# Speech Enhancement with Spectral Magnitude Side Information

by

Charles Kasimer Sestok IV

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1999



Author ..........................................................
Department of Electrical Engineering and Computer Science
February 12, 1999

Certified by ..........................................................
Alan V. Oppenheim
Ford Professor of Engineering
Thesis Supervisor

Accepted by ..........................................................
Arthur C. Smith
Chairman, Department Committee on Graduate Students

# Speech Enhancement with Spectral Magnitude Side Information

by

Charles Kasimer Sestok IV

Submitted to the Department of Electrical Engineering and Computer Science
on February 12, 1999, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering

## Abstract

The work described in this thesis examined a new approach to speech enhancement. Enhancement algorithms to estimate clean speech from a noisy signal and a limited amount of side information were developed and implemented. The work evaluated algorithms that used linear predicitive (LP) coefficients and zero-phase impulse response coefficients as side information. An approximate maximum likelihood estimator for the case of LP coefficient side information, an exact ML estimator for the case of zero phase impulse response side information, and a linear least squares error (LLSE) estimator for the case of LP side information were implemented and tested.

Thesis Supervisor: Alan V. Oppenheim
Title: Ford Professor of Engineering

# Acknowledgments

There are a lot of people who helped make this thesis possible. First and foremost, I'd like to thank my advisor, Professor Alan Oppenheim for his support and encouragement during the course of this research. His skill and insight helped make this work a very educational experience. All that and he taught me how to spell "likelyhood," too! Richard Barron also deserves a great deal of credit for helping me with this research. He developed the theory behind the algorithms I tested, and was always a source of sound advice. The students, faculty, and staff in DSPG also helped make this a quality experience. They all contributed to an excellent environment in which to learn and do research.

Finally, I'd like to acknowledge my Mother and Father, and my brother Evan. Without their love and support, I wouldn't be where I am today.

# Contents

# List of Figures

# Chapter 1

# Introduction

Processing speech to reduce the impact of additive noise is a long-standing research problem. Additive noise models many disturbances that can impair the performance of speech processing systems. For example, the fidelity of a speech compression system's output may degrade rapidly as background interference is added to the input. Additionally, persistent distortions in a speech signal, even if minor, can annoy human listeners. An appropriate enhancement system can improve the fidelity of the vocoder output or make the distorted speech more pleasant to hear. Thus, speech processing systems intended to operate in the presence of disturbances often incorporate noise reduction algorithms.

Noise reduction is a subset of the general enhancement problem. Enhancement systems process a speech signal to improve its characteristics for human listeners or use in another speech processing algorithm [5]. In addition to reducing additive noise, enhancement systems reduce distortions such as degradations from a communications channel or reverberations. Enhancement systems may even process an undistorted signal to improve its clarity.

The performance of an enhancement system is judged by the application for which it is intended. Many enhancement systems are designed to improve the characteristics of speech for human listeners. In other cases, the enhancement system may be intended to improve the robustness of another speech processing algorithm, such as a vocoder or recognition system. In these cases, the speech enhancement system succeeds if it produces output perceptually pleasing to human listeners or allows a subsequent processing system to operate successfully in the presence of disturbances. These goals are difficult to describe using quantitative measurements of the output of the enhancement system. Even systems designed through optimization of some function of the output may perform poorly in practice. Thus, many enhancement systems are evaluated through qualitative testing.

Since accurate performance criteria for enhancement systems are difficult to develop, many enhancement systems are designed using models based on the physics of speech production. A quasi-stationary linear system driven by appropriate excitations captures many key features of speech acoustics [12],[8]. The frequency response of the filter models the resonant frequencies of the vocal tract, and the filter input approximates the excitation from the glottis. Pulse train excitations model voiced sounds and white noise excitations model unvoiced sounds. If the vocal tract filter has only poles, this model reduces to linear prediction. Well-known algorithms can calculate the filter taps that best model a speech waveform [11], [12]. Linear predictive (LP) models have been widely used because they capture the structure of speech, lead to tractable calculations, and perform well in practice.

Some noise reduction algorithms exploit LP models of speech [6]. For example, LP analysis/sythesis has been applied to noise reduction. Algorithms that estimate LP parameters from the noisy speech and then synthesize an enhanced speech waveform have been used for several decades. Unfortunately, as the signal-to-noise ratio (SNR) falls, estimating the model parameters becomes difficult. The inaccurate coefficients limit the quality of the enhanced speech.

The robustness of noise reduction algorithms can be improved by by obtaining more information about the clean speech. A noise reduction system can exploit side information describing the clean speech to form acceptable output at SNRs where single channel algorithms perform poorly. If performance in this SNR range is important, the cost of the extra hardware needed to obtain the side information may be justified.

Contexts in which an enhancement system has side information about the clean speech can arise in distributed sensor or communications networks. For example, a distributed sensor network may have multiple sensors observing a signal source. Some sensors may be far from the source and consequently are disturbed by background interference. Other sensors may be close to the source and therefore receive the signal with minimal interference. If limited communication resources prevent transmission of the clean signal to a human listener at a far field sensor, a low-bandwidth side information signal can be transmitted instead. An enhancement system can use the noisy signal from the far field sensor and the side information from the near field sensor to improve the speech signal. In a communication scenario, a hybrid analog/digital channel can use side information. A speech signal transmitted over an analog channel encounters distortions that are modeled as additive noise. A low-bandwidth digital link from the signal source to the receiver can transmit side information without degradations. Using such a link, an enhancement system can combine the analog signal and the side information on the digital channel to estimate the clean signal. It is possible to imagine other scenarios where side information is available to enhancement systems.

This thesis examines the noise reduction problem when side information describing the clean speech is available. In certain situations, a reliable side channel between the signal source and receiver is available. It is assumed that the side channel has limited capacity and cannot transmit the entire speech signal. Instead, a compressed version of the clean speech is sent to the receiver. There, the noisy signal and the side information are combined to estimate the speech.

The side information estimation problem can be formulated in a variety of ways. Depending on the models chosen for the speech and side information, many estimators can be used to design enhancement algorithms. This thesis tests algorthms based on the side information estimation framework developed by Barron, *et. al.* [1]. In general, the enhancement problem involves the choice of side information and the design of an enhancement algorithm. Barron, *et. al.* simplified the problem through an ad hoc choice of side information. They used the linear predictive (LP) coefficients of the clean speech as the basis for an iterative algorithm to calculate maximum likelihood (ML) estimators for clean speech degraded by additive white Gaussian noise (AWGN).

These ML estimators are compared with linear enhancement algorithms based upon Wiener filtering. Since the LP coefficient side information is related to the power spectrum of the speech, it can be used to generate an approximate Wiener filter for the noisy signal. This algorithm is not iterative and has much lower complexity than the ML algorithms. The comparison tests whether the added complexity of the iterative ML algorithms is justified.

