WIENER FILTERING IN THE WINDOWED DFT DOMAIN

Siouar Bensaid, Dirk Slock

EURECOM, Mobile Communications Dept.
Campus SophiaTech, 450 route des Chappes, 06410 Biot Sophia Antipolis, France
Email: {siouar.bensaid, dirk.slock}@eurecom.fr

ABSTRACT

We focus on the use of windows in the frequency domain processing of data for the purpose of Wiener filtering. Classical frequency domain asymptotics replace linear convolution by circulant convolution, leading to approximation errors. The introduction of windows can lead to slightly more complex frequency domain techniques, replacing diagonal matrices by banded matrices, but with controlled approximation error. Other work observed this recently, proposing general banded matrices in the frequency domain for filtering. Here, we emphasize the design of a window to optimize the banded approximation, and more importantly, we show that the whole banded matrix is in fact still parametrized by a diagonal matrix, which facilitates estimation. We propose here both some non-parametric and parametric approaches for estimating the diagonal spectral parts and revisit in particular the effect of the window on frequency domain Recursive Least-Squares (RLS) adaptive filtering.

***Index Terms*—** frequency domain filtering, DFT, FFT, window, periodogram, recursive least-squares, adaptive filtering.

1. INTRODUCTION

Frequency domain processing is a classical topic in Wiener filtering, especially for the case of long filters. The main motivation is the reduction of computational complexity since convolution becomes a simple product in frequency domain and passing between time domain and frequency domain can be done with limited complexity via the FFT. However, the Fourier transform is defined over infinite time and needs to be approximated by the DFT in practice. There exists of course approaches to make this transition exact, such as overlap-add/overlap-save. However, their use in adaptive filtering requires some computational complexity to enforce an assumed finite length adaptive filter [1], [2]. A somewhat approximate approach is to work with filterbanks, which attempts to isolate the subbands better in the frequency domain. However, subband signals remain correlated, which leads to the introduction of crossband filters [3] which can be further subsampled if the spectral overlap between subbands is reduced. The work we present here is in fact very much related to these filterbank approaches.

In audio processing, it is very customary to work in the DFT domain, replacing linear convolution by circulant convolution, leading to approximation errors. In [4], [5], it was proposed to replace diagonal system representations in the frequency domain by banded matrices. There were no particular considerations for reducing the size of the band or the quality of the approximation. Also, no particular structure was assumed for these banded matrices. In this paper we emphasize the design of the window when applying the DFT to windowed data. We then point out that the banded matrices in frequency domain are in fact parameterized by diagonal matrices, which get smeared out into banded matrices by the Fourier transform of the window, which is known. In fact, what is going on here is a form of subband filtering, with a so-called modulated filterbank, where the window represents a lowpass filter and the subband filters are obtained by modulating the window to the various subband center frequencies, those of the DFT. The subsampling of the subband signals is non-trivial and depends on the overlap between de windows of consecutive frames in the time domain.

In this paper we explore first non-parametric approaches for adapting a Wiener filter in the resulting banded correlation matrices in the DFT domain. We then consider in detail frequency domain adaptive filtering using least-squares cost functions.

EURECOM's research is partially supported by its industrial members: ORANGE, BMW Group, Swisscom, SFR, ST Microelectronics, Symantec, SAP, Monaco Telecom, iABG, and also by the French ANR project DIONISOS and the EU FP7 NoE NEWCOM#.

2. WINDOWING FOR FRAME-BASED PROCESSING

The audio signals considered are by nature non-stationary. If we can consider the parameters constant during a short time, we can process the signal in frames (time segments), over which the signal can be considered stationary, which corresponds to time-invariant filtering. Many of the signal processing operations (e.g. linear time-invariant filtering and filter computation) could be largely simplified by passing to the frequency domain. However, transforming a frame of signal to the frequency domain directly via the DFT (FFT) leads to approximations due to the periodic extension of the frame assumption inherent in the DFT. We shall see later how we can improve these approximations. Just like the original data signal y_k will be cut into a series of windowed frames of length N, a bit like in the Welch method, a processed signal (e.g. extracted source) will be reconstructed by superposing its reconstructed windowed frame segments. Since the window needs to decay towards its edges, consecutive frames need to overlap. Let M be the hop size (time jump) from one frame to the next, then a perfect reconstruction (PR) window w_n requires

$$\sum_{i=-\infty}^{\infty} w_{n-iM} \;=\; 1\;,\;\; \forall n \qquad (1)$$

see the top figures in Fig. 1 for the cases of relative overlap of $(N{-}M)/N = 50\%, 75\%$ (both the individual windows and their sum are shown for a finite set of windows). Note that one could consider extensions to non-PR windows, in which the superposition of windowed signal frames could be followed by a zero-forcing rescaling with $1/(\sum_{i=-\infty}^{\infty} w_{t-iM})$ or (multi-window) MMSE versions thereof. An example of a PR window is a Hann (or raised

