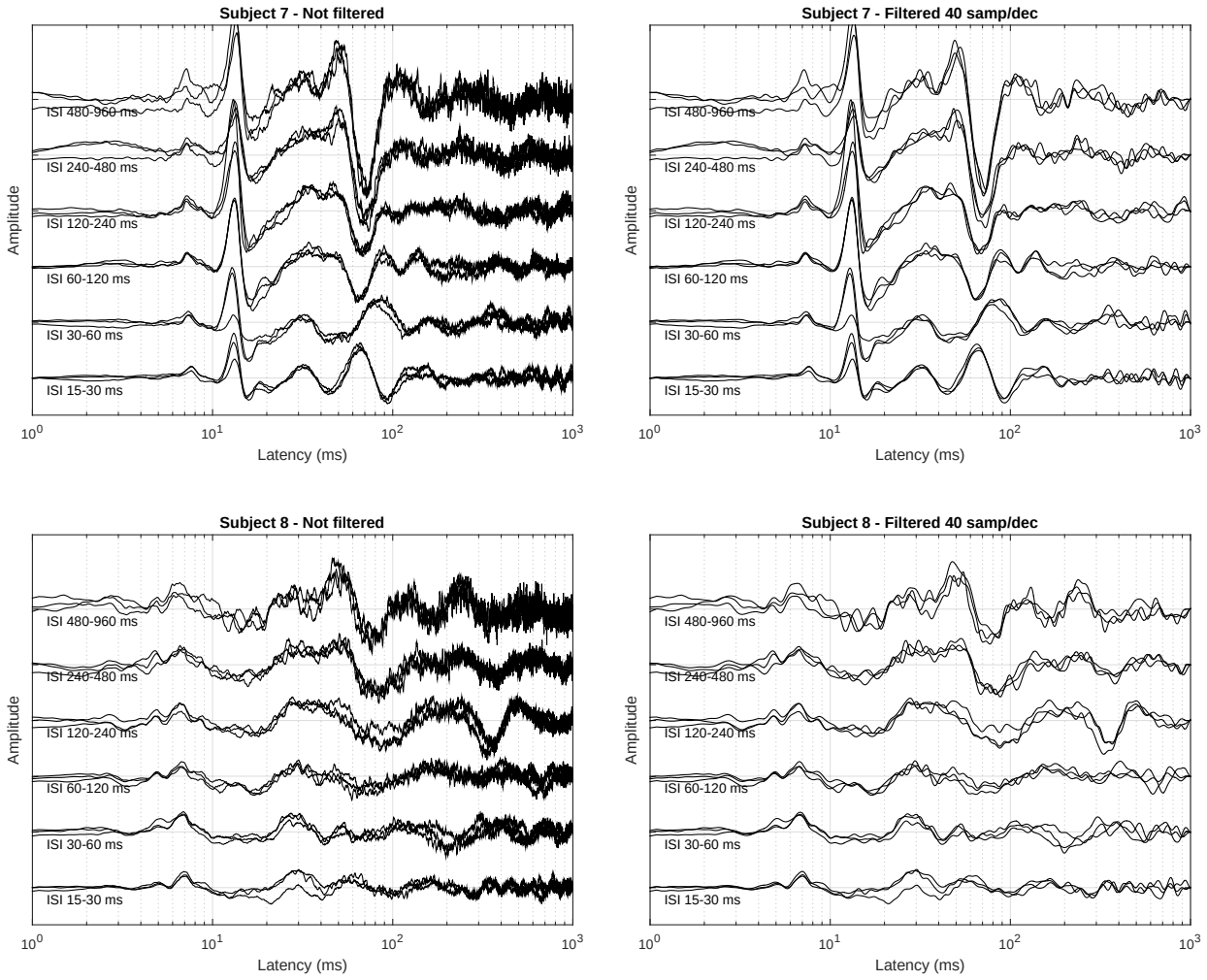

Figure 42: AEP responses, for subjects 7 and 8, before (left) and after (right) the latency-dependent filtering with a resolution of 40 samples/decade.

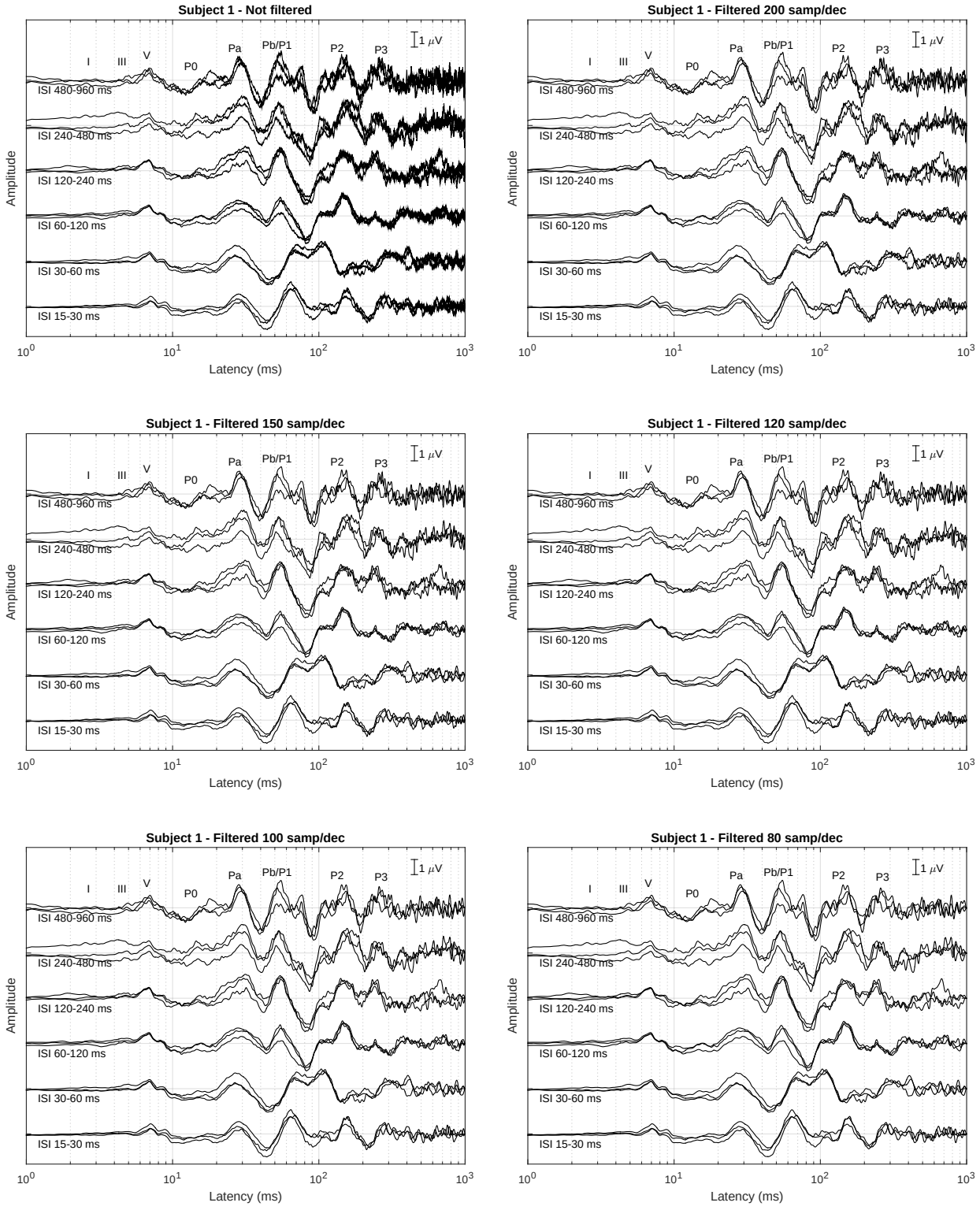

Figure 43: AEP responses for subject 1, filtered with different resolutions: not filtered and latency-dependent filtered with resolutions of 200, 150, 120, 100 and 80 samples/decade.