The ML and Wiener filter estimation algorithms were implemented and their outputs were compared. Additionally, their performance was compared with single channel iterative Wiener filtering algorithms over a range of SNRs. At SNRs greater than 20dB, the side information algorithms produced outputs of comparable quality to the single channel algorithm. At SNRs below 0dB, the output from the algorithms using side information was preferable to the output from iterative Wiener filtering. As the SNR was lowered from this point, the artifacts produced by the algorithms increased, specifically, the ML estimation algorithms produced high frequency tones that distracted the listeners. The Wiener filter muffled the high frequency characteristics of the speaker's voice. Generally, the tonal artifacts from the ML algorithm were more distracting to human listeners.

As an auxiliary study, the ML algorithm was applied to single channel noise reduction. An LP coefficient model was generated from the noisy speech and used in place of the side information. This approach was compared with iterative Wiener filtering. In informal tests at SNRs below 10 dB, the ML algorithm again produced tonal artifacts and the iterative Wiener filter muffled the speech. Despite the presence of artifacts, the speech estimate from the modified side information algoirthm was judged to be clearer than iterative Wiener filtering.

# Chapter 2

# Estimation and Side Information

The side information algorithms in this thesis use additional communication capacity between the signal source and receiver to improve the performance of a noise reduction system. The extra capacity is assumed to be expensive, so the cost of transmitting a significant portion of the clean speech on the side channel is prohibitive. In our work, the LP coefficients of the clean speech are used as side information due to their successful application in other speech processing systems. With the additional information from this side channel, estimation algorithms performed by the receiver ought to achieve higher speech quality than single channel algorithms.

This chapter explains the side information estimation work of Barron, *et. al.*, which presents the general form of estimators with side information and develops algorithms to calculate the ML and linear least squared error (LLSE) estimates of the clean speech given the LP parameters. The first section derives the maximum *a posteriori* probability (MAP), minimum mean squared error (MMSE), and ML estimators using deterministic side information. Next, simplifications to the ML estimator in the presence of AWGN are described and an iterative algorithm to calculate it is developed. Finally, the Wiener filter is derived from the LP parameters of the speech.

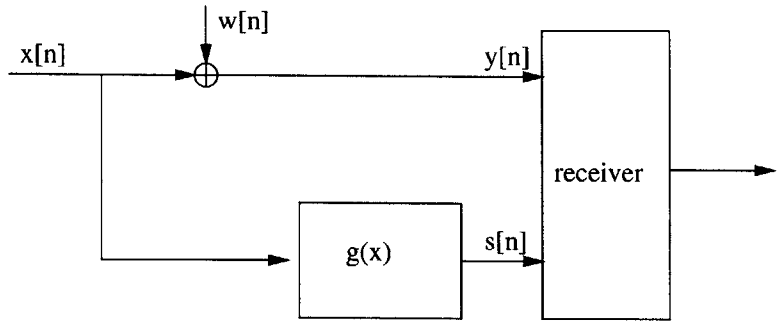


Figure 2-1: Side Information Framework

## 2.1 MAP, MMSE, and ML estimators

The LP side information used in the estimation algorithms is a deterministic function of the clean speech signal. Figure 2.1 shows the side information framework in terms of a communication system. The vector $x$ represents the signal of interest. It is distorted during transmission on a communications channel. The distortions are modeled by an additive random noise signal $w$. The random vector $y$ represents the output of the noisy channel. The transmitter performs some deterministic processing of the original signal before transmission, producing the side information signal $s$. The side information is transmitted over a reliable channel and is received without errors.

The deterministic side information imposes constraints on the possible estimates for the clean speech waveforms. In general, a many-to-one function, $g(x) = s$, generates the side information. There is a unique value of $s$ for every clean speech signal $x$. Multiple waveforms, however, may be mapped to the same value of $s$. When the side information is observed, the space of possible $x$ values can be divided into two sets. The first set, $\mathcal{S} = \{x|\ g(x) = s\}$, contains all possible clean speech signals given $s$. Its complement contains the $x$ values ruled out by the side information. If $x$ is modeled as a random process, the side information restricts the set of signals that have non-zero probability of being transmitted. Instead, if $x$ is modeled as an unknown, non-random signal, the side information reduces its set of possible values to the elements of $\mathcal{S}$. The restrictions on the values of $x$ give the estimators more information about the clean speech and increase their accuracy.

The model chosen for the speech waveform determines which estimators can be used as a basis for designing noise reduction algorithms. When $x$ is modeled as a random process, the MAP and MMSE are two common estimators for the clean speech. Under this model, the original signal $x$, the noisy channel output $y$, and the side information $s$ are related by a joint probability density $p(x, y, s)$. Both estimators depend on the *a posteriori* density for $x$ given the observation of $y$ and $s$, $p(x|y, s)$. The MAP estimate is the mode of the density and the MMSE estimate is the conditional expectation of $x$. The *a posteriori* density is

$$p(x|y,s) = \frac{p(x,y,s)}{p(y,s)} = \frac{p(y)p(x,s|y)}{p(y)p(s|y)}. \tag{2.1}$$

Since $y$ and $s$ are generated from $x$ by unrelated processes, they are conditionally independent given $x$. Knowing $s$ does not affect the distribution for $y$ if $x$ is already known. Likewise, $y$ does not give extra information about $s$ if $x$ is known. Thus, the probability density $p(x, s|y)$ can be simplified to

$$p(x,s|y) = p(x|y)p(s|x,y) = p(x|y)p(s|x).$$

Using this result, the density in equation 2.1 is

$$p(x|y,s) = \frac{p(x|y)p(s|x)}{p(s|y)}. \tag{2.2}$$

With the observed data $s$ and $y$, the denominator of equation 2.2 is a constant. Additionally, the density $p(s|x)$ is 0 if $x \notin \mathcal{S}$ and is 1 otherwise. The traditional form of the MAP estimator can be used as long as the probability density $p(x|y)$ is maximized over the set $\mathcal{S}$ defined by the side information $s$. The formula is

$$\hat{x}_{MAP} = \underset{x \in \mathcal{S}}{\text{argmax}} \; p(x|y). \tag{2.3}$$

The new MMSE estimator is the conditional expectation of $x$ given $s$ and $y$. The estimator is

$$\hat{x}_{MMSE} = E[x|s,y] = \frac{1}{p(s|y)} \int_{x \in \mathcal{S}} x p(x|y) dx. \tag{2.4}$$

The estimator is proportional to the centroid of the distribution $p(x|y)$ over the region $\mathcal{S}$. If $\mathcal{S}$ is not a convex set, then $\hat{x}_{MMSE}$ may not belong to $\mathcal{S}$.

When $x$ is modeled as a non-random parameter, the ML estimator can be used. In this case, the signals $s$, $y$, and $x$ are related by a parameterized probability density. The observations $y$ and $s$ are random quantities and the signal $x$ is an unknown parameter of the probability density $p(y, s; x)$. The ML estimator, $\hat{x}_{ML}(y, s) = \underset{x}{\operatorname{argmax}} \; p(y, s; x)$, now depends on the observations of both random vectors.

