16th European Signal Processing Conference (EUSIPCO 2008), Lausanne, Switzerland, August 25-29, 2008, copyright by EURASIP

WOLA NOISE CANCELLING PERFORMANCE

Guido Schuster and Reto Ansorge

Electrical Engineering Department, University of Applied Sciences of Eastern Switzerland in Rapperswil, gschuste@hsr.ch

ABSTRACT

The goal of noise cancelling for hearing aids is to remove the noise in the corrupted speech so that the hearing impaired can better understand and participate in a conversation. Most digital hearing aids use a Weighted Overlap-Add (WOLA) structure [1] to implement the noise cancelling algorithm. Often, it is attempted to build a short time Wiener filter with a WOLA either in the FFT or in the Bark domain. Such a filter results in real valued gains to the FFT bins of the input signal and can be easily implemented in the WOLA structure. These real valued gains are also symmetrical with respect to the Nyquist rate, which results in a real valued impulse response from the filter.

In this paper we answer the question of how an optimal minimum mean square error (MSE) filtering should be done with a WOLA structure, given the constraint of real valued gains and a real valued filter impulse response. Since the gains obtained with a Wiener approach are between zero and one, we also find the optimal gains given these additional constraints using quadratic programming.

We assume that we have access to all the spectral properties of the speech and the noise signal we need, so that we can find the optimal gains. By assuming that this spectral estimation is ideal, we decouple the estimation problem and the MSE filtering problem with a WOLA structure, and the results show the performance that would be achievable using this popular signal processing structure.

The problem of finding the optimal gains is structured in such a way that we first find the optimal solution for each individual WOLA frame. Since in the WOLA structure, the contribution of a given frame to the final output is weighted by a window function, we next find a weighted optimal solution, which results in a better overall solution. We then introduce an iterative scheme which attempts to find an overall optimal solution by considering the contributions of each frame and improving the frames iteratively. We show that this scheme must converge to a local optima, which can be significantly better than the per frame solutions.

1. INTRODUCTION

In most digital hearing aids, the noise cancelling is achieved using a WOLA structure and a type of Wiener filter. Since neither the power spectral densities of the signal nor the noise are usually known, they need to be estimated. After the estimation, the Wiener filter formula can be used to calculate the gains in either the FFT or the Bark domain, and the gains are then applied in the FFT domain and the resulting signal is transformed back into the time domain. The structure of the Wiener filter formula results in gains for each FFT bin which are real, symmetrical with respect to the Nyquist rate, and between zero and one.

Every practical noise cancelling system can be thought of as an estimator/filter combination, where the estimation process is used to estimate the spectral properties of the noise and speech, and the filter is used to remove the noise from the corrupted speech. Since such noise cancelling systems are almost always tested as whole systems, it is not clear what part of the system, the estimator or the filter or even a particular combination of those, is responsible for the observed performance.

The aim of this paper is to study the filter alone. That is, we assume that the estimator results in perfect estimations of the information we need for the filter. The aim is to find the performance limit of the frequently employed WOLA structure, using real valued gains that are symmetrical with respect to the Nyquist rate, i.e., that would result in a real valued impulse response. Furthermore we also add the constraint that results from the Wiener formula, that the gains have to be between zero and one, and observe the resulting performance loss because of this constraint.

Since in the WOLA structure, the frames have a strong overlap, treating the frames independently will not result in an overall optimal solution. Furthermore, this dependency results in an optimization problem for which a globally optimal solution might be very hard to find. Hence, using the optimal solution formulated for a single frame, we introduce an iterative scheme that is guaranteed to converge to a local optima for the overall system.

The paper is organized as follows. In section 2 we introduce the required notation. In section 3 the optimization problem for a single WOLA frame is defined and a solution in the time and the FFT domain is derived. In section 4, it is observed that each frame of a WOLA has a weighted contribution to the final signal and hence minimizing the MSE might not be the best approach, since the parts in the middle of the frame are much more heavily weighted than the edges of the frame. Hence a weighted MSE solution is presented. In section 5 the constraints imposed by the Wiener formula are introduced to the optimization problem. Hence the gains are only allowed to be between zero and one. This additional constraint set is dealt with using a quadratic programming approach. In section 6, the per frame optimal solution is extended to an iterative scheme for the entire sequence which is guaranteed to converge to a local optima. In section 7, experiments with the different schemes are presented and compared to a Wiener filter. Finally, the paper is summarized and conclusions are drawn in section 8.