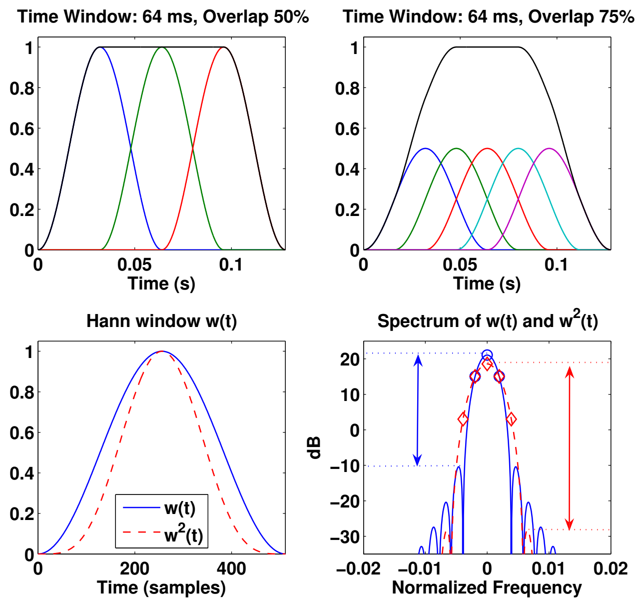

Fig. 1. Perfect reconstruction windowing.

cosine) window

$$w_t \;=\; \frac{1}{2}\left[1 - cos\left(2\pi\frac{t}{N}\right)\right]\;,\quad t = 0, 1, \ldots, N-1\,. \tag{2}$$

The continuity of the window at its edges can be expected to be reflected in the continuity of the reconstructed signal and help reduce blocking artifacts (musical noise). The motivations for the window design will be different however in the parameter estimation part as we shall see. In a separate approach for parameter estimation and source extraction, as considered here, different windows could be used in both parts.

3. WINDOWED DFT DOMAIN PROCESSING

Working in the time domain, we have a full covariance $\mathbf{R}$ to work with. By going to the frequency domain, one typically assumes to be able to work with a diagonal $\mathbf{R}$ because asymptotically, different frequency components are uncorrelated. We shall analyze more precisely the nonasymptotic regime. For the rest of this section, consider parametric spectrum estimation of a signal y_k, e.g. using Gaussian Maximum Likelihood (GML). Now, let the current frame of N samples be $\mathbf{y} = [y_0\, y_1 \cdots y_{N-1}]^T$ and w.l.o.g. we assumed that the first sample starts at time zero. Before applying the DFT, the data get windowed. Let $\mathbf{W} = \operatorname{diag}\{w_0, w_1, \ldots, w_{N-1}\}$ and $\mathbf{F}$ is the $N \times N$ discrete Fourier transform (DFT) matrix, with inverse DFT $\frac{1}{N}\mathbf{F}^* = \frac{1}{N}\mathbf{F}^H$. Then we shall work with the transformed windowed data vector

$$Y = \mathbf{F}\,\mathbf{W}\,\mathbf{y}\,. \tag{3}$$

The data are assumed to have zero mean so that covariance and correlation matrices are equal. Note now that $\mathbf{y}$ is real, but Y is complex due to the DFT. Y is strictly speaking non-circular as both $\mathbf{R} = \operatorname{E} YY^H$ and $\operatorname{E} YY^T$ are nonzero. However, Y is not a genuine complex random vector as only the real vector $\mathbf{y}$ is random and the complex aspect is due to a deterministic transformation. As a result we can continue as if Y has a circular complex Gaussian distribution (which corresponds to a real Gaussian distribution with transposes replaced by Hermitian transposes). Now, all we need for GML is $\mathbf{R}$. Note that component Y_k of $Y = [Y_0\, Y_1 \cdots Y_{N-1}]^T$ is in fact the discrete-time Fourier transform $\mathcal{F}$ (DTFT) $Y^w(f)$ of the windowed signal evaluated at frequency $f = k/N$. To constitute $\mathbf{R}$, we shall need the correlations between different frequencies $\operatorname{E} Y^w(f_1)\, Y^{w*}(f_2)$. For this consider