The side information restricts the ML estimate to a maximum over the constraint set $\mathcal{S}$. The likelihood function is

$$\begin{aligned} p(y, s; x) &= p(y; x) p(s|y; x) \\ &= p(y; x) p(s; x) \\ &= \begin{cases} p(y; x) & x \in \mathcal{S} \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{2.5}$$

The second equality follows because $y$ and $s$ are conditionally independent given the parameter $x$ and the third equality follows because $p(s; x)$ is 1 over $\mathcal{S}$ and zero otherwise. The ML estimate is obtained by maximizing $p(y; x)$ over the set $\mathcal{S}$. It is

$$\hat{x}_{ML}(y, s) = \underset{x \in \mathcal{S}}{\operatorname{argmax}} \; p(y; x). \tag{2.6}$$

## 2.2 ML Estimation in Additive White Gaussian Noise

Barron, *et. al.* present ML algorithms using two related forms of side information. The first calculates an approximate ML estimator using LP side information. The second algorithm calculates an exact ML estimate using the zero-phase impulse response as side information. Both algorithms are similar in structure. They express the constraints imposed by the side information as the intersection of several convex sets and use an iterative projection algorithm to calculate the minimum distance projection onto this intersection.

### 2.2.1 ML Estimation with LP Coefficient Side Information

In the case of estimation in additive white Gaussian noise, the ML estimator simplifies to a minimum distance projection. The noise vector $w = y - x$ models the difference between the clean and noisy speech. For a zero-mean, AWGN model, its probability distribution is given by

$$p(w) = (2\pi\sigma_w^2)^{-\frac{N}{2}} \exp\left(-\frac{1}{2\sigma_w^2}\sum_{i=0}^{N-1} w^2[i]\right).$$

The distribution of the received data is

$$p(y) = (2\pi\sigma_w^2)^{-\frac{N}{2}} \exp\left(-\frac{1}{2\sigma_w^2}\sum_{i=0}^{N-1} (y[i] - x[i])^2\right).$$

If $x$ is viewed as a parameter, this probability density is also the likelihood function $p(y; x)$. The LP coefficient side information defines a constraint set $\mathcal{S}$ of allowable clean signals. By equation 2.6, the ML estimate maximizes $p(y; x)$ over the set $\mathcal{S}$. The maximum occurs when the term $\sum_{i=0}^{N-1}(y[i] - x[i])^2$ is minimized. This term is the Euclidean distance between $y$ and a signal in the constraint set. Thus, the ML estimator is given by

$$\hat{x}_{ML} = \underset{x\in\mathcal{S}}{\text{argmin}} \sum_{i=0}^{N-1} (y[i] - x[i])^2. \tag{2.7}$$

The minimization in equation 2.7 can be analyzed in the frequency domain. By Parseval's Theorem, the ML estimate is

$$\hat{X}_{ML}(e^{j\omega}) = \underset{X\in\mathcal{S}}{\text{argmin}} \int_{-\pi}^{\pi} \left|X(e^{j\omega}) - Y(e^{j\omega})\right|^2 d\omega. \tag{2.8}$$

This estimate minimizes the distance between $X(e^{j\omega})$ and $Y(e^{j\omega})$ at each frequency. The integrand $|X(e^{j\omega}) - Y(e^{j\omega})|^2$ is always positive. If it is greater than the minimum value at any frequency, the integral can be further reduced. At a specific frequency $\omega_0$, the Fourier transforms are $X(e^{j\omega_0}) = |X|e^{j\phi_X}$ and $Y(e^{j\omega_0}) = |Y|e^{j\phi_Y}$. The distance

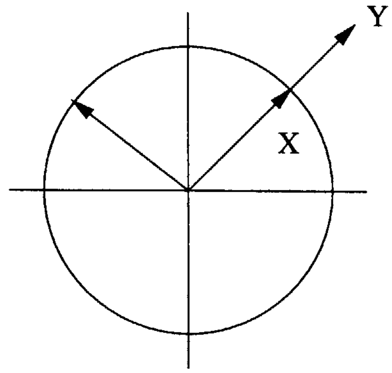


Figure 2-2: Minimum distance between $|X|$ and $|Y|$ at a fixed frequency.

at a fixed frequency is

$$\begin{aligned}
|X(e^{j\omega}) - Y(e^{j\omega})|^2 &= (|X|e^{j\phi_X} - |Y|e^{j\phi_Y})(|X|e^{-j\phi_X} - |Y|e^{-j\phi_Y}) \qquad (2.9)\\
&= |X|^2 + |Y|^2 - |X||Y|e^{j(\phi_X - \phi_Y)} - |X||Y|e^{j(\phi_Y - \phi_X)}\\
&= |X|^2 + |Y|^2 - 2|X||Y|\cos(\phi_X - \phi_Y).
\end{aligned}$$

For fixed $\omega_0$, the distance is minimized by choosing a vector with the same phase function as the noisy speech, $y$. For any choice of $|X|$, the third term of equation 2.9 is minimized when $\phi_X = \phi_Y$. This choice of phase does not conflict with the side information constraint set. LP side information does not restrict the phase of the signal, so any choice of phase is possible as long as $|X(e^{j\omega})|$ belongs to the constraint set.

Figure 2.2 shows the situation at a specific value of frequency. The large vector represents $Y(e^{j\omega_0})$, and the circle represents a constraint on $|X(e^{j\omega_0})|$. The distance between the estimate and the received value is minimized when the two vectors are collinear. Thus, the phase of the estimate and the phase of the received signal ought to be the same.

Once the phase has been chosen, the magnitude function that minimizes equation 2.7 must be determined. It can be calculated by a minimization over the set of all allowed magnitude functions, $\mathcal{S}' = \{x \in \mathcal{S}| \ \phi_X(\omega) = \phi_Y(\omega)\}$. The desired magnitude function minimizes the expression

$$\int_{x \in \mathcal{S}'} \left| |X(e^{j\omega})| - |Y(e^{j\omega})| \right|^2 d\omega. \tag{2.10}$$

The distance measure in expression 2.10 does not permit an easy solution. The minimization takes place over a constraint set that is not a vector space, so standard quadratic optimization algorithms cannot be applied. The set $\mathcal{S}'$ requires the signal with magnitude $|X(e^{j\omega})|$ to have the LP coefficients given in $s$. This in turn requires the signal to match the deterministic autocorrelation coefficients defined by the LP coefficients. A set of vectors with matching autocorrelation coefficients, unfortunately, is not a vector space. If both $x$ and $y$ are members of $\mathcal{S}'$, the sum may not be. The autocorrelation of $x + y$ contains cross correlation terms that may be non-zero.