2. MATHEMATICAL NOTATION

The idea behind a WOLA is that the input signal is cut into frames of a given length and each frame is weighted by a window function. These windowed frames are then transformed with an FFT into the frequency domain, where they can be manipulated. Normally, successive frames have a strong overlap, resulting in a data expansion in this process. After the frames have been manipulated in the frequency domain, they need to be transformed back into the time domain. Since each frame is contributing an overlapping part to the final result, the frames need to be added. It is interesting to note that it is possible that, through the manipulation in the frequency domain, there is no corresponding time sequence anymore. In other words, there is no time sequence which would result in the manipulated frequency information. Hence the goal is to find that time sequence which, if transformed by a WOLA, would result in a frequency representation that would have the smallest mean squared error compared with the manipulated frequency information. This is achieved by adding the frames, weighted with the original window, and normalizing by a term that depends on the window used. In this paper a raised cosine window (also called a Hann window) having 25% overlap is used and the term is in this case a constant, 1.5.

The manipulation in the frequency domain has the goal of reducing the noise in the composite signal. Simply speaking, this is achieved by suppressing the bins that contain a lot of noise. The factors that are used to lower the bins are called the gains or weights and these are real values. Since the original signal is a real signal

and the filtered signal should also be real, the gains are symmetrical with respect to half the sampling frequency (Nyquist rate).

Hence, given one wants to stay within this established framework of noise cancelling in a WOLA structure, these gains, which have to be real and symmetrical, are the free parameters that we want to estimate, such that the resulting mean squared error of the filtered signal compared with the original speech signal is as small as possible.

We now introduce the notation necessary to formulate the problem mathematically. Let $s(n)$ be the sampled speech signal and $n(n)$ the sampled noise signal at the discrete time n. Let $x(n) = s(n) + n(n)$ be the composite signal, which is the noisy speech signal. Let $w(n)$ be the real valued weights of the filter and $y(n) = x(n) \otimes w(n)$ be the output of the filter, which is the circular convolution of the input signal (noisy speech) with the impulse response of the filter. The convolution is circular since it is the result of a multiplication of the gains in the FFT domain. Since in this section and the next two sections we are concentrating on the single frames, and not on the entire sequence, we use vectors to represent the signals for a given frame. For illustrative purposes we pick a frame length N of 128 samples and a sampling frequency f_s of 20480 Hz. Furthermore, the frames have a 32 sample overlap and the window function is the well known raised cosine window function (Hann window) $h(i) = 0.5 - 0.5\cos(\frac{2\pi(i-1)}{N})$, where i is the frame index going from 1 to N. The speech signal vector for a given frame f multiplied with the h window function is called s_f. Correspondingly the windowed noise vector is called n_f and the windowed composite vector $x_f = s_f + n_f$. The corresponding vectors in the FFT domain are called S_f, N_f and X_f, where the frequency index k also goes from 1 to N. The real valued weights can also be written as a vector w_f and its corresponding FFT is called W_f.

3. OPTIMAL PER FRAME SOLUTION

In the per frame optimal solution, the goal is to find the real valued weights W_f, which must be even symmetric about the Nyquist rate, such that the filter output y_f has the smallest possible MSE when compared to the original windowed signal vector s_f.

3.1 Optimal solution in the time domain

Since the filter process is done in the FFT domain using a multiplication $Y_f(k) = X_f(k)W_f(k)$, the time operation is a circular convolution, which can be written as a matrix operation. I.e., $y_f = Aw_f$, where A is a Toeplitz matrix filled with the elements of x_f. Ideally, $y_f = s_f$, and if this is not possible, then the MSE between y_f and s_f should be minimized. Since N equations have to be satisfied with N variables, the error will be zero, but as we will see later on, the N variables are not independent. The time domain error e_f can be calculated as the difference between the windowed speech and the filter output: $e_f = s_f - Aw_f$. The goal is now to find a solution w_f, such that the squared sum of the error $e_f^T e_f$ is minimized. The well known solution to this MSE problem is

$$w_f = (A^T A)^{-1} A^T s_f. \tag{1}$$

Clearly the weights w_f are all real, but there is no guarantee that, in the FFT domain, W_f will be real. This will only be the case if w_f is even symmetric about the middle point. This can be easily accomplished by reducing the degrees of freedom from N to $N/2+1$, (in our example from 128 to 65). In other words, to force the required symmetry, the following must hold: $w_f(2) = w_f(N), w_f(3) = w_f(N-1), \ldots, w_f(N/2) = w_f(N/2+2)$.