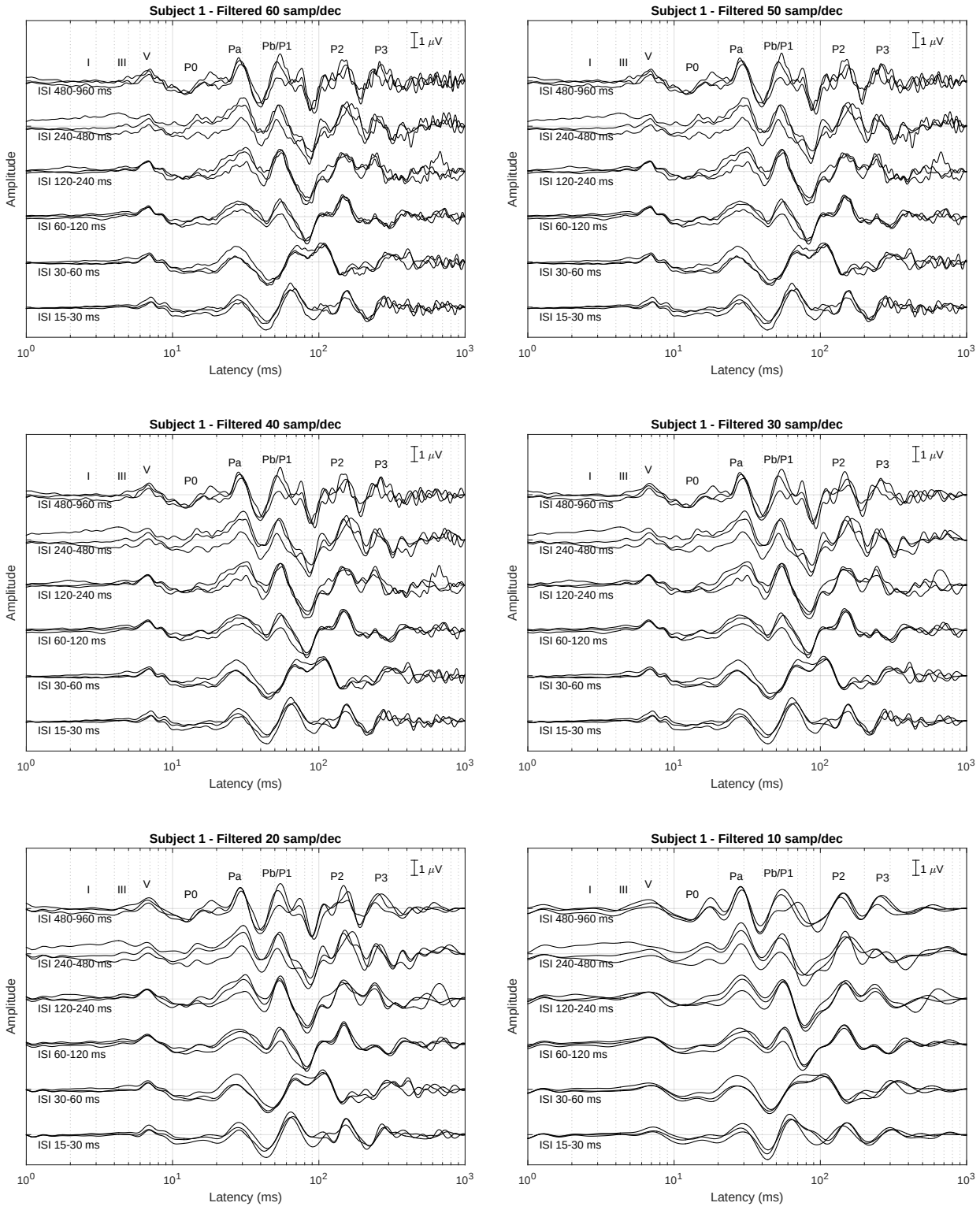

Figure 44: AEP responses for subject 1, filtered with resolutions of 60, 50, 40, 30, 20 and 10 samples/decade.

12.3 Estimation of the SNR for different resolutions

The efficiency of the latency-dependent filtering for noise reduction has been evaluated in terms of the SNR. The estimation of the SNR with real AEP responses is more difficult than in the case of simulations, since now the clean response is not available. Therefore, a reasonable response must be prepared to be used as reference (for estimating the energy of the signal and that of the noise). For each subject and stimulation condition there are three EEG portions, each one with a duration of 228 s. Each individual response (from each 228 s EEG portion recorded from each subject for each stimulation condition) has been compared with the corresponding average response (from the complete 684 s EEG) filtered with a resolution K_{dec}=40 samples/decade. This resolution has been selected for filtering the reference response based on the typical filter configurations used in conventional AEP recording. In addition to the subject-based estimations of SNR, a grand-average-based SNR has been estimated, using the responses from 228 s EEG portions averaged across subjects as individual responses, and those from 684 s EEG (also averaged across subjects, and also filtered with K_{dec}=40 samples/decade) as reference.

Table 2 shows the SNR results for the subject-based analysis with real AEP responses, while table 3 contains the results for the grand-average-based analysis. The subject-based results are presented in two sub-tables. In the upper one, the SNR results are presented for each subject (i.e. averaged for the different stimulation conditions). The last column contains the SNR averaged across subjects. The results are compared for different resolutions in the latency-dependent filtering, between K_{dec}=200 samples/decade and K_{dec}=5 samples/decade, including also the not-filtered condition. The table in the bottom includes, for each resolution, the mean and standard deviation of the SNR, the improvement provided by the latency-dependent filtering with respect to the not-filtered condition, and the p-value provided by the Student's t-test (probability of the null hypothesis of statistical independence of the observed improvement). The results for the grand-average-based analysis are also presented in two sub-tables. The upper one includes the SNR results for each stimulation condition (between ISI=15-30 ms and ISI=480-960 ms) for different resolutions, and the table in the bottom presents the average SNR (mean and standard deviation), the improvement with respect to the not-filtered condition and the p-value of the Student's t-test.

As can be observed in the table 2, in the subject-based analysis, the average SNR estimated for the not-filtered responses is 7.54 dB, and the highest SNR (10.74 dB) is achieved with a resolution of 15 samples/decade, which corresponds to an improvement of 3.20 dB. This improvement is statistically significant ($p < 10^{-7}$). In the grand-average analysis, the average across subjects provides a global improvement in the SNR for all the conditions. Since 8 subjects are included, the expected improvement would be $10\log_{10}(8)$ dB = 9.03 dB; however, the inter-subject variability results in a slight reduction of the signal energy, and the SNR improvement is smaller. In the grand-average-based analysis, the SNR improves from 13.44 dB (for the not-filtered condition) up to 15.93 dB (for a resolution of 25 samples/decade). The improvement is also statistically significant ($p \approx 0.01$).

The SNR results are also represented in the figure 45 (top panel) for both the grand-average-based (averaged for the six stimulation conditions) and the subject-based (averaged for the six stimulation conditions and the eight subjects) estimations, as a function of the resolution. In both cases the SNR progressively increases as the resolution decreases because the high-frequency noise is more effectively reduced by a more restrictive latency-dependent low-pass filtering. However, beyond a given resolution (25 samples/decade in the grand-average-based analysis, 15 samples/decade in the subject-based analysis), the SNR decreases due to the distortion caused by the application of a too restrictive low-pass filtering. Since the noise level is smaller in the grand-average responses, the highest SNR is achieved at a resolution slightly greater than in the subject-based analysis, (because, as expected, the most appropriate filtering depends on the noise level).

The application of a latency-dependent filtering to the reference responses with a resolution of 40 samples/decade was based on the expected frequency content of the AEP responses (according to the filter set-up in conventional AEP recording). In order to study the sensitivity to the resolution selected for filtering the reference, the SNR analysis has been repeated for different resolutions between 30 and 60 samples/decade. The bottom panel in the figure 45 provides a comparison of the effect of the resolution in the reference responses. According to this analysis, as the resolution decreases in the reference, the SNR for those responses estimated with higher resolutions (and the not-filtered estimations) decreases (because the noise level of the reference is smaller) and the SNR for those responses estimated with lower resolution increases (because the estimations are closer to the reference). However, the resolution providing the maximum SNR does not depends on the resolution used for the reference (25 samples/decade in the case of grand-average-based estimations, 15 samples/decade in the case of subject-based estimations).

13 Grand average of the responses of the complete auditory pathway

Figure 46 and 47 provide the grand-average of the AEP responses obtained from the eight subjects. In the first figure, the responses are filtered with a resolution of 40 samples/decade and reconstructed in the interval between 60 μs and 1 000 ms, while the second one presents the grand average responses without latency dependent filtering. The stimulation artifact in the interval 100-600 μs can be observed in the first decade.

Condition	Num. of Samples	SNR (dB) Subj.1	Subj.2	Subj.3	Subj.4	Subj.5	Subj.6	Subj.7	Subj.8	Average SNR (dB)
Not Filtered	14700	9.17	8.13	5.20	6.06	7.62	7.32	9.57	7.26	7.54
Lat.Dep.Filt. 200	446	9.53	8.96	5.90	6.97	8.26	7.80	10.61	8.52	8.32
Lat.Dep.Filt. 150	353	9.65	9.18	6.08	7.20	8.39	7.94	10.85	8.80	8.51
Lat.Dep.Filt. 120	294	9.76	9.39	6.24	7.41	8.48	8.10	11.01	9.02	8.68
Lat.Dep.Filt. 100	253	9.85	9.63	6.43	7.59	8.56	8.20	11.17	9.21	8.83
Lat.Dep.Filt. 90	231	10.00	9.76	6.59	7.85	8.66	8.36	11.38	9.35	9.00
Lat.Dep.Filt. 80	210	10.04	9.91	6.63	7.89	8.71	8.41	11.39	9.47	9.06
Lat.Dep.Filt. 70	187	10.27	10.20	6.93	8.15	8.82	8.78	11.67	9.66	9.31
Lat.Dep.Filt. 60	165	10.38	10.50	6.95	8.33	8.92	8.86	11.78	9.87	9.45
Lat.Dep.Filt. 50	141	10.70	10.98	7.41	8.67	9.12	9.20	12.18	10.10	9.80
Lat.Dep.Filt. 40	117	10.94	11.68	7.87	9.15	9.46	9.66	12.59	10.47	10.23
Lat.Dep.Filt. 35	104	10.96	11.58	7.96	9.21	9.58	9.73	12.55	10.49	10.26
Lat.Dep.Filt. 30	91	11.11	11.60	7.87	9.36	9.77	9.84	12.66	10.64	10.36
Lat.Dep.Filt. 25	78	11.21	11.71	7.94	9.34	9.87	10.06	12.78	10.68	10.45
Lat.Dep.Filt. 20	64	11.43	11.76	8.16	9.40	10.25	10.56	12.76	10.16	10.56
Lat.Dep.Filt. 15	50	11.74	11.78	8.51	9.73	10.34	10.76	12.70	10.33	10.74
Lat.Dep.Filt. 12	41	10.95	11.64	8.77	9.60	10.52	11.06	12.52	9.96	10.63
Lat.Dep.Filt. 10	35	10.11	11.12	8.90	9.58	10.29	10.82	11.28	9.81	10.24
Lat.Dep.Filt. 8	29	9.17	9.80	8.20	9.07	9.93	10.20	10.16	9.71	9.53
Lat.Dep.Filt. 7	25	8.67	9.31	8.07	8.67	8.68	9.97	9.12	9.16	8.96
Lat.Dep.Filt. 6	22	8.22	8.64	7.98	8.07	8.06	8.47	8.08	8.85	8.29
Lat.Dep.Filt. 5	19	7.47	8.08	7.23	7.11	6.82	6.65	7.12	7.19	7.21