Barron proposed a modified distance measure that leads to a soluble minimization problem. The distance measure

$$\int_{x \in \mathcal{S}'} \left| |X(e^{j\omega})|^2 - |Y(e^{j\omega})|^2 \right|^2 d\omega, \tag{2.11}$$

maps directly to a problem that has a convenient characterization in terms of the LP coefficients. Using Parseval's Theorem, the form of the modified distance measure is

$$\hat{x} = \underset{x \in \mathcal{S}'}{\operatorname{argmin}} \sum_i K_{xx}[i] - K_{yy}[i]. \tag{2.12}$$

Here, $K_{xx}[i]$ and $K_{yy}[i]$ are the autocorrelation functions of the estimate and received data vectors. The set $\mathcal{S}'$ maps to a constraint on the autocorrelation $K_{xx}[i]$. The LP parameters transmitted as side information are directly related to the autocorrelation of the clean signal, $x_0$. If the side channel transmits $M + 1$ LP parameters, the autocorrelation from $K_{xx}[-M]$ to $K_{xx}[M]$ is fixed. For this choice of side information, all allowable values of the autocorrelation for the estimate $x$ are contained in the constraint set $\mathcal{S}_K = \{K_{xx} | \ K_{xx}[i] = K_{x_0 x_0}[i], \ i = -M, \ldots, M\}$. The estimate in equation 2.12 is the minimum distance projection of $K_{yy}$ onto the set $\mathcal{S}_K$.

The new distance measure does not produce an ML estimate of $|X|$. There is not a monotonic mapping from each element in the set under the first distance measure in expression 2.10 to the modified distance measure in equation 2.11. The estimator derived from this distance measure does not minimize the same distance measure as the true ML estimate.

### 2.2.2 Projection onto Convex Sets

The autocorrelation function that minimizes the distance in equation 2.12 can be calculated through projection onto convex sets (POCS). The constraints imposed on the autocorrelation function of any signal $x \in \mathcal{S}_{\mathcal{K}}$ can be expressed as a group of convex sets in the Hilbert space of finite norm sequences ($l^2$). Any sequence in $\mathcal{S}_{\mathcal{K}}$ will have an autocorrelation sequence that lies in the intersection of the sets. The orthogonal projection of the noisy speech's autocorrelation onto the intersection solves equation 2.12. With proper corrections, a sequence of projections onto the individual constraint sets will converge to this solution [2]. Since there is a 1-to-1 relationship between the autocorrelation function and the spectral magnitude of the sequence, this iteration calculates the magnitude function that solves equation 2.11.

A sequence that is a valid autocorrelation function of an element in $\mathcal{S}_{\mathcal{K}}$ must satisfy several constraints. First, it must be real and even, and possess a real, even, and positive Fourier transform. Additionally, the value at sample 0 must be the maximum value of the sequence. Once the sequence is a valid autocorrelation, it must match the constraints from the side information, i.e. it must belong to the set $\mathcal{S}_{\mathcal{K}}$.

Two constraint sets guarantee that these conditions are fulfilled. The two constraint sets are

$$C_1 = \{u \in l^2 \mid u[i] = K_{x_0 x_0}[i],\ i = -M, ..., M\} \tag{2.13}$$

$$C_2 = \{u \in l^2 \mid U(e^{j\omega}) \text{ real, positive } \forall \omega\}. \tag{2.14}$$

The first set guarantees that the sequence belongs to $\mathcal{S}_{\mathcal{K}}$. The second set ensures that it is a legitimate autocorrelation function. If a sequence is in $C_2$, its Fourier transform is real and positive, and its maximum sample is $u[0]$. Since $u$ is real, its Fourier transform satisfies $U(e^{j\omega}) = \Re\{U(e^{j\omega})\}$ and is even. Thus the sequence is

$$u[n] = 2\int_0^{\pi} U(e^{j\omega})\cos(\omega n)d\omega. \tag{2.15}$$

This expression is maximized when $n = 0$. Thus, these two sets guarantee that an element in their intersection is a valid autocorrelation function for a sequence in $\mathcal{S}_K$.

Both sets are convex. If $a[i] \in C_1$ and $b[i] \in C_1$, then $\lambda a[i] + (1-\lambda)b[i]$ satisfies the condition for $C_1$ since $a[i] = b[i] = K_{x_0x_0}$ for $i = -M, \ldots, M$. The convexity of $C_2$ follows directly from the linearity of the Fourier transform.

The minimum distance projections onto each set are written $P_i$, where $P_i$ is an orthogonal projection to the set $C_i$. These projections guarantee that the result is the minimum distance from the initial vector to the convex set. The projection $P_1$ is best defined in the time domain, and the projection $P_2$ is best defined in the frequency domain. The projection operations are

$$P_1u[n] = \begin{cases} K_{x_0x_0}[n] & n = -M, ..., M \\ u[n] & \text{otherwise} \end{cases} \tag{2.16}$$

$$P_2U(e^{j\omega}) = \begin{cases} U(e^{j\omega}) & \text{if } U(e^{j\omega}) > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{2.17}$$

A conventional POCS algorithm starts with $u_0 = K_{yy}$ and uses the iteration $u_{i+1} = P_1P_2u_i$. The sequence $\{u_i\}_{i=0}^{\infty}$ converges to some point in the intersection. As an example, consider figure 2.3. It shows a two dimensional case where the POCS algorithm converges to a point in the intersection of $C_1$ and $C_2$ after one iteration. The fixed point, however, is not the minimum distance projection from the starting point to the intersection of the sets.

A correction term in the iteration forces the algorithm to converge to the minimum distance projection onto the constraint sets. The modified algorithm starts with

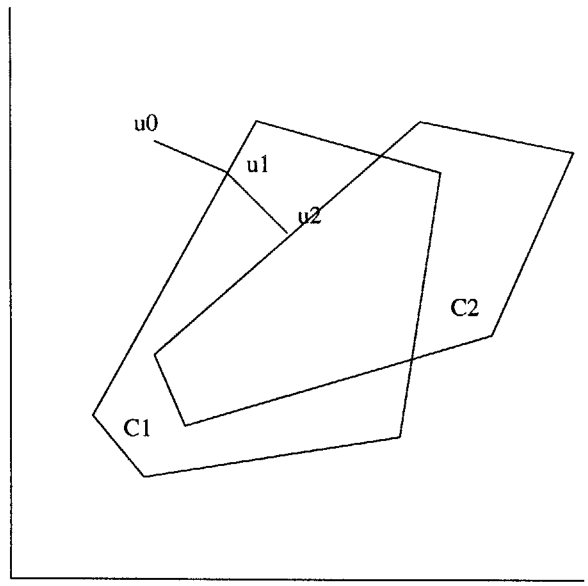


Figure 2-3: Standard POCS procedure does not converge to minimum distance projection.

$u_0 = K_{yy}$ and performs the iteration

$$
\begin{aligned}
u_1 &= P_1 u_0, & v_1 &= u_1 - u_0 \\
u_2 &= P_2 u_1, & v_2 &= u_2 - u_1 \\
u_3 &= P_1(u_2 - v_1), & v_3 &= v_1 + u_3 - u_2 \\
u_4 &= P_2(u_3 - v_2), & v_4 &= v_2 + u_4 - u_3 \\
u_5 &= P_1(u_4 - v_3), & v_5 &= v_3 + u_5 - u_4 \\
u_6 &= P_2(u_5 - v_4), & v_6 &= v_4 + u_6 - u_5 \\
u_7 &= P_1(u_6 - v_5), & v_7 &= v_5 + u_7 - u_6 \\
u_8 &= P_2(u_7 - v_6), & v_8 &= v_6 + u_8 - u_7 \\
&\vdots & &\vdots
\end{aligned}
\tag{2.18}
$$

The limit of the projections, $u = \lim_{i \to \infty} u_i$, is the desired autocorrelation function. Its Fourier transform is the squared magnitude function that minimizes expression 2.11.