In our example with $N = 128$ we now have only 65 degrees of freedom and clearly the matrix A must be adjusted to reflect that fact by adding the corresponding columns together which results in a temporary matrix T. Now the new MSE solution t for the first 65 dimensions of w_f can be found using the same formula as above, $t = (T^T T)^{-1} T^T s_f$. The remaining coefficients of w_f can then simply be copied from t such that the resulting w_f has the required symmetry.

3.2 Optimal solution in FFT the domain

In the FFT domain, the output of the filter $Y_f = X_m W_f$ can be stated as the multiplication between a diagonal matrix X_m, where the diagonal is the X_f vector and all other elements are zero, and the weight vector W_f. Calculating the squared error in the time domain or in the FFT domain leads to equivalent results, except that the in the FFT domain, the result is scaled by N. In other words, the squared error in the FFT domain needs to be divided by N to result in the same value as in the time domain. The error in the FFT domain can hence be simply calculated by taking the difference between the windowed and transformed speech S_f and the output of the filter in the FFT domain $E_f = S_f - X_m W_f$. The sum of the squared error, which has to be minimized is thus, $E_f^H E_f = (S_f - X_m W_f)^H (S_f - X_m W_f)$.

Since minimizing the MSE in time or in FFT results in the same W_f, the MSE solution is $W_f = (X_m^T X_m)^{-1} X_m^T S_f$. The problem with this solution is that, in general, the weights W_f will not be real, which violates the original assumptions. Fortunately, it is relatively easy to force the solution to become real by converting the complex matrix X_m and the complex vector S_f into a stacked real matrix and a stacked real vector of twice the number of rows. This is achieved by stacking the real parts on top of the imaginary parts. The matrix and the vector resulting from this procedure are then called X_s and S_s, where the s subscript stands for the stacking. Now the MSE problem has the following form $W_f = (X_s^T X_s)^{-1} X_s^T S_s$, which will result in a real W_f that is minimizing the MSE between S_s and $Y_s = X_s W_f$. This MSE is calculated as the sum of the squared differences of all elements between S_s and Y_s. Since the first N elements are the differences between the real parts and the second N elements are the differences between the imaginary parts, the sum over these squared differences is equivalent to $(S_f - Y_f)^H (S_f - Y_f)$, which is the sum of the squared errors in the complex FFT domain.

Since w_f has to be real, W_f has to be even symmetric about the middle point (Nyquist rate). As it turns out, since all the signals involved are real time signals, the above formulation results automatically in an even symmetric W_f. This symmetry results from the matrix inversion but it would be better to force it from the beginning, so that the inversion can be carried out with a matrix of size $(N/2+1) \times (N/2+1)$ versus $N \times N$.

This can be easily accomplished by reducing the degrees of freedom from N to $N/2+1$ (in our example from 128 to 65). In other words, to force the required symmetry, the following must hold: $W_f(2) = W_f(N), W_f(3) = W_f(N-1), \ldots, W_f(N/2) = W_f(N/2+2)$.

In our example with $N = 128$ we now have only 65 degrees of freedom and clearly the matrix X_s must be adjusted to reflect that fact by adding the corresponding columns together, resulting in the temporary matrix T. Now the new MSE solution t for the first 65 dimensions of W_f could be found using the same formula as above: $t = (T^T T)^{-1} T^T S_s$. Since $T^T T$ is a diagonal matrix, the inversion is much simpler and faster than in the time domain. The remaining coefficients of W_f can then simply be copied from t such that the resulting W_f has the required symmetry.