Condition	Num. of Samples	SNR (dB) mean	(std.dev)	Improvement (dB)	p
Not Filtered	14700	7.54	(1.46)	0.00	-
Lat.Dep.Filt. 200	446	8.32	(1.47)	0.78	$1.441 \cdot 10^{-4}$
Lat.Dep.Filt. 150	353	8.51	(1.48)	0.97	$1.073 \cdot 10^{-4}$
Lat.Dep.Filt. 120	294	8.68	(1.47)	1.13	$7.850 \cdot 10^{-5}$
Lat.Dep.Filt. 100	253	8.83	(1.47)	1.29	$6.619 \cdot 10^{-5}$
Lat.Dep.Filt. 90	231	9.00	(1.47)	1.45	$3.947 \cdot 10^{-5}$
Lat.Dep.Filt. 80	210	9.06	(1.47)	1.52	$3.880 \cdot 10^{-5}$
Lat.Dep.Filt. 70	187	9.31	(1.46)	1.77	$1.533 \cdot 10^{-5}$
Lat.Dep.Filt. 60	165	9.45	(1.50)	1.91	$1.871 \cdot 10^{-5}$
Lat.Dep.Filt. 50	141	9.80	(1.50)	2.25	$9.076 \cdot 10^{-6}$
Lat.Dep.Filt. 40	117	10.23	(1.51)	2.69	$7.82 \cdot 10^{-6}$
Lat.Dep.Filt. 35	104	10.26	(1.45)	2.72	$4.605 \cdot 10^{-6}$
Lat.Dep.Filt. 30	91	10.36	(1.48)	2.82	$2.740 \cdot 10^{-6}$
Lat.Dep.Filt. 25	78	10.45	(1.50)	2.91	$1.643 \cdot 10^{-6}$
Lat.Dep.Filt. 20	64	10.56	(1.43)	3.02	$2.195 \cdot 10^{-7}$
Lat.Dep.Filt. 15	50	10.74	(1.32)	3.20	$9.145 \cdot 10^{-8}$
Lat.Dep.Filt. 12	41	10.63	(1.19)	3.09	$2.994 \cdot 10^{-6}$
Lat.Dep.Filt. 10	35	10.24	(0.81)	2.70	$9.912 \cdot 10^{-5}$
Lat.Dep.Filt. 8	29	9.53	(0.68)	1.99	$1.728 \cdot 10^{-3}$
Lat.Dep.Filt. 7	25	8.96	(0.57)	1.41	0.02059
Lat.Dep.Filt. 6	22	8.29	(0.32)	0.75	0.1842
Lat.Dep.Filt. 5	19	7.21	(0.43)	-0.33	0.5327

Table 2: SNR estimations as a function of the resolution in the latency-dependent filtering. Subject-based estimations of the SNR. Top table: results for each subject; bottom table: statistical results including the mean, the standard deviation, the improvement with respect to the not-filtered condition and the p-value of the Student's t-test.

Condition	Num. of Samples	SNR (dB) ISI (ms) 15-30	ISI (ms) 30-60	ISI (ms) 60-120	ISI (ms) 120-240	ISI (ms) 240-480	ISI (ms) 480-960	Average SNR (dB)
Not Filtered	14700	13.05	14.11	11.94	13.79	13.21	14.55	13.44
Lat.Dep.Filt. 200	446	13.26	14.42	12.64	14.50	14.19	16.14	14.19
Lat.Dep.Filt. 150	353	13.30	14.50	12.77	14.67	14.40	16.48	14.36
Lat.Dep.Filt. 120	294	13.34	14.56	12.87	14.88	14.70	16.69	14.51
Lat.Dep.Filt. 100	253	13.39	14.61	13.04	15.02	14.90	16.96	14.65
Lat.Dep.Filt. 90	231	13.62	14.70	13.14	15.08	15.15	17.13	14.80
Lat.Dep.Filt. 80	210	13.54	14.68	13.20	15.23	15.22	17.34	14.87
Lat.Dep.Filt. 70	187	13.72	14.76	13.29	15.41	15.56	17.46	15.03
Lat.Dep.Filt. 60	165	13.76	14.87	13.34	15.48	15.62	17.81	15.15
Lat.Dep.Filt. 50	141	14.06	14.95	13.57	15.72	16.25	18.18	15.45
Lat.Dep.Filt. 40	117	14.41	15.06	13.74	15.89	16.52	18.72	15.72
Lat.Dep.Filt. 35	104	14.28	14.93	13.68	15.96	16.78	18.85	15.75
Lat.Dep.Filt. 30	91	14.34	15.17	13.72	16.02	17.20	18.91	15.89
Lat.Dep.Filt. 25	78	14.33	15.03	13.76	16.03	17.17	19.25	15.93
Lat.Dep.Filt. 20	64	14.57	15.00	13.39	16.09	16.99	18.77	15.80
Lat.Dep.Filt. 15	50	14.36	14.74	13.13	16.15	16.87	19.02	15.71
Lat.Dep.Filt. 12	41	14.68	14.23	12.96	16.08	17.30	18.32	15.60
Lat.Dep.Filt. 10	35	13.46	12.91	12.71	15.57	16.54	17.46	14.77
Lat.Dep.Filt. 8	29	12.54	13.12	11.42	13.48	14.47	16.12	13.53
Lat.Dep.Filt. 7	25	11.78	11.11	11.97	11.77	14.02	15.05	12.62
Lat.Dep.Filt. 6	22	7.06	9.53	12.15	11.11	13.84	12.03	10.95
Lat.Dep.Filt. 5	19	7.29	8.79	11.51	10.39	9.82	6.89	9.12

Condition	Num. of Samples	SNR (dB) mean	(std.dev)	Improvement (dB)	p
Not Filtered	14700	13.44	(0.92)	0.00	-
Lat.Dep.Filt. 200	446	14.19	(1.20)	0.75	$1.418 \cdot 10^{-2}$
Lat.Dep.Filt. 150	353	14.36	(1.28)	0.92	$1.359 \cdot 10^{-2}$
Lat.Dep.Filt. 120	294	14.51	(1.34)	1.07	$1.223 \cdot 10^{-2}$
Lat.Dep.Filt. 100	253	14.65	(1.39)	1.21	$1.159 \cdot 10^{-2}$
Lat.Dep.Filt. 90	231	14.80	(1.40)	1.36	$7.979 \cdot 10^{-3}$
Lat.Dep.Filt. 80	210	14.87	(1.48)	1.43	$1.026 \cdot 10^{-2}$
Lat.Dep.Filt. 70	187	15.03	(1.49)	1.59	$7.615 \cdot 10^{-3}$
Lat.Dep.Filt. 60	165	15.15	(1.59)	1.71	$8.18 \cdot 10^{-3}$
Lat.Dep.Filt. 50	141	15.45	(1.67)	2.01	$6.842 \cdot 10^{-3}$
Lat.Dep.Filt. 40	117	15.72	(1.77)	2.28	$5.983 \cdot 10^{-3}$
Lat.Dep.Filt. 35	104	15.75	(1.89)	2.31	$8.857 \cdot 10^{-3}$
Lat.Dep.Filt. 30	91	15.89	(1.92)	2.45	$7.728 \cdot 10^{-3}$
Lat.Dep.Filt. 25	78	15.93	(2.03)	2.49	$1.004 \cdot 10^{-2}$
Lat.Dep.Filt. 20	64	15.80	(1.91)	2.36	$7.892 \cdot 10^{-3}$
Lat.Dep.Filt. 15	50	15.71	(2.09)	2.27	$1.437 \cdot 10^{-2}$
Lat.Dep.Filt. 12	41	15.60	(2.01)	2.16	$1.913 \cdot 10^{-2}$
Lat.Dep.Filt. 10	35	14.77	(2.02)	1.33	0.1109
Lat.Dep.Filt. 8	29	13.53	(1.63)	0.09	0.8506
Lat.Dep.Filt. 7	25	12.62	(1.55)	-0.82	0.2417
Lat.Dep.Filt. 6	22	10.95	(2.37)	-2.49	$6.583 \cdot 10^{-2}$
Lat.Dep.Filt. 5	19	9.12	(1.80)	-4.32	$8.077 \cdot 10^{-3}$

Table 3: SNR estimations as a function of the resolution in the latency-dependent filtering. Grand-average-based estimations of the SNR. Top table: results for each stimulation condition; bottom table: statistical results including the mean, the standard deviation, the improvement with respect to the not-filtered condition and the p-value of the Student's t-test.