The correction terms prevent convergence to an undesired fixed point. If $u_i$ is in the intersection of the constraint sets, the correction term moves it out of one of the sets and allows the iteration to continue. Figure 2.4 shows an example of the

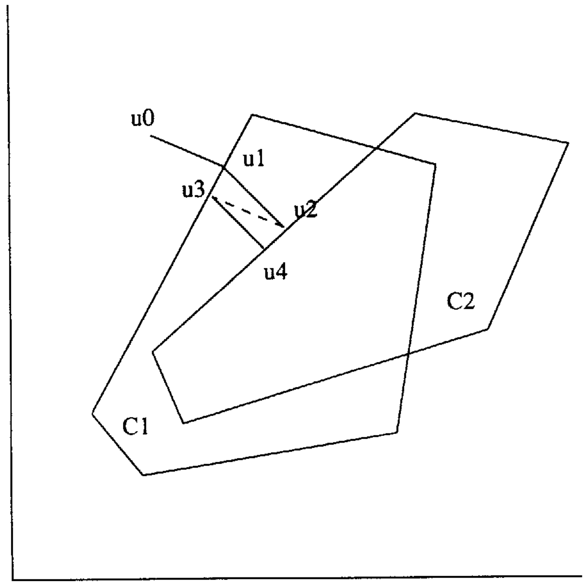


Figure 2-4: POCS with correction term converges to a minimum distance projection.

correction term action. The sets and starting point are the same as figure 2.3. In the figure, the correction term moves the iteration from $u_2$ to a point inside $C_1$. From this point, a projection onto $C_2$ produces the minimum distance projection onto the intersection.

### 2.2.3 Maximum Likelihood Estimation for Zero-Phase Impulse Response Side Information

Barron, *et. al.* also suggest modification of the projection algorithm that produces a true ML estimate of the clean speech. The POCS algorithm was obtained by modifying the distance measure in expression 2.10 to simplify the side information constraint. The constraint can also be simplified by modifying the side information. A problem with similar structure to equation 2.12 results when zero-phase impulse response coefficients are transmitted on the side channel.

The zero-phase impulse response of a signal is the inverse Fourier transform of its spectral magnitude. For a signal $x[n]$, the zero-phase impulse response is $x_{zp}[n] = F^{-1}\{|X(e^{j\omega})|\}$. For real sequences, it is a real, even sequence with a real, even, and positive Fourier transform.

This sequence contains information similar to the LP coefficients of the clean speech. There is a 1-to-1 mapping between the Fourier transforms of the autocorrelation function and the zero-phase impulse response. The transform of the autocorrelation sequence is $|X|^2$ and the transform of the zero-phase impulse response is $|X|$. Even though it is not derived directly from a physical model for speech, the zero phase impulse response is closely related to the LP parameters, and it provides similar constraints on the clean speech estimates.

The zero phase side information signal has a direct relationship to the distance measure in expression 2.10. In the time domain, the criteria for an ML estimator is

$$\hat{x} = \underset{x \in S'}{\operatorname{argmin}} \sum_{i} x_{zp}[i] - y_{zp}[i]. \tag{2.19}$$

This structure is analogous to the condition in equation 2.12. The distance measure depends on the Euclidean distance between $x_{zp}$ and $y_{zp}$, and the side information fixes $x_{zp}$ over the range $i = -M, \ldots, M$.

The POCS algorithm developed for the LP side information problem can solve equation 2.19. Changes in the initialization of the algorithm and the constraint set $C_1$ produce an algorithm that calculates the minimum distance projection for zero-phase impulse response side information. The new constraint set enforces the restrictions imposed by the side information. It is

$$C_1 = \{u \in l^2 \mid u[i] = x_{0,zp}[i],\ i = -M, \ldots, M\}. \tag{2.20}$$

The new projection in the time domain is

$$P_1 u[n] \quad = \quad \begin{cases} x_{0,zp}[n] & n = -M, \ldots, M \\ u[n] & \text{otherwise.} \end{cases} \tag{2.21}$$

The second constraint set and projection are unchanged because the frequency domain properties of the autocorrelation and zero-phase impulse response sequences are the same. The algorithm is initialized with $u_0 = y_{zp}$ and the iteration follows equation 2.18 with the new $P_1$. The iteration ultimately converges to the sequence that solves equation 2.19.

## 2.3 Linear Least Squares Estimator for LP Side Information

A linear least squared error (LLSE) estimator was also constructed for the LP side information. The LP coefficients give information on the power spectrum of the original signal. Using the random process model of speech, a speech vector is assumed to be the output of an all-pole filter driven by white noise. For a set of LP coefficients $\{\sigma^2, a[k], k = 1, \ldots, M\}$, the transfer function modeling the vocal tract is

$$A(e^{j\omega}) = \frac{\sigma^2}{1 - \sum_{k=1}^{M} a[k] e^{j\omega}}. \tag{2.22}$$

When a random process with power spectrum $S(e^{j\omega})$ is filtered by the vocal tract filter, the resulting power spectrum is $|A(e^{j\omega})|^2 S(e^{j\omega})$. Since the model assumes that the filter's input is unit variance white noise, the output power spectrum is $|A(e^{j\omega})|^2$. The LP side information is sufficient to estimate the power spectrum of the original signal. Additionally, the variance of the channel noise can be calculated when the transmitter is silent. Thus, the receiver has all of the information necessary to compute a Wiener filter for the noisy signal.

A non-causal Wiener filter for a received speech vector can be constructed in the frequency domain. The frequency response of the non-causal Wiener filter is

$$H(e^{j\omega}) = \frac{S_{xy}(e^{j\omega})}{S_{yy}(e^{j\omega})}. \tag{2.23}$$

Since the LP model assumes that the speech and noise are uncorrelated, all of the quantities in the expression are known and the Wiener filter for this problem is

$$H(e^{j\omega}) = \frac{A(e^{j\omega})}{A(e^{j\omega}) + \sigma_w^2}. \tag{2.24}$$

Given the frequency domain expression, the received signal can be filtered to produce the LLSE estimate of the clean speech based upon side information.

# Chapter 3

# Implementation and Results

The estimators described in Chapter 2 were implemented on sampled speech data. This chapter describes the approximations used to form practical noise cancellation algorithms and compares their output. First, techniques to capture the time variation of the speech signal are described. Next, the implementations of the ML and Wiener filter algorithms are presented. Finally, the comparisons of these algorithms are described

## 3.1 Short-Time Algorithm Implementation

The mathematical models used in Chapter 2 do not consider the time variation of the speech signal. The LP parameter model of speech works well on time scales where the acoustic characteristics of the vocal tract do not change significantly [12], [8]. A single set of parameters does not accurately describe long speech segments such as multiple-syllable words or sentences. If the LP side information available at the receiver is intended to describe such long speech segments, it will not provide accurate constraints on the estimates of the clean speech. Inaccurate side information descriptions violate the fundamental assumption of Chapter 2 and the estimation algorithms described there are likely to perform poorly.