The above result can also be explicitly derived by setting the derivative of the sum of the squared error in the FFT to zero. This derivation leads to a formula which is closely related to the well known Wiener filter formula. Let $E_f = S_f - X_m W_f$ be the error in the FFT domain. Then the goal is to minimize

$$\begin{aligned} E_f^H E_f &= (S_f - X_m W_f)^H (S_f - X_m W_f) \\ &= S_f^H S_f - S_f^H X_m W_f - W_f^H X_m^H S_f + W_f^H X_m^H X_m W_f, \end{aligned} \tag{2}$$

where it is important to note that for real valued weights, $W_f = W_f^H$. This can be achieved by taking the derivative with respect to W_f and setting it to zero. This results in $2X_m^H X_m W_f = (S_f^H X_m)^T + X_m^H S_f$. Since $X_m^H X_m$ is a diagonal matrix, this result can be written for any

given FFT bin k without a matrix inversion as

$$W_f(k) = \frac{S_f(k)^* X_f(k) + X_f(k)^* S_f(k)}{2X_f(k)^* X_f(k)}. \quad (3)$$

Since $X_f = S_f + N_F$, it becomes clear that if the signal and the noise are uncorrelated, averaging over the above formula results in the well known Wiener weights for wide sense stationary processes,

$$W(k) = \frac{S(k)^* S(k)}{S(k)^* S(k) + N(k)^* N(k)}, \quad (4)$$

where the subscript f has been removed to show that these are average values averaged over all frames. Given that the above vectors are known (or estimated) for each frame, the following formula can be used as an instantaneous Wiener filter,

$$W_f(k) = \frac{S_f(k)^* S_f(k)}{S_f(k)^* S_f(k) + N_f(k)^* N_f(k)}. \quad (5)$$

4. WEIGHTED OPTIMAL PER FRAME SOLUTION

In the previous section we have derived the MSE solution for a given WOLA frame. In a WOLA structure, the resulting contribution of each frame to the final sequence is weighted by a function that depends on the analysis window. In this paper we use the raised cosine (Hann) window and in this case the final output of the filter operation y_f is multiplied with this window h and divided by a factor 1.5. In our example, the frames are overlapped by 25%, which means that each final sample is a weighted sum of samples from four consecutive frames. Hence if the weight is high, then the sample in y_f should be as accurate as possible, while when the weight is low, it can be less accurate. Since the weight is known for each sample, (it is the window h divided by 1.5 for the Hann window,) it should be incorporated into the optimization per frame procedure. In other words, the goal should be to minimize the average (sum) of a weighted error squared, where the weight for each sample in y_f is known from the window function.

4.1 Weighted optimal solution in the time domain

As shown in the previous section, the output of the filter is $y_f = Aw_f$, where A is a Toeplitz matrix filled with the elements of x_f. The resulting error is $e_f = s_f - Aw_f$. This error is minimized using the MSE criterion, which is the sum of the squared error which can be written as $e_f^T e_f$. Now the goal is to minimize the sum of the weighted error squared, where the weight is the squared window function h. (Note that the fixed factor 1.5 does not change the optimal solution.) This can be easily achieved by considering a matrix g which is a diagonal matrix with the diagonal equal to h. Then left multiplying the above error equation yields $ge_f = gs_f - gAw_f$. Solving this using the same approach as in the previous section will minimize $e_f^T g^T g e_f$, which was our objective, since this is the sum of the weighted square error where the weight is the window function squared.

4.2 Weighted optimal solution in the FFT domain

As shown in the previous section, the output of the filter is $Y_f = X_m W_f$. Since the weighting is done in the time domain, it becomes a bit more complicated to add this functionality in the FFT domain than it was above in the time domain. The error in the FFT domain can be written as $E_f = S_f - X_m W_f$. Multiplying in time with the window h results in a circular convolution in the FFT domain, divided by the factor N. This can be expressed as a multiplication between a vector and a Toeplitz matrix, which we call Q. Hence the error is now $QE_f = QS_f - QX_m W_f$. Finding the MSE solution (for a real W_f) can now be done just as in the previous section and the optimal solution to the weighted optimization problem in the FFT domain is solved. Note though, that the matrix inversion is now much more involved than for the non-weighted case, since the resulting matrix $T^T T$ is no longer a diagonal matrix.