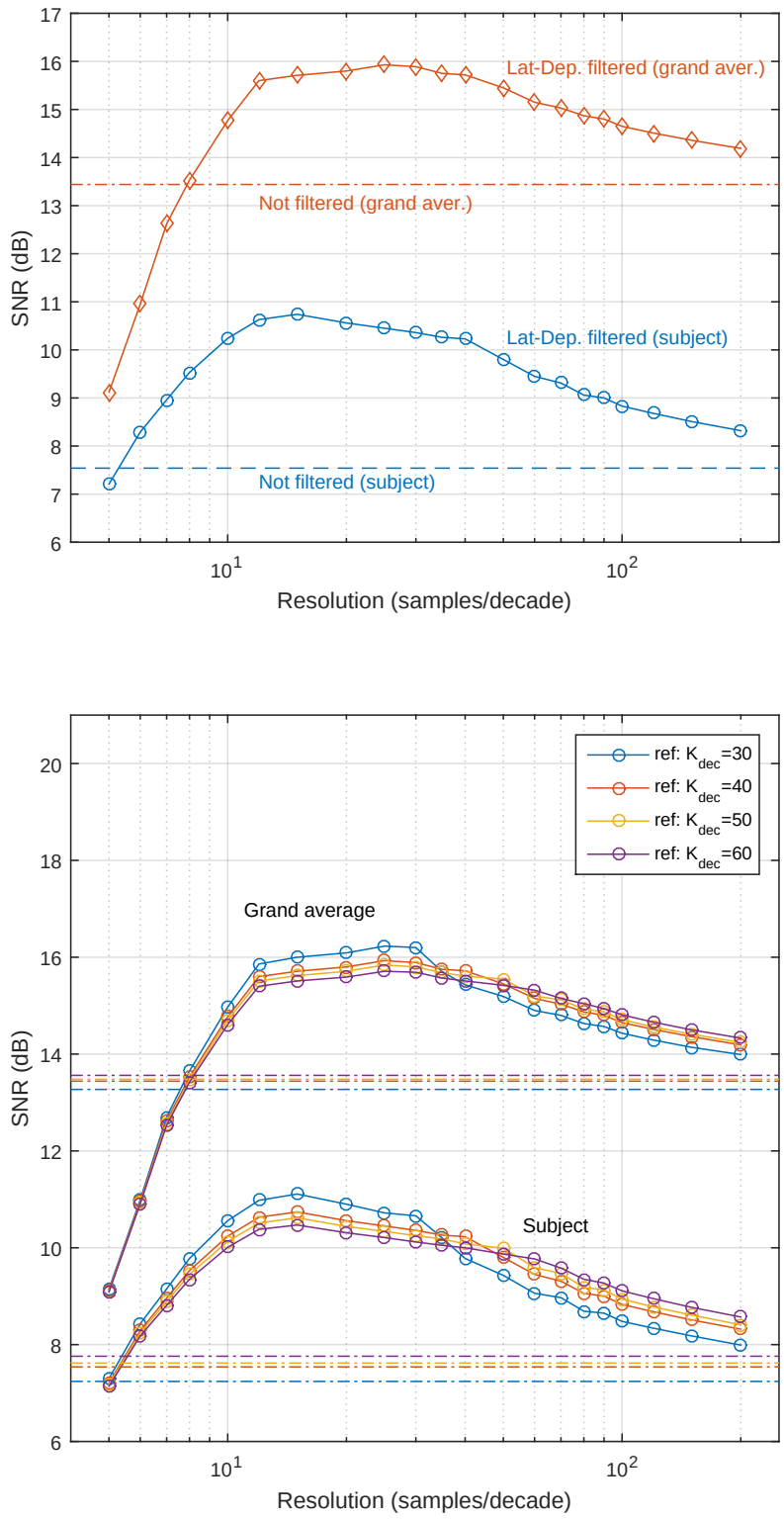

Figure 45: SNR of the real AEP responses as a function of the resolution K_{dec} used in the latency-dependent filtering, for both grand-average-based and subject-based estimations. The dashed lines represent the SNR for the not-filtered responses. The upper plot shows the analysis using a resolution of 40 samples/decade for filtering the reference responses. The lower plot compares the SNR estimations for different resolutions (between 30 and 60 samples/decade) in the filtering of the reference responses.

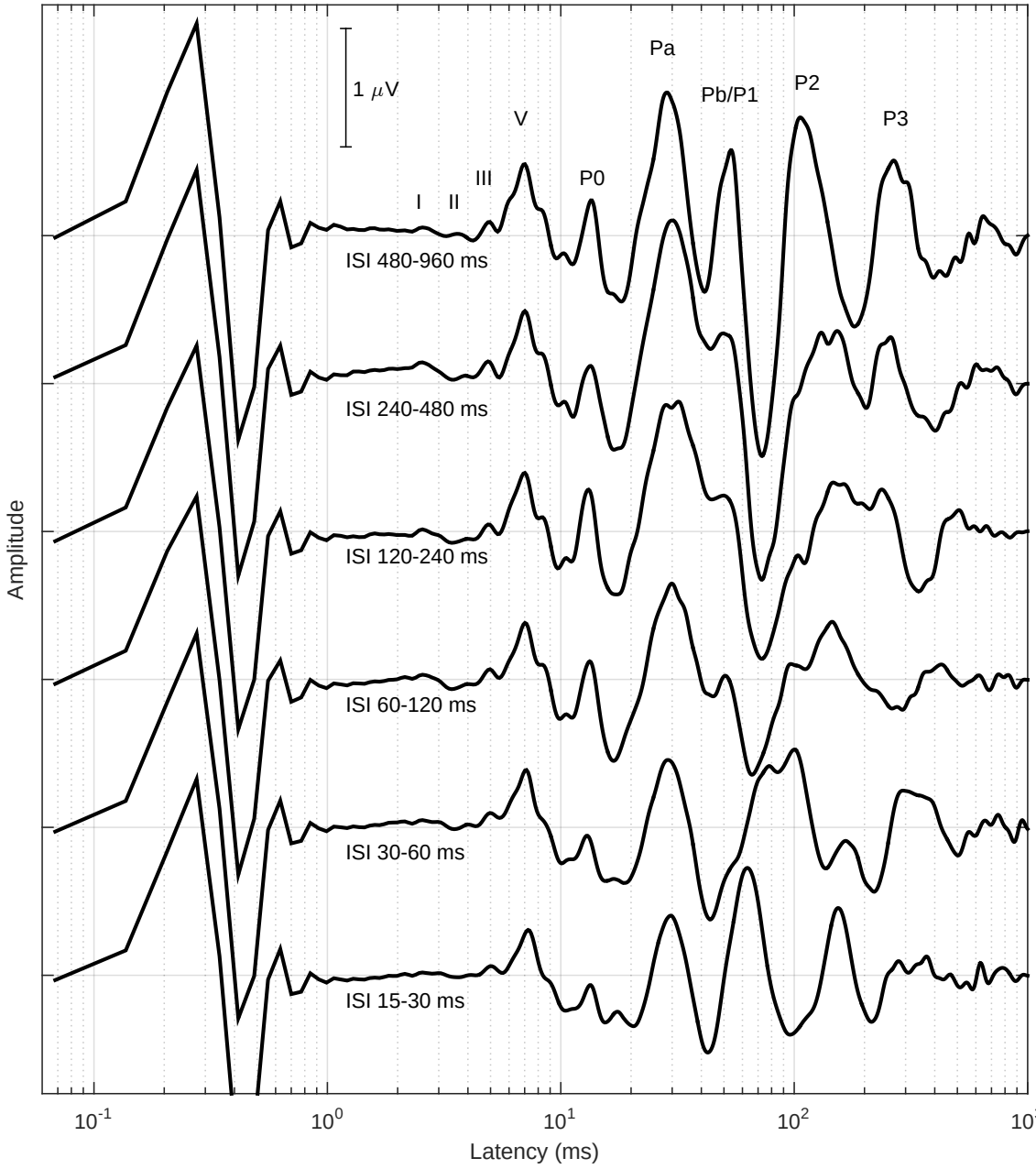

Figure 46: Response of the complete auditory pathway: Grand average of the AEP responses filtered with a resolution of 40 samples/decade reconstructed in the interval between 60 μs and 1 000 ms.