The implementation of the algorithms used block processing to preserve the accuracy of the side information. The estimation algorithms were applied separately to short vectors of data that were well-modeled by their LP parameters. The individual vectors were selected by windowing the data. Consecutive vectors overlapped each other by half of the data window length. After the windowing was complete, the side information was calculated from the clean speech vectors. Next, the appropriate POCS algorithm estimated the clean speech vector from the corresponding noisy speech vector and side information. Once the POCS algorithm was complete, an estimate of the entire clean speech waveform was formed from the individual vector estimates. Because the clean speech vectors were generated by overlapped data windows, each sample in the clean speech waveform could be expressed as the sum of samples from consecutive clean speech vectors. Summing the corresponding samples from the estimated speech vectors generated the samples of the final speech estimate.

## 3.2 Data Windowing

The block processing approximation produces LP coefficient side information that accurately parameterizes the speech. Without careful design, the window used in the algorithm can introduce unnecessary distortion in the final speech estimate. Two mechanisms can produce errors in the estimate. First, the window can cause spectral smearing or leakage in the Fourier transform of the clean speech vectors. The distortions in the transform are consequences of the well-known windowing theorem of Fourier analysis [10]. Windows that have desirable frequency domain properties such as a narrow mainlobe and low sidelobes reduce these distortions and are assumed to lead to good estimates of the LP parameters [8].

A second source of distortion arises from the time domain properties of the window. As discussed in section 3.1, the final speech estimate is constructed by adding overlapping vectors of speech together. The overlap and add reconstruction of the speech can produce a modulation at the window frame rate. Even if the POCS algorithm estimates the individual windowed speech segments accurately, the estimate of the whole waveform will be noticeably distorted. Figure 3.1 shows the modulation introduced by a generic window. Distortions this severe are audible in the final speech

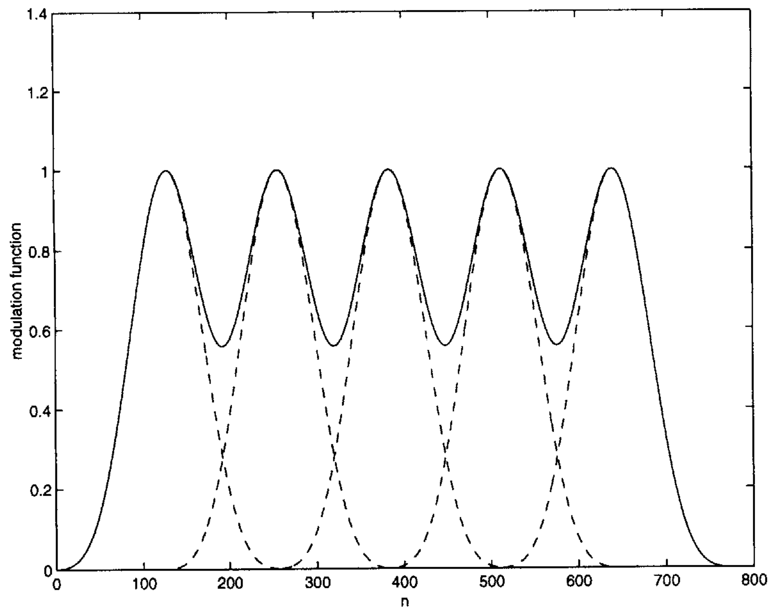


Figure 3-1: Windows can produce significant distortion

estimate. It is assumed that this distiortion can be avoided if the window selected sums to a constant sequence when overlapped and added.

### 3.2.1 Additive Reconstruction

Windows that sum to a constant sequence when overlapped and added are said to possess the additive reconstruction property. Windows that possess this characteristic must satisfy a set of linear constraints on their shape. Consider a finite length window, $w[n]$, that has $N$ non-zero samples. The window's region of support is $n = 0, \ldots, N-1$. It is shifted by increments of $k$ samples and then added together. If $w[n]$ possesses the additive reconstruction property, the window must satisfy

$$f[n] = \sum_{t=-\infty}^{\infty} w[n - tk] = 1 \;\forall n. \tag{3.1}$$

For any window, the sequence $f[n]$ is periodic. This can be seen because

$$f[n+k] = \sum_{t=-\infty}^{\infty} w[n+k-tk] = \sum_{t'=-\infty}^{\infty} w[n-t'k] = f[n].$$

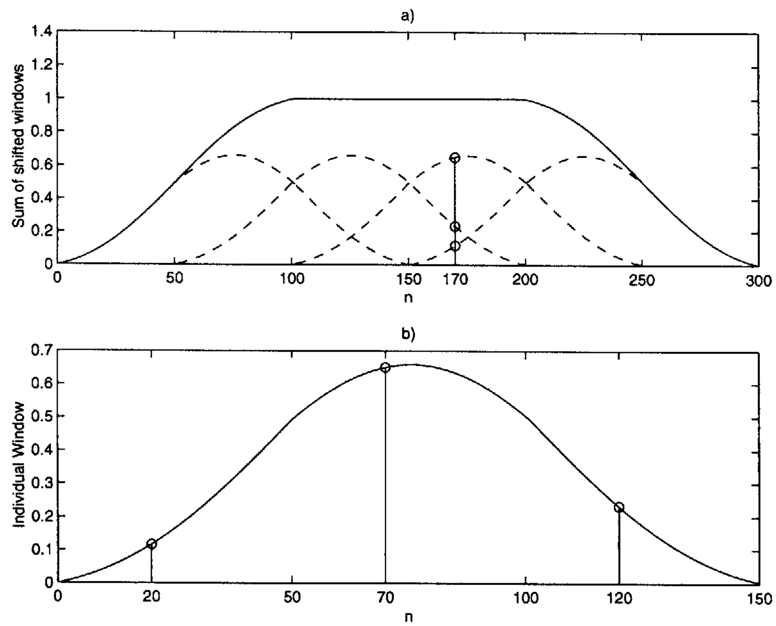


Figure 3-2: Additive reconstruction condition

Thus, to determine whether the window satisfies the condition for additive reconstruction, we only need to examine a single period of $f[n]$, say the period from $n = 0, \ldots, k-1$. If this period is constant, then the window possesses the additive reconstruction property.

The requirement for additive reconstruction yields a set of k linear equations. The constraints are derived by identifying which samples sum to form $f[0]$ to $f[k-1]$. Figure 3.2 shows that all of the samples of $w[n]$ that sum to a value of $f$ have the same remainder modulo $k$. For example, $f[0]$ is the sum of $w[0]$, $w[k]$, $w[2k]$, and so forth. The additive reconstruction condition is expressed by the equations

$$\sum_{a=0}^{r} w[p + ak] = 1 \text{ for } 0 \leq \mathrm{p} \leq \mathrm{k} - 1, \tag{3.2}$$

where $r$ is the greatest integer less or equal to than $\frac{N}{k}$. These equations are a discrete-time analog of the Nyquist criterion for avoiding intersymbol interference in a PAM communication system [4]. The conditions for additive reconstruction do not have a unique solution. Many common window functions, such as the raised cosine window, satisfy equation 3.2.