One could try to derive the above result explicitly by setting the derivative of the sum of the weighted squared error in the FFT to zero. Let $QE_f = QS_f - QX_m W_f$ be the error in the FFT domain. Then the goal is to minimize

$$\begin{aligned} E_f^H Q^H Q E_f &= (QS_f - QX_m W_f)^H (QS_f - QX_m W_f) \quad (6) \\ &= S_f^H Q^H Q S_f - S_f^H Q^H Q X_m W_f \\ &\quad - W_f^H X_m^H Q^H Q S_f + W_f^H X_m^H Q^H Q X_m W_f. \end{aligned}$$

where it is important to note that for real valued weights $W_f = W_f^H$. This can be achieved by taking the derivative with respect to W_f and setting it to zero. This results in $2X_m^H Q^H Q X_m W_f = (S_f^H Q^H Q X_m)^T + X_m^H Q^H Q S_f$. Hence the solution is

$$W_f = (2X_m^H Q^H Q X_m)^{-1} ((S_f^H Q^H Q X_m)^T + X_m^H Q^H Q S_f). \quad (7)$$

Note that the first routine is significantly faster, since it only needs to invert a matrix of dimensions 65×65 and not 128×128 like the routine above. The above formula is nevertheless interesting, since, given that $X_f = S_f + N_F$ and the signal and noise are uncorrelated, averaging over the above formula and applying it framewise results in something we call a weighted instantaneous Wiener filter in the FFT domain,

$$\begin{aligned} W_f &= (2S_m^H Q^H Q S_m + 2N_m^H Q^H Q N_m)^{-1} \quad (8) \\ &\quad ((S_f^H Q^H Q S_m)^T + (S_m^H Q^H Q S_f)), \end{aligned}$$

where the subscript m stands for a diagonal matrix. That is, S_m is a diagonal matrix with S_f in the diagonal and N_m is also a diagonal matrix with N_f in the diagonal.

5. CONSTRAINED WEIGHTED OPTIMAL PER FRAME SOLUTION

We have shown in the previous two sections, that we can find the optimal weights W_f which minimize the weighted MSE. We have formulated the solution in the time and the FFT domain. This optimization was constrained by the conditions that the weights W_f have to be real and even symmetric about the Nyquist rate. The symmetry requirement guarantees a real filter impulse response w_f.

In this section, we introduce additional constraints so that we can compare the performance impact of these constraints with the previous unconstrained methods. There are two sets of constraints that are relevant. First, we have experimentally observed that the unconstrained optimal solutions to W_f contain negative values, which is not possible when the gains are selected using Wiener type schemes. In fact, when a Wiener approach is used, the gains (weights) have to stay between zero and one, since a Wiener filter can only attenuate a particular FFT bin, but never amplify and/or change its sign. It is important to note that these constraints are introduced simply so that we can show their implications on performance, seeing as they are so commonly employed. In a WOLA structure, weights greater than one and smaller than zero are perfectly acceptable and pose no problem for the implementation.

In general, finding the optimal solution to a constrained problem is much harder than finding the optimal solution to a non-constrained problem. Early approaches for solving such constrained problems used to convert a constrained problem to a unconstrained one using Lagrangian multipliers. By moving the constraints to the objective function weighted by Lagrangian multipliers, it becomes possible to solve an unconstrained problem in more variables and deduce the constrained solution from the optimal solution of the relaxed problem.

While these methods work well, they have recently been outperformed by other methods. The Matlab function `quadprog`, which

we use to solve the constrained quadratic programming problem, employs an active set strategy. It operates in two stages, where in a first stage, a feasible point is calculated using the simplex method to solve a linear programming problem. Then in the second stage, it generates iterative solutions using a line search method which converge towards the optimal solution.