$$\begin{array}{l} Y^w(f_1) = \displaystyle\sum_{n=0}^{N-1} w_n\, y_n e^{-j2\pi f_1 n} = \sum_{n=-\infty}^{\infty} w_n e^{-j2\pi f_1 n}\, y_n \\ = \displaystyle\sum_{n=-\infty}^{\infty} h_{-n}\, y_n = h_n * y_n|_{n=0} \\ = \int_{-\frac{1}{2}}^{\frac{1}{2}} H(f)\, Y(f)\, df = \int W(f_1 - f)\, Y(f)\, df \end{array} \tag{4}$$

where we zeropadded the finite window to infinity. Now we get

$$\begin{array}{l} \operatorname{E} Y^w(f_1)\, Y^{w*}(f_2) \\ = \operatorname{E} \int W(f_1 - f)\, Y(f)\, df \int W^*(f_2 - f_0)\, Y^*(f_0)\, df_0 \\ = \int df\, W(f_1 - f) \int df_0 W^*(f_2 - f_0)\, \operatorname{E} Y(f) Y^*(f_0) \\ = \int df\, W(f_1 - f) \int df_0 W^*(f_2 - f_0)\, S_{yy}(f)\, \delta_1(f - f_0) \\ = \int df\, W(f_1 - f)\, W^*(f_2 - f)\, S_{yy}(f) \end{array} \tag{5}$$

where $Y(f) = \sum_{k=-\infty}^{\infty} y_k e^{-j2\pi f k}$ is the DTFT of the stationary random process y_n with spectrum $S_{yy}(f)$, $W(f)$ is the DTFT of the window w_n, and $\delta_1(f) = \sum_{k=-\infty}^{\infty} \delta(f - k)$ is the periodicized delta function. Now let us introduce the vector of DFT frequencies $\underline{f} = [0\, 1 \cdots N{-}1]^T/N$ and the $N \times 1$ vector of ones $\underline{1}$, let $W(\underline{f})$ denote the column vector of $W(.)$ evaluated at the components of $\underline{f}$, then we can write for

$$\mathbf{R} = \int df\, W(\underline{f} - f\underline{1})\, W^H(\underline{f} - f\underline{1})\, S_{yy}(f)\,. \tag{6}$$

We get in particular for the diagonal elements $\mathbf{R}_{kk} = \int df\, |W(\frac{k-1}{N} - f)|^2\, S_{yy}(f)$ which is the well-known spectrum smearing appearing in the mean of the periodogram. Now, to limit complexity in the frequency domain based methods, one should sparsify $\mathbf{R}$ as much as possible. Here is where the window design comes in. For a properly designed window, $W(f)$ can be neglected outside of its main lobe (see e.g. the lower right corner in Fig. 1). Note that from this point of view, a rectangular window is (again) not a very good choice since the sidelobes are not much attenuated. If Δf is the doublesided width of the main lobe of $W(f)$, then $\int df\, W(f_1 - f)\, W^*(f_2 - f)\, S_{yy}(f)$ can be approximated to zero for $|f_1 - f_2| > \Delta f$. This means that $\mathbf{R}$ can be approximated by a banded matrix with only $\lceil N\, \Delta f \rceil$ non-zero diagonals. E.g. the inversion of $\mathbf{R}$ can then be done efficiently using the LDU triangular factorization of $\mathbf{R}$ in which the triangular factors will also be banded. Compared to classical frequency-domain asymptotics, the spectrum gets smeared on the diagonal and spills onto the main sub- and super-diagonals, leading to correlations between neighboring frequencies (only). In those classical asymptotics, the smearing effect of $W(f)$ gets neglected, leading to $\mathbf{R} = \operatorname{diag}\{S_{yy}(\underline{f})\}$.

If $S_{yy}(f)$ is sufficiently smooth, the integral in (6) can be approximated by a sum over frequencies spaced more densely at $\underline{f}'$, containing multiples of $1/N'$, where $N' > N$. This can be obtained by zeropadding the signal from N to N' samples and applying the DFT of size N'. We then get $\mathbf{R}'$ of the form

$$\mathbf{R}' = \mathcal{C}(W(\underline{f}'))\, \operatorname{diag}\{S_{yy}(\underline{f}')\}\, \mathcal{C}^H(W(\underline{f}')) \tag{7}$$

where $\mathcal{C}$ denotes a circulant matrix constructed from the vector argument. The entries of $\mathbf{R}'$ can be downsampled to obtain $\mathbf{R}$ if desired.