### 3.2.2 Window Design

The final window used in the block processing needed desirable frequency domain characteristics and the additive reconstruction property. The window was designed through a convolution procedure. A base window that met the frequency domain requirements of the processing algorithm was convolved with a square window. The resulting window met the same requirements in the frequency domain and had the additive reconstruction property.

Convolving a length $\frac{N}{2}+1$ sequence with a length $\frac{N}{2}$ square window produces a sequence that satisfies the additive reconstruction condition for shifts of $k=\frac{N}{2}$. If $a[n]$ is the base window and $s[n]$ is the square window, the resulting sequence is

$$\begin{aligned} w[n] &= \textstyle\sum_{k=0}^{\frac{N}{2}} a[k]s[n-k] \\ &= \begin{cases} \sum_{k=0}^{n} a[k] & 0 \leq n \leq \frac{N}{2}-1 \\ \sum_{k=n+1-\frac{N}{2}}^{\frac{N}{2}} a[k] & \frac{N}{2} \leq n \leq N-1 \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{3.3}$$

For a half-overlapped window, the additive reconstruction condition is

$$w[n] + w[n+\frac{N}{2}] = 1 \text{ for n} = 0,\ldots,\frac{\text{N}}{2}-1.$$

Using equation 3.3 the additive reconstruction sum is

$$\sum_{k=0}^{n} a[k] + \sum_{k=n+1}^{\frac{N}{2}} a[k] = \sum_{k=0}^{\frac{N}{2}} a[k]. \tag{3.4}$$

This is constant over all values $n=0,\ldots,\frac{N}{2}-1$, so the additive reconstruction property is satisfied.

Figure 3.3 illustrates the convolution result. The two windows cover the samples that sum to form $w[n]$ and $w[n+\frac{N}{2}]$. Shifting the windows does not change the overall sum. In all cases, the windows cover the complete sequence.

If $A(e^{j\omega})$ has desirable frequency domain characteristics, then $W(e^{j\omega})$ is likely

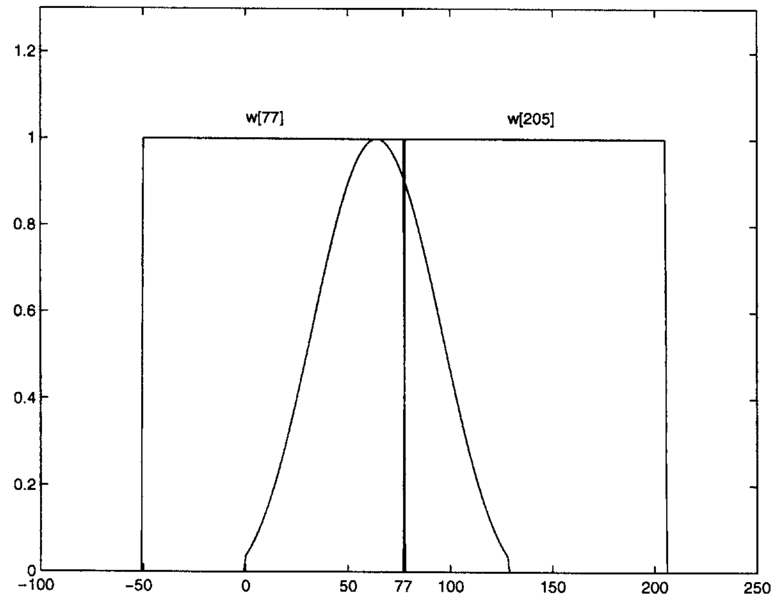


Figure 3-3: Design of window with additive reconstruction by convolution

to have desireable characteristics as well. The mainlobe of $S(e^{j\omega})$ is narrower than the mainlobe of most other windows. Usually, the mainlobe of $W(e^{j\omega})$ is narrower than the mainlobe of $A(e^{j\omega})$. The peaks of $W(e^{j\omega})$ at the same frequency as the first sidelobe of $A(e^{j\omega})$ are lower than the sidelobe in $A(e^{j\omega})$. The first sidelobe of $W(e^{j\omega})$ falls between the first zeros of $S(e^{j\omega})$ and $A(e^{j\omega})$. It is not guaranteed to be lower than the sidelobes of $A(e^{j\omega})$. In practice, however, it is usually not large enough to cause significant distortion through the windowing operation.

The convolution procedure was used to generate the final data window. A 129 sample Kaiser window with parameter $\beta = 5$ was convolved with a 128 sample square window. The resulting 256 sample window had the additive reconstruction property for shifts of $k = 128$ samples. Since the sample rate was 10 KHz, the window duration is 256 ms, and the time resolution was 128 ms. For someone speaking at a normal rate, the vocal tract behavior was approximately constant on this time scale, and the LP parameter side information described the signal well [12].

In addition to its time domain characteristics, the window possessed good behavior in the frequency domain. Its mainlobe width was $0.015\pi$ radians and its first sidelobes were -25 dB down . Despite the attempt to design the window with time and frequency domain properties specified independently, the resulting window had similar properties to the familiar raised cosine window. Both windows had nearly identical mainlobe widths, but the sidelobes of the raised cosine window were actually 3 dB lower than the window designed through convolution.

## 3.3 Side Information Computation

The estimation algorithms discussed in Chapter 2 used two types of side information. For each algorithm, the side information was calculated from the vectors of clean speech. The estimation algorithm discussed in section 2.2.2 and the Wiener filter discussed in section 2.3 used LP parameters as the side information. In the case of the ML algorithm, the final constraints on the estimate were expressed through an equivalent set of autocorrelation coefficients. Determining the LP parameters required computation unnecessary for the POCS algorithm, so only the autocorrelation sequences of the clean speech vectors were calculated. The Wiener filter, however, required a power spectrum estimate, so the LP parameters were calculated from the autocorrelation sequence.

The exact ML algorithm discussed in section 2.2.3 used zero-phase impulse response coefficients as side information. In practice, this sequence was calculated approximately. An exact determination of the coefficients required a continuous representation of the spectral magnitude of the clean speech vector. In the digital computer implementation, however, only a discrete representation was available. The magnitude was approximated by the clean speech vector's discrete Fourier transform (DFT) magnitude. The inverse discrete Fourier transform (IDFT) of this sequence produced the approximate side information.

The approximation for the zero-phase impulse response introduced errors into the side information. In general, zero-phase impulse responses are infinite length sequences, so finite-length approximations are distorted by time aliasing. To reduce the effects of this error, the side information was estimated with discrete transforms four times longer than the data window. The window was 256 samples long and

1024 sample transforms were used for each vector. It was judged that the additional reduction in error from a longer transform did not justify the extra computation required.