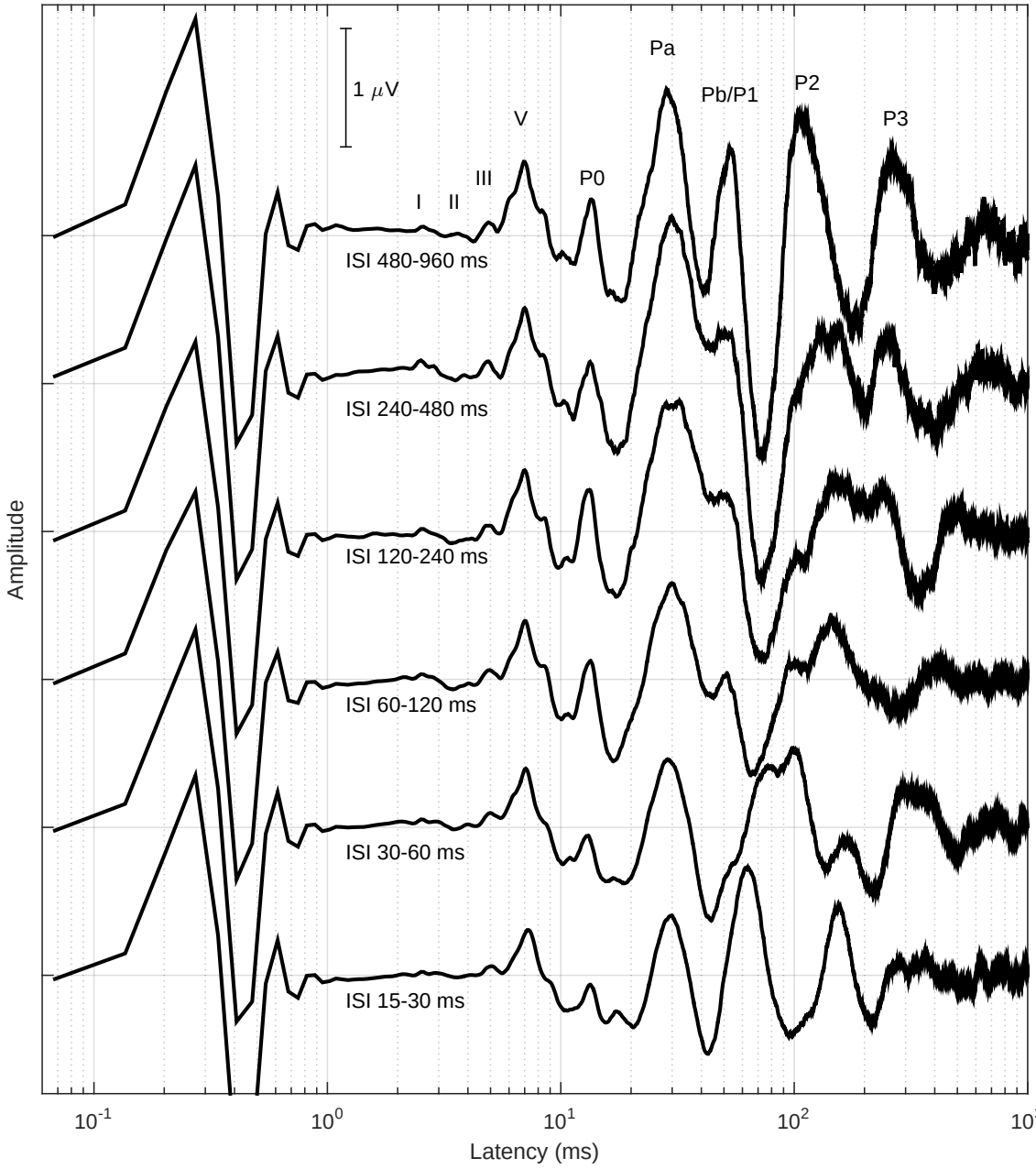

Figure 47: Response of the complete auditory pathway: Grand average of the AEP responses without latency-dependent filtering, represented in the in the interval between 60 μs and 1 000 ms.

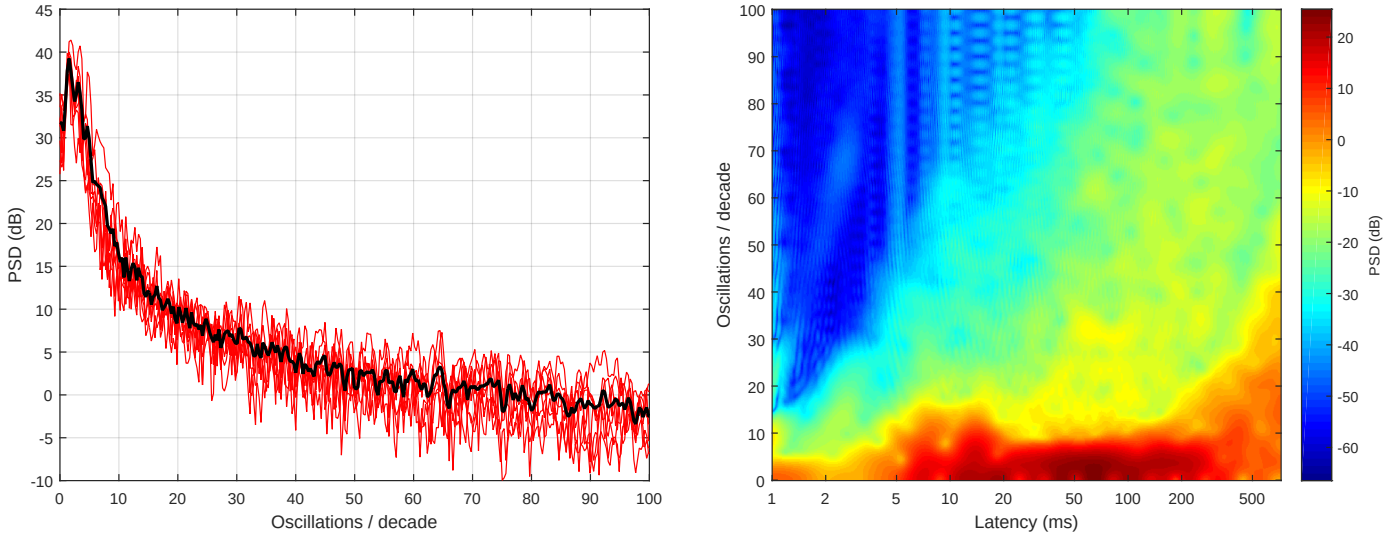

Figure 48: Spectral distribution of the energy in the AEP responses. Left panel: power spectral density as a function of the number of oscillations per decade for each subject (red lines) and average (black line). Right panel: average spectrogram of the AEP responses (power spectral density as a function of the latency and the number of oscillations per decade).

14 Spectral distribution of energy in the AEP responses

The expected morphology of the AEP responses at the different latency ranges (waves I, II, III, IV, V and VII in the ABR, N0, P0, Na, Pa, Nb, Pb in the MLR and P1, N1, P2, N2, P3 in the CAEP), provides a boundary in the number of oscillations per decade (around 10 oscillations/decade if all the waves are present). A detailed knowledge of the spectral distribution of the AEP responses is useful since it provides information about the spectral range in which the AEP response is expected and the portions that can be filtered for reducing the noise level with minimal distortion of the response.

Taking into account the quasi-stationary structure of the AEP response when it is represented in a compressed latency axis, the spectral analysis of the AEP responses has been performed including this compression. In order to perform this spectral analysis, the AEP responses have been reconstructed at the time instants $\{t_{rec}\}$ between 0.6 ms and 1000 ms (i.e. about 3.22 decades) with a resolution of 200 samples/decade. This provides a total of 645 time instant, uniformly distributed in the logarithmically scaled latency axis between 0.6 ms and 1000 ms. The fast Fourier transform of these samples provides the power spectral density corresponding to the log-scaled latency axis, and therefore, the frequency is expressed in oscillations per decade, being the sampling rate 200 oscillations/decade.

Figure 48 (left panel) shows the power spectral distribution in the AEP responses for each subject (red lines) and the average for all the subjects (black line), as a function of the number of oscillations per decade. As observed, most of the energy is concentrated in the lowest range of oscillations, with most of the energy in the range between 1 and 5 oscillations/decade, and an attenuation greater than 30 dB (with respect to the maximum) for more than 20 oscillations/decade. The range bellow 7 oscillations/decade contains the 95% of the energy, and the range below 20 oscillations/decade more than the 99%, being most of the remaining energy corresponding to background noise. This plot support the latency-dependent filtering using resolutions of 40 samples/decade or even lower. If the minimum range to be preserved is established in 10 oscillations/decade, since the proposed procedure preserves (without aliasing) the content below the 40% of the sampling rate, a resolution of 25 samples/decade would be enough.

In order to analyze the evolution of the frequency content with the latency, a spectrogram has been represented in the figure 48 (right panel). The spectrogram has been estimated as the average including all the responses from all the subjects. It represents the power spectral density as a function of the number of oscillations per decade (vertical axis) and the log-scaled latency (in the horizontal axis). The spectrogram confirms the quasi-stationary structure of the AEP response in the logarithmically scaled latency axis. However, it also shows some differences at the different portions of the AEP response, with more energy in the latency range between 10 ms and 200 ms, and some dispersion of the energy up to 15 oscillations/decade in the range between 5 and 20 ms (late ABRs and early MLRs). The energy above 10 oscillations observed at latency greater than 300 ms corresponds to brain waves, and, taking into account the frequency content, this activity probably corresponds to alpha or beta waves (the band 10-30 oscillations/decade at 500 ms corresponds to the band 8.6-26 Hz).