4. FREQUENCY DOMAIN CRAMER-RAO BOUNDS (CRBS)

For a Gaussian process with zero mean, the element (i,j) (pertaining to θ_i and θ_j) of the Fisher Information Matrix (FIM) are obtained as

$$FIM_{i,j} = \text{tr}\{\mathbf{R}^{-1}\frac{\partial \mathbf{R}}{\partial \theta_i}\mathbf{R}^{-1}\frac{\partial \mathbf{R}}{\partial \theta_j}\}\ . \tag{8}$$

Here, $\mathbf{R}$ is given in (6) and we get for the derivatives

$$\frac{\partial \mathbf{R}}{\partial \theta_i} = \int df\, W(\underline{f}-f\underline{1})\, W^H(\underline{f}-f\underline{1})\,\frac{\partial S_{yy}(f)}{\partial \theta_i}\ . \tag{9}$$

In the classical asymptotics, the FIM gets then approximated as

$$\begin{aligned} FIM_{i,j} &= \int df\, S_{yy}^{-2}\frac{\partial S_{yy}(f)}{\partial \theta_i}\frac{\partial S_{yy}(f)}{\partial \theta_j} \\ &= \int df\, \frac{\partial \ln S_{yy}(f)}{\partial \theta_i}\frac{\partial \ln S_{yy}(f)}{\partial \theta_j}\ . \end{aligned} \tag{10}$$

5. AR MODELING OF SPEECH MIXTURES

The application of the proposed windowed frequency domain processing approach to (short-term plus long-term) Autoregressive (AR) modeling of speech signals in a mixture is considered in [6]. This is done by considering the equivalence of Gaussian Maximum Likelihood, Optimally Weighted Covariance Matching and Itakura-Saito distance minimization. The model of periodic signals with (short-term) AR spectral envelopes is also considered there.

6. NON-PARAMETRIC DFT DOMAIN ADAPTIVE FILTERING

Wiener Filtering in the DFT domain leads to

$$\hat{X} = \mathbf{H}Y\ ,\ \ \mathbf{H} = \mathbf{R}_{XY}\mathbf{R}_{YY}^{-1} \tag{11}$$

where both $\mathbf{R}_{XY}$, $\mathbf{R}_{YY}$ are of the form in (7), from which one can see that one needs to estimate the spectra $S_{xy}(f)$ and $S_{yy}(f)$. Various approaches are possible of course, such as the periodogram and its variants averaged and smoothed periodogram. These estimates obviously need to be averaged in time also, e.g. with an exponential forgetting factor, at the rate at which the overlapping windows appear, or at a higher rate with a sliding DFT.

Whereas both $\mathbf{R}_{XY}$, $\mathbf{R}_{YY}$ could be estimated in this way, a hybrid approach would be to estimate only $\mathbf{R}_{YY}$ in this way, and to estimate S_{xy} in $\mathbf{R}_{XY}$ with a classical adaptive filtering approach such as the LMS or RLS algorithms.

As far as carrying out the inverse for $\mathbf{R}_{YY}^{-1}$ is concerned, note that a LDU triangular factorization of a banded matrix leads to banded triangular factors. Multiplying a vector with the inverse of a banded triangular factor has the same complexity as mutliplying with the same factor.

7. FREQUENCY DOMAIN WINDOWED SUBSAMPLED UPDATING (FWSU) RLS

We shall here work out the details for the windowed frequency domain approach applied to Subsampled Updating Recursive Least-Squares (SU RLS) adaptive filtering. Subsampled updating corresponds to updating once per frame instead of updating sample by sample. SU RLS was originally considered in [7] where a fast version was developed by exploiting the displacement structure of the Kalman gain and various other matrices that appear in SU RLS. What we propose here is to introduce (non-rectangular) windows in the SU RLS formulation, which will allow different fast algorithms, via the banded frequency domain approximation. This approach will also provide a link to adaptive filtering in subbands via filter-banks [3]. In related work, a frequency domain approach for RLS was developed in [8], with the goal to develop the RLS version of various frequency domain LMS algorithms, such as the (extended) multidelay adaptive filter. However, [8] introduces a very rough approximation (replacing a singular banded matrix by a multiple of identity) compared to the much more precise approximation introduced here. It is true that for adaptive filtering purposes this approximation is less dramatic since it leads to a Gauss-Newton adaptive filter with an approximated Hessian compared to RLS.