This quadratic programming approach is well suited for our problem, since our cost function is quadratic (the MSE) in the variables (W_f). Hence to solve the constraint quadratic program, we use the Matlab function `quadprog`, which allows us to specify constraints on the variables W_f, such as our limits zero and one. Hence in all the above unconstrained MSE optimization tasks, Matlab code like the following would be used to find the MSE solution,

```
t=inv(T'*T)*(T'*(Qs*Ss));
```

where $(Qs*Ss)$ is the particular vector needed to solve the particular problem at hand. This code can now be replaced with the statement,

```
t=quadprog((T'*T),-(T'*(Qs*Ss)),[],[],[],[],
zeros(22,1),ones(22,1));
```

and the constrained optimization problem where zero and one are the limits is solved. As long as $T^T T$ is positive semi-definite, the objective function is convex and an optimal solution will be found. Because of the way the matrix is constructed, it will always be positive semi-definite and hence an optimal constrained solution is always found.

As we mentioned before, the optimal solutions will sometimes pick negative weights in the frequency domain. From a Wiener point of view, it is hard to understand why switching the sign of a frequency bin should be needed to reduce the MSE, nevertheless, the optimal solutions do use negative values from time to time. With the quadratic programming approach, the values can easily be constrained to stay non-negative by simply using the code below,

```
t=quadprog((T'*T),-(T'*(Qs*Ss)),[],[],[],[],
zeros(22,1)).
```

This will allow us to gain an understanding of how much performance is lost by not allowing negative weights (gains).

6. ITERATIVE SCHEME FOR THE ENTIRE SEQUENCE

So far we have only worked one frame at a time. Since the resulting sequence is the weighted sum of all the frames, one should try to find the optimal solution to the entire sequence. Clearly this will result in a large optimization problem, since each frame has 65 (FFT) free parameters and a sequence can consist of an arbitrary number of frames. Furthermore, it is not clear how to formalize the optimal solution, since the frames overlap and hence there are strong dependencies among the frame solutions. In our example, each sample in the final sequence is made up of the weighted sum of four frames. One possible approach would be to employ a dynamic programming strategy, since the current frame depends on a finite number of previous frames. But even with a such an approach, the problem is still quite large.

The method which is presented in this section is based on a simple iterative scheme. The idea is that, for each frame, we calculate what its contribution should be, given that all the other frames are fixed. In other words,we subtract from the clean speech frame the contributions from the neighboring WOLA frames and the remaining residual vector, which we call r_f, becomes the goal of the MSE optimization of the current frame. Since the MSE solution for this frame is the optimal solution for this particular frame, the total MSE (i.e., the MSE between the entire filtered sequence and the original speech signal) becomes smaller or stays the same.

Since we now have a procedure, where each frame can be replaced with a frame that results in the same or a smaller MSE, an iterative scheme that selects frames one-by-one and replaces them with the optimal frame solution (given all the other frames) is guaranteed to converge to a local optima. Furthermore, our experiments show that the starting point is not important for the quality of the final solution.

What is left to show is that the optimal MSE solution for an arbitrary residual vector r_f can be found. If this is possible, then the above iterative scheme is guaranteed to converge to a local optimum. There are many ways one can iterate over the frames of a sequence. All of these strategies have the above property and hence will converge to a local optima. In the implementation used in this paper, we first go over the frames $1,5,9$ etc., then $2,6,10$, etc., then $3,7,11$, etc., and finally $4,8,12$, etc. and then we start again at the beginning. This strategy has the advantage that, with our overlap of 25%, frames are not changed just after a neighboring frame has changed. This seems to lead to faster initial convergence but after a large number of passes over the entire sequence other schemes reach the same quality.

6.1 Optimal solution in the time domain

The output of the filter y_f is weighted by the window function h, which must be taken into account. In other words, since we know that the output will be windowed by h, we know how much bigger y_f has the be at the ends of the frame to match an arbitrary residual vector r_f. Since the output $y_f = Aw_f$, we can simply premultiply the circular convolution matrix A with a diagonal matrix g having h as its diagonal. That way, the product of gAw_f results in an element-wise multiplication of y_f and h which corresponds to the weighting of the output y_f with the window h. Since the goal is to match r_f instead of s_f, the error we need to minimize becomes $e_f = r_f - gAw_f/K$. The factor K equals 1.5 for a Hann window and an overlap of 25%. This is needed since in a WOLA gAw_f the weighted output is divided by this factor K before it is added to the final output sequence.