15 Comparison with conventional presentation of the auditory evoked potentials

In this work, the auditory evoked potentials are represented with the latency axis logarithmically scaled, in order to simultaneously represent all the components including the ABR, MLR and CAEP waves. However, AEP responses are conventionally represented with the latency axis linearly scaled, and with latency intervals and bandwidths adapted to each AEP component. This way, ABR responses are expected in the latency interval 1-10 ms and are band-pass filtered according to the expected frequency content in the band 100-3000 Hz; MLR responses are expected in the latency interval 10-100 ms and the frequency band 10-300 Hz; CAEP responses are expected in the latency interval 50-500 ms and the frequency band 1-30 Hz.

In order to compare the proposed latency dependent filtering with the conventional processing of the auditory evoked potentials, the responses provided by the proposed method have been compared with the not-filtered ones, and with those filtered with the conventional filtering configuration for the different AEP components (i.e. ABR, MLR and CAEP) in the corresponding latency intervals in linear scale.

Figure 49 represents the AEP responses of subject 1 for stimulation configuration with ISI=480-960 ms, estimated from a 228 s EEG portion (top figures) and a 684 s EEG portion (bottom figures). From left to right, the figures represent the ABR, MLR and CAEP portions of the AEP responses with the latency axis linearly scaled. Each figure includes three plots: the first one (blue) corresponds to the not filtered response; the second one (red) is the response filtered with the conventional band-pass filter for each component (100-3000 Hz for ABR; 10-300 Hz for MLR; 1-30 Hz for CAEP). The last plot (black) is the latency-dependent low-pass filtered response. The plots corresponding to 228 s EEG are more affected by noise than those corresponding to 684 s EEG, and therefore, allow a comparison of the effect of the conventional and the latency dependent filtering for different noise conditions.

As observed in these figures, the not-filtered responses are strongly affected by noise (particularly in the responses estimated from the 228 s EEG), and more specifically by high frequency noise at late latency. Conventional filtering provides some reduction of the high frequency noise, adapted to the analyzed portion of the response. However, the conventional filtering is too restrictive at the beginning of the responses (for example, in the MLR portion, the waves III and V are severely distorted; in the CAEP portion, the Pa and Pb/P1 components are also distorted) and too permissive at the end of the response (high frequency noise is not appropriately filtered at the end of the ABR, MLR or CAEP portions). Compared with the conventional filtering, the proposed latency-dependent low-pass filtering provides a more appropriate filtering of the response, preserving all the AEP components and providing a more effective reduction of the high frequency noise thanks to the continuous adaptation of the bandwidth.

Figures 50, 51 and 52 provide a comparison of the not-filtered, conventional filtered and latency-dependent filtered responses in conventional representations of the ABR, MLR and CAEP portions, for the grand average responses estimated for the different ISI configurations (480-960 and 240-480 ms in figure 50; 120-240 and 60-120 ms in figure 51; and 30-60 and 15-30 ms in figure 51). Since these plots are obtained as grand average of the responses from 8 subjects, are less affected by the noise, and the noise level also decreases as the stimulation rate increases (because the number of stimuli involved in the response estimation increases with the stimulation rate).

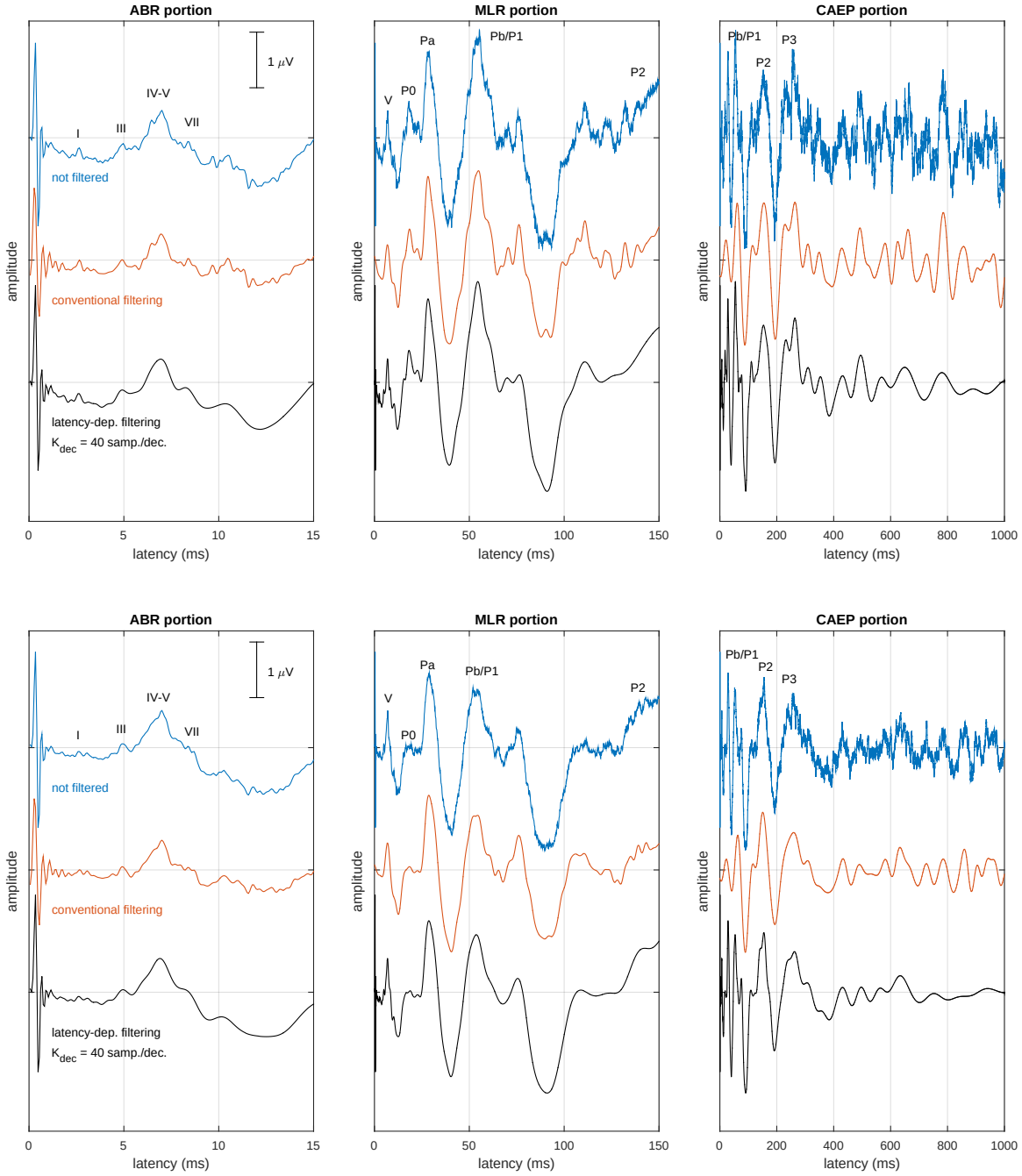

Figure 49: Conventional representation of the AEP responses for ABR, MLR and CAEP components. Responses from a 228 s (top) and 684 s (bottom) EEG portions from subject 1, with stimulation configured for ISI=480-960 ms. In each figure, the not-filtered response (top) the response filtered with conventional filtering (middle) and the latency-dependent filtered response (bottom) are compared.