In what follows, we shall mostly follow the notation of [7] and we often assume for simplicity that the window length$L$ is a power of two and that the length N of the transversal filter coefficients $H_{N,k} = \left[h_{N,k}^1 \cdots h_{N,k}^N\right]^H$ is such that $M = N/L$ is an integer, though more general cases can be considered equally well. We shall introduce the following notation. Let $\lambda \in (0,1]$ be the exponential weighting factor, $\Lambda = \text{diag}\{w(L-1),\ldots,w(1),w(0)\}$ contains the window, $\|v\|_\Lambda^2 = v\Lambda v^H$, $\|.\| = \|.\|_I$. The desired response signal is $d(k)$ and the filter input is $x(k)$. Let

$$\begin{aligned} d_{L,k} &= \begin{bmatrix} d^H(k{-}L{+}1) \\ \vdots \\ d^H(k) \end{bmatrix},\ x_{L,k} = \begin{bmatrix} x^H(k{-}L{+}1) \\ \vdots \\ x^H(k) \end{bmatrix}, \\ X_{N,L,k} &= \begin{bmatrix} X_N^H(k{-}L{+}1) \\ \vdots \\ X_N^H(k) \end{bmatrix} = [x_{L,k} \cdots x_{L,k-N+1}] \end{aligned} \tag{12}$$

and $X_N(i) = \left[x^H(i)\, x^H(i{-}1) \cdots x^H(i{-}N{+}1)\right]^H$. The WSU RLS algorithm minimizes the cost function

$$\xi_k = \lambda \xi_{k-L} + \|d_{L,k} - X_{N,L,k}H_{N,k}\|_\Lambda^2 \tag{13}$$

where on could replace λ by λ^L and modulate the window Λ with the exponential weighting also in order to get an exact correspondence with exponentially weighted RLS if desired. This cost function (13) gets minimized for the filter $H_{N,k}$ satisfying the normal equations $R_{N,k}\, H_{N,k} = P_{N,k}$ where

$$\begin{aligned} P_{N,k} &= \lambda P_{N,k-L} + X_{N,L,k}^H \Lambda d_{L,k}\ , \\ R_{N,k} &= \lambda R_{N,k-L} + X_{N,L,k}^H \Lambda X_{N,L,k}\ . \end{aligned} \tag{14}$$

Continuing with e.g. $P_{N,k}$, consider a partitioning in $M = N/L$ subvectors of length L:

$$P_{N,k} = \left[P_{N,k}^{1\,H} \cdots P_{N,k}^{M\,H}\right]^H \tag{15}$$

then (14) reduces for subvector j to

$$P_{N,k}^j = \lambda P_{N,k-L}^j + X_{L,L,k-(j-1)L}^H \Lambda d_{L,k}\ ,\ \ j = 1,\ldots,M\ . \tag{16}$$

In other words, we have essentially $2M$ times $2L$ multiplications (for e.g. $\lambda P_{N,k-L}^j$ and $\Lambda d_{L,k}$) plus $2M$ times the product of a $L \times L$ Toeplitz matrix with a vector of length L. Such a product can be efficiently computed in basically two different ways. One way is to use the overlap-save method. We can embed the $L \times L$ Toeplitz matrix $X_{L,L,k}$ into a $2L \times 2L$ circulant matrix, viz.

$$\overline{X}_{L,L,k}^H = \begin{bmatrix} * & X_{L,L,k}^H \\ X_{L,L,k}^H & * \end{bmatrix} = \mathcal{C}\left(x_{2L,k}^H\right) \tag{17}$$

where $\mathcal{C}(c^H)$ is a right shift circulant matrix with c^H as first row. Then we get for the matrix-vector product

$$X^H_{L,L,k-(j-1)L}\Lambda d_{L,k} = [I_L\ 0_{L\times L}]\ \mathcal{C}\Big(x^H_{2L,k-(j-1)L}\Big)\begin{bmatrix}0_{L\times 1}\\ \Lambda d_{L,k}\end{bmatrix}. \tag{18}$$

The product of a circulant matrix $\mathcal{C}(c^H)$ with a vector v where c and v are of length m can be computed efficiently as follows. Let F_m be the Discrete Fourier Transform matrix for a DFT of length m. Then using the property that a circulant matrix can be diagonalized via a similarity transformation with a DFT matrix, we get

$$\mathcal{C}(c^H)\,v\ =\ \frac{1}{m}\,\mathcal{C}(c^H)\,F_m^H F_m v\ =\ \frac{1}{m}\,F_m^H\ \mathrm{diag}^H\,(F_m\,c)\ F_m v \tag{19}$$