6.2 Optimal solution in the FFT domain

The above solution can also be found in the FFT domain, following the approach described in the previous sections. The FFT transformed residual vector r_f is called R_f. Hence the error $E_f = R_f - QX_mW_f$, where Q is the Toeplitz matrix introduced before. Note that here, the Toeplitz matrix is used for the weighting of the filter output vector, while before, the same matrix was used for weighting the error. Since for a Hann window, the output of the filter is divided by $K = 1.5$, the matrix is also scaled by this factor.

As before, a solution can also be found minimizing the squared error directly by setting the derivative of the squared error with respect to W_f equal to zero. This results in the following formula,

$$W_f = (2X_m^H Q^H Q X_m)^{-1}((R_f^H Q X_m)^T + X_m^H Q^H R_f). \qquad (9)$$

7. EXPERIMENTS

In this section we present the different experiments we have conducted using the above introduced schemes. These experiments are conducted at a 0dB SNR, which means that the noise power is equal to the signal power. The signal is 30 seconds of a male speaker giving an answer to an interviewer's question, while the noise is either a uniformly distributed white noise or a party noise. The party noise is a mixture of many human voices, such as one would expect to hear in the background at a party. The parameters are the ones we have used throughout the paper, the sampling frequency is 20480 Hz, the frame length is 128 samples, the window is a Hann window and the overlap is 32 samples.

We have introduced many different schemes, and each of them we tested using the above two noise scenarios, white and party noise, and two different frequency domains, the FFT and the Bark domain. Note that, because of the lack of space, and since the derivation in the Bark domain is similar to the derivation in the FFT domain, the Bark domain derivation has not been presented in this paper.

	White noise		Party noise	
	FFT	Bark	FFT	Bark
Inst. Wiener	13.0	12.5	10.1	9.9
$1 \geq$ Optimal ≥ 0	14.0	13.2	10.8	10.7
Weighted inst. Wiener	14.3	13.2	11.3	11.1
$1 \geq$ Weighted optimal ≥ 0	14.5	13.6	11.2	11.1
Optimal ≥ 0	15.3	14.1	12.1	11.8
Optimal	16.1	14.4	12.8	12.4
Weighted optimal ≥ 0	16.9	15.1	13.8	13.3
Weighted optimal	19.4	15.8	15.8	15.0
10 Iterations	34.2	18.9	31.6	23.4

Table 1: The MSE improvement in dB using the proposed filters

The different schemes are: the instantaneous Wiener filter (Eq.5), the optimal scheme (Eq.3), the weighted optimal scheme (Eq.7), the weighted instantaneous Wiener scheme (Eq.8) and the iterative scheme (Eq.9). We furthermore conducted experiments using constrained quadratic programming for the optimal and the weighted optimal schemes, where two different constraint sets were used: W_f as non-negative and W_f as between zero and one.

Table 1 shows the results of the different experiments. As a quality metric, the MSE between the final sequence and the original speech signal is taken and converted to dB using the following formula: $10\log_{10}(\frac{1}{MSE})$. Since we normalized the original noise (and also the original signal) to have zero mean and a variance of one, the unfiltered signal has an MSE of one (i.e., the variance of the noise). The above formula gives therefore the improvement in the MSE because of the filtering procedure in dB.

Table 1 shows several facts very clearly. First, better noise reduction can be achieved in the FFT domain than in the Bark domain. Since the weights are found optimally, and we have more free parameters in the FFT domain than in the Bark domain, this does not come as a surprise. Second, the iterative approach can result in very good noise reduction. In fact, for the FFT domain, the reduction can be so good, that one cannot acoustically distinguish the original from the filtered speech. Again, this is because of the 65 free parameters, which concentrated on matching about half of the frame (64 samples). For the Bark domain, this is different, since here only 22 parameters are available for the 64 samples. Hence one converges to a lower dB improvement, which is still quite good. Nevertheless there is still an acoustically recognizable difference to the clean speech. Third, constraining the weights W_f to be positive results in a performance drop. This drop is more significant for the weighted optimal solution (1.5dB on average) than for the optimal one (0.6dB on average). This performance drop is even more severe when we constrain the weights to be between zero and one, which is the usual solution of a Wiener type filter. In this case the weighted optimal solution loses on average 3.8dB and the optimal solution 1.75dB. In the most severe case (FFT domain, white noise, weighted optimal solution) the performance loss is 4.9dB, which is a significant performance loss. In other words, one might want to consider schemes where the weights (gains) do not have to be between zero and one. Fourth, it is interesting to observe the performance gain from the optimal scheme to the weighted optimal scheme. The gain is on average 2.6dB, which again is quite significant, considering the fact that these are basically identical schemes, except that the weighted optimal scheme is focusing on reducing the squared error in the middle of the frame while the optimal scheme minimizes the squared error over the entire frame. The gain is not as large (1.1dB) when comparing the instantaneous Wiener filter with the weighted instantaneous Wiener filter, but nevertheless, the gain is there.