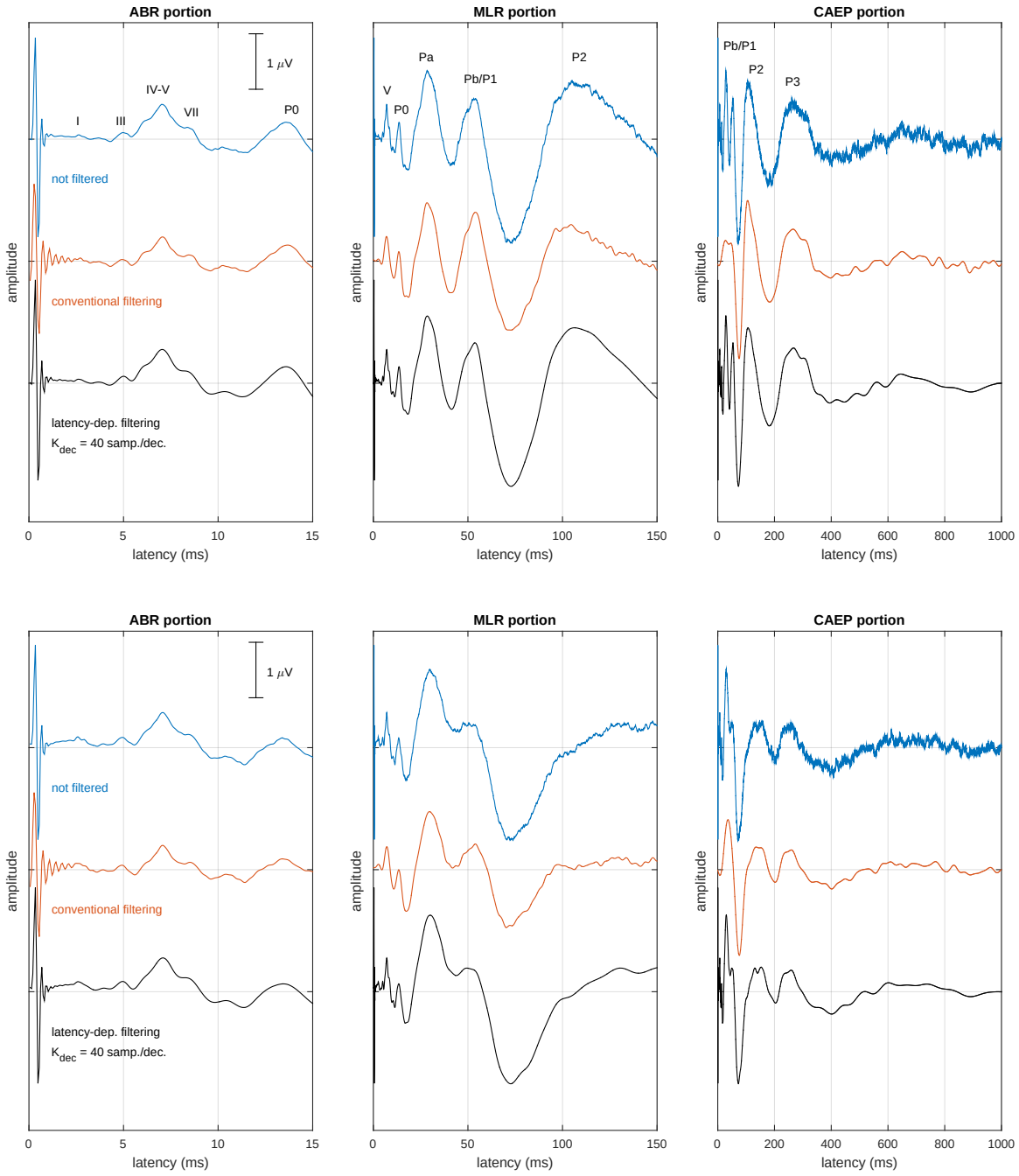

Figure 50: Conventional representation of the AEP responses for ABR, MLR and CAEP components. Grand average of the responses for stimulation configured with ISI=480-960 ms (top) and ISI=240-480 ms (bottom). In each figure, the not-filtered response (top) the response filtered with conventional filtering (middle) and the latency-dependent filtered response (bottom) are compared.

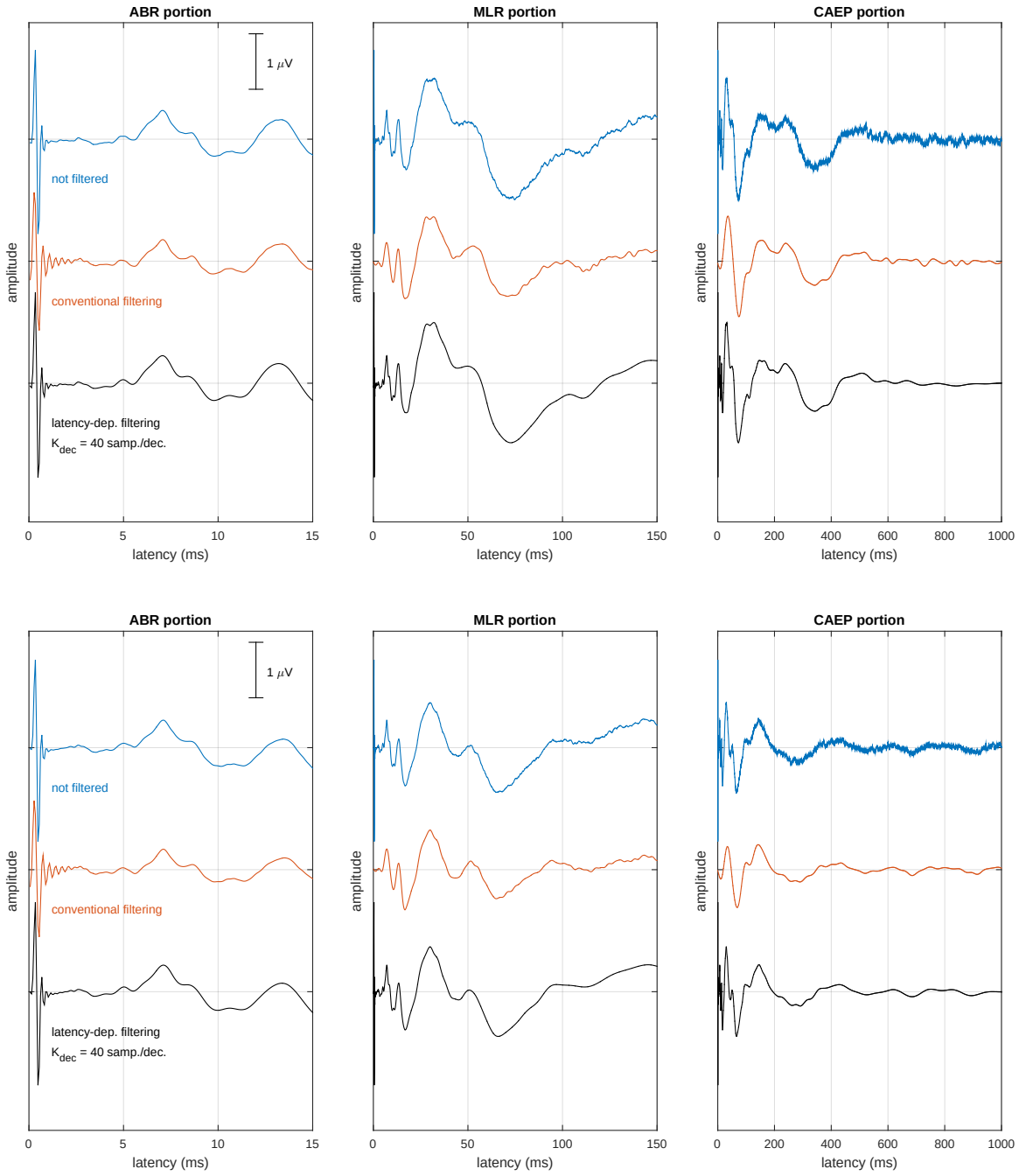

Figure 51: Conventional representation of the AEP responses for ABR, MLR and CAEP components. Grand average of the responses for stimulation configured with ISI=120-240 ms (top) and ISI=60-120 ms (bottom). In each figure, the not-filtered response (top) the response filtered with conventional filtering (middle) and the latency-dependent filtered response (bottom) are compared.

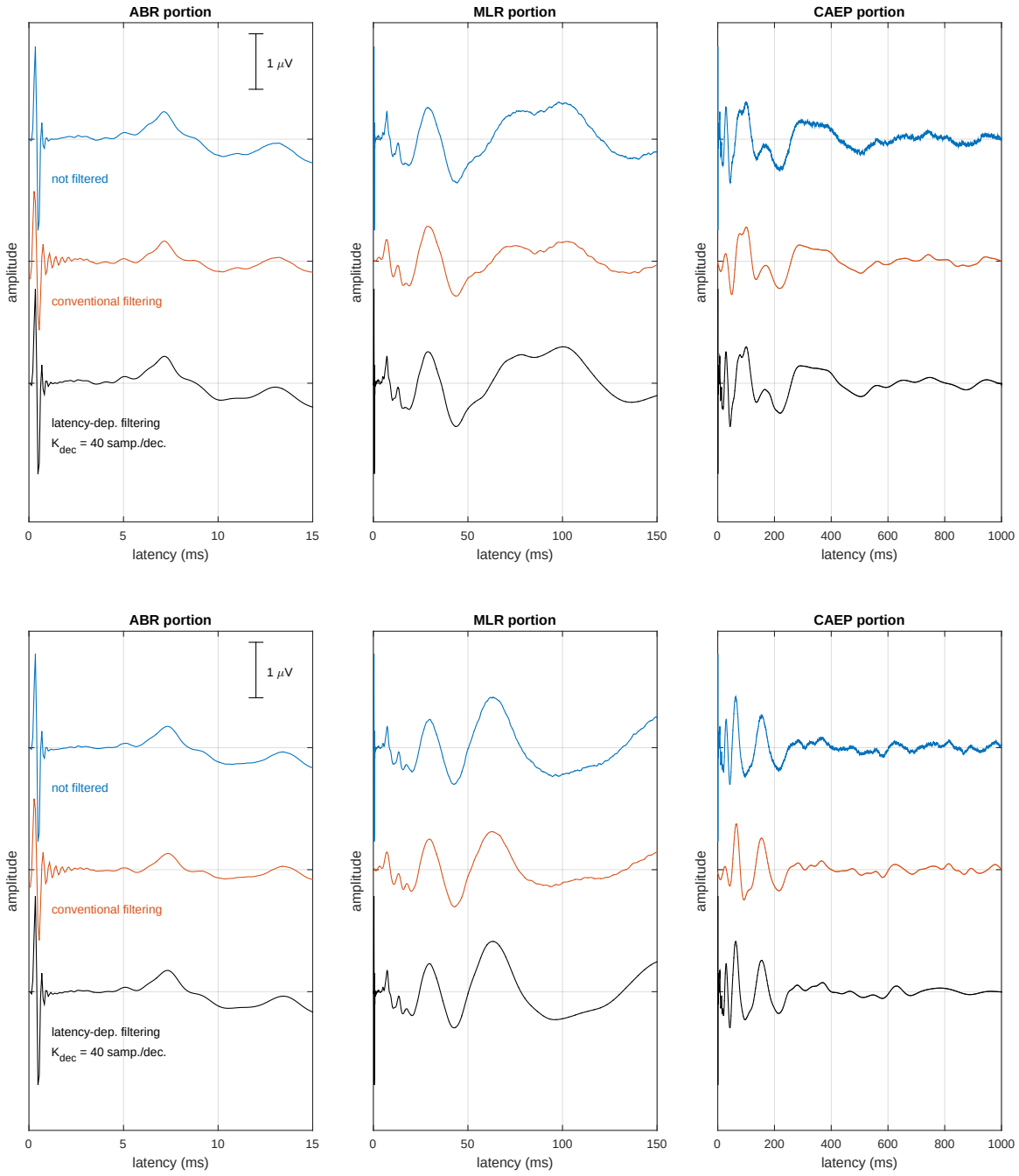

Figure 52: Conventional representation of the AEP responses for ABR, MLR and CAEP components. Grand average of the responses for stimulation configured with ISI=30-60 ms (top) and ISI=15-30 ms (bottom). In each figure, the not-filtered response (top) the response filtered with conventional filtering (middle) and the latency-dependent filtered response (bottom) are compared.