where $\mathrm{diag}(w)$ is a diagonal matrix with the elements of the vector w as diagonal elements. So the computation of the vector in (18) requires L multiplications to form the product $\frac{1}{2L}\Lambda d_{L,k}$, the padding of the resulting vector with L zeros, the DFT of the resulting vector, the DFT of $x_{2L,k-(j-1)L}$, the product of the two DFTs, and the (scaled) IDFT of this product. Note that at time k, only the FFT of $x_{2L,k}$ needs to be computed; the FFTs of $x_{2L,k-jL}, j=1,\ldots,M-1$ have been computed at previous time instants. The above procedure reduces the $2N(1+\frac{1}{L})$ computations per sample for $P_{N,k}$ and $r_{N,k}$ (the first column of $R_{N,k}$) to

$$2N\left[\frac{2\,\mathrm{FFT}(2L)}{L^2}+\frac{4}{L}\right]+\frac{\mathrm{FFT}(2L)}{L} \tag{20}$$

computations per sample (FFT(L) signifies the computational complexity associated with a FFT of length L) or basically $\mathcal{O}\left(N\ \frac{\log_2(L)}{L}\right)$ operations.

7.1. Fast Computation of the Filtering Errors using the FFT

In a block processing approach, also the filtering errors can be computed more efficiently than by computing an inner product every sample period. Indeed, consider the following vector of (block) a priori filtering errors

$$\epsilon^p_{N,L,k}\ =\ \begin{bmatrix}\epsilon^H_N(k{-}L{+}1|k{-}L)\\ \vdots\\ \epsilon^H_N(k|k{-}L)\end{bmatrix}\ =\ d_{L,k}-X_{N,L,k}H_{N,k-L} \tag{21}$$

with $\epsilon_N(i|k)=d(i)-H^H_{N,k}\,X_N(i)$. We can use the same circular matrix embedding and DFT techniques as in the previous subsection. Remark that the input data have been transformed before (in the previous subsection) and that we only need to apply the inverse DFT once after having summed up the M products in the frequency domain. This leads to a computational complexity of

$$N\left[\frac{\mathrm{FFT}(2L)}{L^2}+\frac{2}{L}\right]+\frac{\mathrm{FFT}(2L)}{L} \tag{22}$$

per sample instead of N.

7.2. The SU RLS Algorithm

Instead of computing the filter $W_{N,k}$ from scratch every L samples by solving the normal equations, we can exploit the recursions for $P_{N,k}$ and $R_{N,k}$ to get a recursive solution

$$\begin{aligned}\widetilde{C}_{N,k} &= \lambda^{-1}R^{-1}_{N,k-L}\,X^H_{N,L,k} && (23)\\ \gamma^{-1}_N(k) &= \Lambda^{-1}+X_{N,L,k}\widetilde{C}_{N,k} && (24)\\ R^{-1}_{N,k} &= \lambda^{-1}R^{-1}_{N,k-L}\ -\ \widetilde{C}_{N,k}\gamma_N(k)\widetilde{C}^H_{N,k} && (25)\end{aligned}$$

$$\begin{aligned}\epsilon^p_{N,L,k} &= d_{L,k}-X_{N,L,k}H_{N,k-L} && (26)\\ \gamma^{-1}_N(k)\,\epsilon_{N,L,k} &= \epsilon^p_{N,L,k} && (27)\\ H_{N,k} &= H_{N,k-L}+\widetilde{C}_{N,k}\,\epsilon_{N,L,k} && (28)\end{aligned}$$

where $\epsilon_{N,L,k}$ is a vector of a posteriori errors:

$$\epsilon_{N,L,k}\ =\ \begin{bmatrix}\epsilon^H_N(k{-}L{+}1|k)\\ \vdots\\ \epsilon^H_N(k|k)\end{bmatrix}. \tag{29}$$

While the Subsampled-Updating RLS algorithm thus obtained constitutes a valid algorithm to provide the filter solution $W_{N,k}$ every L samples, it does not represent much computational gain w.r.t. the original RLS algorithm ($L=1$). We could exploit the FFT technique introduced above to reduce the computational complexity in equations (23),(24) and (26) by a factor $\mathcal{O}\left(\frac{L}{\log_2 L}\right)$. On the other hand, we have to invert $\gamma^{-1}_N(k)$, a $L\times L$ matrix. Below, we shall introduce a fast version of the SU RLS algorithm.