Figure 1 shows 300ms (about 6000 samples, i.e., about 200 frames) of the difference between the WOLA output and the clean speech, where the noise is a uniformly distributed white noise. Iteration 0 means that we have simply used the weighted optimal method in the FFT domain and no additional pass has been made

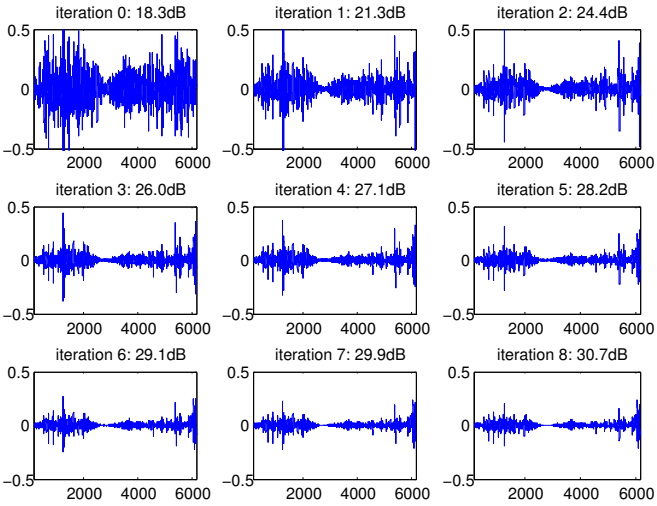

Figure 1: Convergence of the iterative scheme in the FFT domain

over the sequence using the iterative scheme. Iteration x means that x passes have been made over the entire sequence using the iterative scheme in the FFT domain. To indicate the convergence in MSE reduction, next to the iteration number the delta MSE in dB is stated. As Figure 1 clearly shows, the iterative scheme converges to a local optima.

8. SUMMARY AND CONCLUSIONS

In this paper, the aim was to find the limits of the well known and widely used WOLA structure for the task of de-noising corrupted speech. To this end, we assumed that the estimation part of a complete noise cancelling system would be ideal, i.e., it can deliver the filtering part with whatever information is needed to accomplish its task. The focus was on the filtering part, to be exact, on the performance limits of a WOLA structure.

We showed that the WOLA structure itself does not impose a severe limit to the ability of speech de-noising. In fact, staying in the FFT domain and using an iterative scheme, we were able to produce WOLA weights which could produce filtered speech that was acoustically indistinguishable from the clean speech.

Several different schemes were developed. The optimal solution to these schemes were always developed in the time and the FFT domain. We derived a frame-wise optimal solution that will result in the well known instantaneous Wiener filter, when we assume that the noise and the speech are uncorrelated and when we average over all frames. Furthermore, we also developed a so-called weighted optimal solution, where the MSE is minimized in the middle of the frame more than at the edges, and showed experimentally that this results in better filtered sequences. We then showed that by averaging over all frames, a kind of weighted instantaneous Wiener filter results, which also outperforms the instantaneous Wiener filter.

Another major point was that we developed a quadratic programming solution to solve the above problems, given some constraints on the weights W_f. These optimal constraint solutions showed that constraining W_f to be between zero and one comes with a relative high penalty in performance. Since most WOLA based noise cancelling schemes do this today, this is an important result.

REFERENCES

[1] R. Crochiere, "A weighted overlap-add method of short-time Fourier analysis/synthesis," *IEEE Transactions on Acoustics, Speech, Signal Processing*, vol. 28, pp. 99–102, Feb. 1980.