16 Code of a MatLab / Octave script for testing the latency-dependent filtring and down-sampling

The following MatLab / Octave code is a script that reads some AEP responses and applies a latency-dependent filtering and down-sampling according to the selected resolution. It can be used to test the proposed procedures and the MatLab / Octave functions provided in this manuscript, and to compare the results obtained for different resolutions in the latency-dependent filtering.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% script_LatDepFiltering.m
% This script provides a demonstration of the latency-dependent filtering
% * Some AEP responses (from EEGs recorded at different ISI) are loaded
% * The orthonormalized latency-dependent filtering and down-sampling
%     matrix V_r is generated for the selected resolution Kdec (Jr x J matrix)
% * The compact representation is estimated: x_r = V_r * x  (Jr samples)
% * The filtered signal is estimated for each sample in the original
%     representation: x_lp = V_r' * x_r  (J samples)
% * The matrix providing reconstruction at at specific latency values is
%     also generated: V_rec (Jrec x Jr matrix)
% * The filtered signal, reconstructed at the specific latency values, is
%     estimated: x_rec = V_rec * x_r  (Jrec samples)
% * The different responses are represented in figures 1, 2, 3, 4
%    not filtered:            x (J samples)    -> figure 1
%    compact representation:  x_r (Jr samples) -> figure 2
%    filtered at standard representation: x_lp (J samples) -> figure 3
%    filtered at specific latency values: x_rec (J_rec smaples) -> figure 4
% * The rows of the matrix (i.e. the orthonormal functions of the basis)
% are also represented in figures 5 (for linearly scaled latency axis) and
% 6 (logarithmically scaled latency axis).
%
% This script requires the functions:
%     * Basis_LinLog_RRC.m
%     * Basis_reconstruction.m
% This script also requires the data file (containing 18 AEP responses):
%     * data_subject_1.mat
% The script is configured by selecting appropriate resolution in the
% latency dependent filtering (Kdec) and reconstruction resolution
% (Rec_res). "Kdec" should be in the range 5 to 200; "Rec_res" should be
% greater than Kdec (recommended Rec_res=2*Kdec at least).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear; clc; warning('off','all');

Kdec=25;                % number of samples/decade in the latency-dependent filtering
Rec_res=200;            % reconstruction resolution in samples/decade

load data_subject_1.mat  % Loading the responses: AEP_resp (18 AEP responses); tms (time in ms)
J=length(tms);           % number of samples in the response
Fs=1000/(tms(2)-tms(1)); % sampling rate

Vr=Basis_LinLog_RRC(J,Kdec);  % orthonormalized matrix for lat-dep-filt and down-sampling
Jr=length(Vr(:,1));      % number of samples in the compact representation
xr=Vr*AEP_resp;          % compact representation of the lat-dep filtered responses
xlp=Vr'*xr;              % lat-dep filtered responses at standard representation
tr0ms=1.0; tr1ms=1000;   % limits of the latency interval for reconstruction, in ms
trms=tr0ms*10.^(0:1/Rec_res:log10(tr1ms/tr0ms))'; % specific latency values for reconstruction (in ms)
tr=trms/1000;            % specific latency values for reconstrution (in s)
Vrec=Basis_reconstruction(Vr,Fs,tr); % matrix providing reconstruction at tr
Jrec=length(Vrec(:,1));  % number of reconstructed samples
xrec=Vrec*xr;            % lat-dep filtered responses reconstructed at tr
```

```
% preparing separation between responses for the plots
SEP=3;                  % standard separation
shift_J=ones(J,1)*[0 0 0 1 1 1 2 2 2 3 3 3 4 4 4 5 5 5]*SEP; % vertical position of each AEP response
shift_Jr=ones(Jr,1)*[0 0 0 1 1 1 2 2 2 3 3 3 4 4 4 5 5 5]*SEP;
shift_Jrec=ones(Jrec,1)*[0 0 0 1 1 1 2 2 2 3 3 3 4 4 4 5 5 5]*SEP;

% NOT FILTERED RESPONSES
figure(1); clf;
semilogx(tms,AEP_resp+shift_J,'-r');
xlim([1 1000]); ylim([-2 6*SEP]); grid on
title(sprintf('Not filtered:   x  (%d samp.)',J));
xlabel('Latency (ms)'); ylabel('Amplitude')
text(1.2,-0.6+SEP*5,'ISI 480-960 ms'); text(1.2,-0.6+SEP*4,'ISI 240-480 ms')
text(1.2,-0.6+SEP*3,'ISI 120-240 ms'); text(1.2,-0.6+SEP*2,'ISI 60-120 ms')
text(1.2,-0.6+SEP*1,'ISI 30-60 ms'); text(1.2,-0.6+SEP*0,'ISI 15-30 ms')

% FILTERED RESPONSES: COMPACT REPRESENTATION
figure(2); clf;
plot(xr+shift_Jr*10,'.-r');
xlim([1 Jr]); ylim([-20 60*SEP]); grid on
title(sprintf('Lat-dep filtered (%.1f samp./dec.):   x_r=V_r·x  (compact repres, %d samp.)',Kdec,Jr));
xlabel('Sample index in the compact representation'); ylabel('Amplitude')
text(4,-5+SEP*50,'ISI 480-960 ms'); text(4,-5+SEP*40,'ISI 240-480 ms')
text(4,-5+SEP*30,'ISI 120-240 ms'); text(4,-5+SEP*20,'ISI 60-120 ms')
text(4,-5+SEP*10,'ISI 30-60 ms'); text(4,-5+SEP*0,'ISI 15-30 ms')

% FILTERED RESPONSES: RECONSTRUCTED AT ORIGINAL REPRESENTATION
figure(3); clf;
semilogx(tms,xlp+shift_J,'-r');
xlim([1 1000]); ylim([-2 6*SEP]); grid on
title(sprintf('Lat-dep filtered (%.1f samp./dec.):   x_{lp}=V_r^T·x_r  (%d samp.)',Kdec,J));
xlabel('Latency (ms)'); ylabel('Amplitude')
text(1.2,-0.6+SEP*5,'ISI 480-960 ms'); text(1.2,-0.6+SEP*4,'ISI 240-480 ms')
text(1.2,-0.6+SEP*3,'ISI 120-240 ms'); text(1.2,-0.6+SEP*2,'ISI 60-120 ms')
text(1.2,-0.6+SEP*1,'ISI 30-60 ms'); text(1.2,-0.6+SEP*0,'ISI 15-30 ms')

% FILTERED RESPONSES: RECONSTRUCTED AT SPECIFIC LATENCY VALUES tr
figure(4); clf;
semilogx(trms,xrec+shift_Jrec,'-r');
xlim([1 1000]); ylim([-2 6*SEP]); grid on
title(sprintf('Lat-dep filtered (%.1f samp./dec.):   x_{rec}=V_{rec}·x_r  (%d samp.)',Kdec,Jrec));
xlabel('Latency (ms)'); ylabel('Amplitude')
text(1.2,-0.6+SEP*5,'ISI 480-960 ms'); text(1.2,-0.6+SEP*4,'ISI 240-480 ms')
text(1.2,-0.6+SEP*3,'ISI 120-240 ms'); text(1.2,-0.6+SEP*2,'ISI 60-120 ms')
text(1.2,-0.6+SEP*1,'ISI 30-60 ms'); text(1.2,-0.6+SEP*0,'ISI 15-30 ms')

% ORTHONORMALIZED MATRIX PROVIDING LAT-DEP-FILTERING
figure(5); clf;
h0=Vr(round(Jr/2),:);
y0=min(h0)-0.6*(max(h0)-min(h0)); y1=max(h0)+0.6*(max(h0)-min(h0));
plot(tms,Vr','r',tms,h0,'.-k')
xlim([0 1000]); ylim([y0 y1]); grid on;
xlabel('Latency (ms)'); ylabel('Amplitude')
title('Rows of the filtering matrix (functions of the basis)')
% ORTHONORMALIZED MATRIX PROVIDING LAT-DEP-FILTERING (Log-scaled latency axis)
figure(6); clf;
semilogx(tms,Vr','r',tms,h0,'.-k')
xlim([1 1000]); ylim([y0 y1]); grid on;
xlabel('Latency (ms)'); ylabel('Amplitude')
title('Rows of the filtering matrix (functions of the basis)')

return;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```