7.3. The FWSU RLS Algorithm

We shall transform the following update equations

$$\begin{aligned}R_{N,k} &= \lambda R_{N,k-L}+X^H_{N,L,k}\Lambda X_{N,L,k} && (30)\\ C_{N,k} &= R^{-1}_{N,k}X^H_{N,L,k} && (31)\end{aligned}$$

$$\begin{aligned}\epsilon^p_{N,L,k} &= d_{L,k}-X_{N,L,k}H_{N,k-L} && (32)\\ H_{N,k} &= H_{N,k-L}+C_{N,k}\,\Lambda\,\epsilon^p_{N,L,k} && (33)\end{aligned}$$

into the frequency domain. Introducing circulant and related embeddings:

$$\begin{aligned}&\overline{\Lambda}=\begin{bmatrix}0&0\\0&\Lambda\end{bmatrix},\ \overline{W}=\begin{bmatrix}W&0\\0&0\end{bmatrix},\ W=I_L\,,\\ &\overline{W}_M=I_M\otimes\overline{W}\ ,\ \overline{H}_{N,k}=\left(I_M\otimes\begin{bmatrix}I_L\\0\end{bmatrix}\right)H_{N,k}\,,\\ &\overline{X}_{N,2L,k}=\left[\overline{X}_{L,L,k}\,\overline{X}_{L,L,k-L}\cdots\overline{X}_{L,L,k-(M-1)L}\right].\end{aligned} \tag{34}$$

Note that $\overline{W}\,\overline{H}_{N,k}=\overline{H}_{N,k}$. Then one can derive similarly as in [8]

$$\begin{aligned}\overline{R}_{N,k} &= \lambda\overline{R}_{N,k-L}+\overline{X}^H_{N,2L,k}\overline{\Lambda}\,\overline{X}_{N,2L,k} && (35)\\ \overline{C}_{N,k} &= \overline{R}^{-1}_{N,k}\overline{X}^H_{N,2L,k} && (36)\end{aligned}$$

$$\begin{aligned}\overline{\epsilon}^p_{N,2L,k} &= d_{2L,k}-\overline{X}_{N,2L,k}\overline{H}_{N,k-L} && (37)\\ \overline{H}_{N,k} &= \overline{H}_{N,k-L}+\overline{W}_M\,\overline{C}_{N,k}\,\overline{\Lambda}\,\overline{\epsilon}^p_{N,2L,k}\,. && (38)\end{aligned}$$

Now we pass to the frequency domain by taking DFTs of size $2L$. Let

$$\begin{aligned}&\underline{\Lambda}=\mathbf{F}_{2L}\overline{\Lambda}\mathbf{F}^{-1}_{2L},\underline{W}=\mathbf{F}_{2L}\overline{W}\mathbf{F}^{-1}_{2L}\ ,\ \underline{W}_M=I_M\otimes\underline{W}\\ &\underline{H}_{N,k}=(I_M\otimes\mathbf{F}_{2L})\ \overline{H}_{N,k}\ ,\ \underline{d}_{2L,k}=\mathbf{F}_{2L}d_{2L,k}\,,\\ &\underline{X}_{2L,k}=\mathcal{C}^H\left(x^H_{2L,k}\right)\,,\\ &\underline{X}_{N,2L,k}=\left[\underline{X}_{2L,k}\,\underline{X}_{2L,k-L}\cdots\underline{X}_{2L,k-(M-1)L}\right].\end{aligned} \tag{39}$$

Then we get the FWSU RLS algorithm

$$\underline{R}_{N,k} = \lambda \underline{R}_{N,k-L} + \underline{X}^H_{N,2L,k} \underline{\Lambda}\, \underline{X}_{N,2L,k} \quad (40)$$
$$\underline{C}_{N,k} = \underline{R}^{-1}_{N,k} \underline{X}^H_{N,2L,k} \quad (41)$$
$$\underline{\epsilon}^p_{N,2L,k} = \underline{d}_{2L,k} - \underline{X}_{N,2L,k} \underline{H}_{N,k-L} \quad (42)$$
$$\underline{H}_{N,k} = \underline{H}_{N,k-L} + \underline{W}\, \underline{C}_{N,k}\, \underline{\Lambda}\, \underline{\epsilon}^p_{N,2L,k} \quad (43)$$

Note that $\underline{H}_{N,k}$ contains now the filter coefficients in the frequency domain. Let $\mathcal{P}$ be a permutation matrix so that the M blocks of $2L$ coefficients in $\underline{H}_{N,k}$ get transformed to $2L$ blocks of M coefficients in $\mathcal{P}\underline{H}_{N,k}$. Due to overlap-save, the frequency response is oversampled in frequency domain, containing $2L$ subbands at each of which sits an FIR filter of M coefficients, combining input signal from M consecutive frames (the subsampling factor in each subband is L). $\mathcal{P}\underline{X}^H_{N,2L,k}$ is a $2L \times 2L$ block diagonal matrix with blocks of size $M \times 1$, containing M consecutive input samples at each respective subband. In [8], where a rectangular window Λ was used, the matrix $\underline{\Lambda}$ was approximated to be $\underline{\Lambda} \approx \frac{1}{2} I_{2L}$. However, the rank of $\underline{\Lambda}$ is only L, half its size. $\underline{\Lambda}$ is a circulant matrix, containing the zero padded DFT of the window in Λ. With a proper window design mentioned earlier, $\underline{\Lambda}$ can be approximated by a banded matrix if the DFT of the window can be approximated by its main lobe. In this case we have a modulated filterbank with the window as lowpass filter. Note that due to the circulant nature, $\underline{\Lambda}$ is in principle a circulant banded matrix. However, the triangles in the lower-left and the upper-right corners can be neglected if the high frequencies (near half the sampling frequency) are negligible in the input signal (by e.g. the use of an anti-aliasing filter with a slightly reduced cutoff frequency). If $\underline{\Lambda}$ is banded, then $\underline{R}$ is a banded block matrix with $M \times M$ blocks. If $\underline{\Lambda}$ has a main diagonal and $B-1$ upper and lower diagonals, then $\underline{R}$ has equally many block diagonals and its LDU triangular factorization takes a complexity of $\mathcal{O}(N\,B^2\,M^2)$ (as opposed to $\mathcal{O}(N^3)$ for a full matrix). The computation of the Kalman gain in (41) is of similar complexity. The singularity of $\underline{R}$ will show up in a number of the diagonal entries in the diagonal factor being zero. For the inversion then of $\underline{R}$ a pseudo-inverse can be used.

Note that the computation of the $2L$ error signals in the $2L$ subbands occurs in a decoupled fashion in (42). The computation of the filter outputs $\underline{X}_{N,2L,k}\underline{H}_{N,k-L}$ however can be interpreted as $\underline{X}_{N,2L,k}\underline{W}\underline{H}_{N,k-L}$ where $\underline{W}$ is again a circulant matrix, mixing contributions from filters in neighboring subbands. If $\underline{W}$ can be approximated by a banded matrix, then this corresponds to the use of cross-band filters as has been proposed for adaptive filtering in the filterbank domain [3]. However, here the cross-band filters are parameterized in terms of the inband filters. Even though the error signals are computed per subband, the update of the filter coefficients in (42) is coupled between neighboring subbands due do the Kalman gain $\underline{C}$ not being block diagonal and the presence of $\underline{\Lambda}$ again.

The use of non-trivial windows for W also could be explored further.

REFERENCES

[1] E. Moulines, O. Ait-Amrane, and Y. Grenier, "The Generalized Multi-Delay Filter: Structure and Convergence Analysis," *IEEE Trans. Signal Proc.*, Jan. 1995.

[2] K. Maouche and D. Slock, "Fast Subsampled-Updating Stabilized Fast Transversal Filter (FSU SFTF) RLS Algorithm for Adaptive Filtering," *IEEE Trans. Signal Proc.*, Aug. 2000.

[3] A. Gilloire and M. Vetterli, "Adaptive Filtering in Subbands," in *IEEE Int'l Conf. Acoustics, Speech and Signal Proc. (ICASSP)*, New York, USA, 1988.

[4] Y. Avargel and I. Cohen, "System Identification in the Short-Time Fourier Transform Domain With Crossband Filtering," *IEEE Trans. Audio, Speech, and Language Proc.*, May 2007.

[5] Y. Avargel and I. Cohen, "Adaptive System Identification in the Short-Time Fourier Transform Domain Using Cross-Multiplicative Transfer Function Approximation," *IEEE Trans. Audio, Speech, and Language Proc.*, Jan. 2008.

[6] S. Bensaid and D. Slock, "Blind Audio Source Separation Exploiting Periodicity and Spectral Envelopes," in *IEEE Int'l Workshop on Acoustic Signal Enhancement (IWAENC)*, Aachen, Germany, 2012.

[7] K. Maouche and D. Slock, "The Fast Subsampled-Updating Recursive Least-Squares (FSU RLS) Algorithm for Adaptive Filtering based on Displacement Structure and the FFT," *Eurasip Signal Processing*, 1994.

[8] H. Buchner, J. Benesty, and W. Kellermann, "Generalized Multichannel Frequency-Domain Adaptive Filtering: Efficient Realization and Application to Hands-free Speech Communication," *Eurasip Signal Processing*, 2005.