## 3.4 POCS Implementation

The discrete transform representation of the frequency domain also necessitated approximations in the POCS algorithms. The approximations counteracted errors in the implemenation of the projections. Again, the first source of error was time aliasing. The signals at each stage of the POCS algorithm were not restricted to be finite. The finite-length, discrete implementation of the algorithm could create time aliasing, distorting the final speech estimates. Additionally, the discrete frequency domain represenation prevented an exact implementation of the $P_2$ projection defined in equation 2.17. The exact $P_2$ operation set $U_{i+1}(e^{j\omega})$ to zero over any interval where $U_i(e^{j\omega})$ was negative. In discrete realizations of the projection, negative samples were set to zero. This implementation of $P_2$ satisfied the requirements of equation 2.17 at isolated values of $\omega$. Portions of the frequency axis between samples were not guaranteed to be positive.

To counteract the errors in the POCS algorithm, the sequences used in the iteration were 1024 samples long. Just as in the case of the zero-phase impulse response computation, this implementation reduced the impact of time aliasing. In the frequency domain, the long sequences reduced errors in the $P_2$ operation. The intervals between samples were smaller than those in 256 sample sequences, so deviations from the ideal $P_2$ projection were reduced.

The POCS procedure's convergence behavior was verified empirically. The algorithm required many iterations to converge to an estimate. Some experiments were performed to determine the appropriate number of iterations. Figure 3.4 shows the Euclidean distance between successive sequences after $P_1$ has been performed. The vertical axis measures the distance in a logarithmic scale. The horizontal axis measures the number of iterations. As the distance between successive sequences

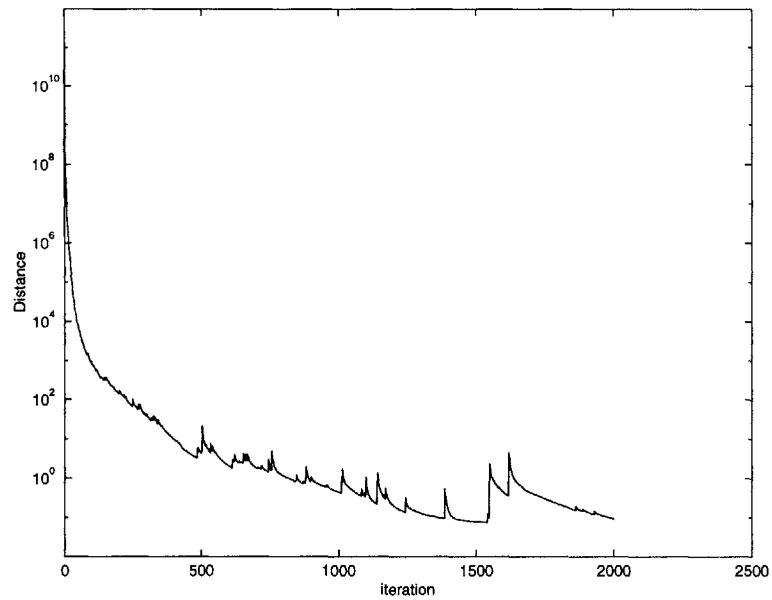


Figure 3-4: Convergence of POCS

decreases, the algorithm converges. After 1000 iterations, the distance between successive sequences was on the order of 100. In practice, the procedure was performed 100 times to generate the clean speech estimates.

The result of the POCS algorithm was used to estimate the spectral magnitude of the clean speech. The algorithm converged to an estimate of either the autocorrelation or zero-phase impulse response of the clean speech vector. As described in Chapter 2, these sequences are related to the spectral magnitude of the speech vector estimate by a Fourier transform. The DFT of the speech estimate was produced by combining the spectral magnitude estimate from the POCS algorithm with the phase of the noisy speech vector. The IDFT was taken and the first 256 samples were selected as the final estimate of the clean speech vector.

## 3.5 Wiener Filter Implementation

The approximate ML estimates using the LP parameters were compared with an LLSE estimate from the same side information. In Chapter 2, the non-causal Wiener filter was given in equation 2.23. The non-causal filter was used in this case because the data was transmitted and processed in vectors, and the noise cancellation algorithm had access to all of the data at once.

The Wiener filter was implemented as time-varying linear filter. Using the formula in equation 2.23, a 1024 point frequency sampled filter was calculated for each speech vector. The IDFTs of these sequences produced FIR filters approximating the Wiener filter for each vector. The time variation of the impulse response was obtained by linearly interpolating between the approximate filters. The interpolation avoided distortions present in signals filtered in the frequency domain caused by sudded jumps of the frequency response at frame boundaries.

## 3.6 Results

Three types of comparisons were performed between the noise reduction algorithms implemented in this thesis. First, the side channel noise reduction algorithms were compared with an iterative Wiener filtering algorithm described in Chapter 4. Next, the algorithms were compared as the amout of side information available was increased. Finally, the ML and Wiener filter algorithms were compared when the SNR was varied.

The data used to test the algorithms consisted of several sentences sampled at 10 KHz and 16 bits of precision. The data was processed on a workstation using floating point arithmetic. Male and female speakers were used in the experiments to test the algorithms on the frequency characteristics of different speakers.

The noise in these experiments was artificially generated and added after the signal was digitized. The noise was generated with the pseudorandom number generator provided with MATLAB and had approximately unit-variance, Gaussian statistics. The SNR was controlled by scaling the noise vector by an appropriate fraction of the signal power.

All of the comparisons were performed through informal listening. The goal of this research was to implement the side information noise reduction algorithms and qualitatively characterize their behavior. No attempt was made to perform formal trials and generate quantitative performance scores for these algorithms.

The first comparison measured the performance of the Wiener using side information against a single channel noise reduction algorithm. Out of a wide variety of options for a single channel algorithm, the iterative Wiener filter was chosen [6], [7]. This algorithm, as described in Chapter 4, estimates the LP parameters from the noisy speech signal and processes the speech with a Wiener filter based on these estimates. This algorithm is related to the Wiener filter described in section 2.3, and it was thought that the comparison between them would show the impact of accurate side information on noise reduction performance. Qualitatively, the output of both algorithms was similar at SNRs above 20 dB. At SNRs below 10 dB, however, the Wiener filter using side information was judged to have superior output in a casual comparison. As the SNR increased, the accuracy of the LP parameter estimates degraded and the performance of the iterative Wiener filter suffered relative to the Wiener filter with side information.

The second set of comparisons tracked the performance of the algorithms as the amount of side information was increased. Initial experiments were performed with twelve coefficients of side information per speech vector. These experiments simulated a side channel with a data rate one-tenth the rate of the noisy channel. For every 128 speech samples transmitted on the noisy channel, twelve coefficients of side information were transmitted on the side channel. This was seen as a reasonable ratio between the channel bandwidths for practical side information algorithms. Additionally, twelve LP parameters are viewed as an adequate number of coefficients to model speech on a short-time basis [12], [8].

As the amount of side information was increased, the performance of the ML estimation algorithm steadily improved. In casual listening, the presence of artifacts in the output was less noticeable and the similarities with the clean speech increased. As the number of side information coefficients was increased, larger portions of the autocorrelation sequence or zero-phase impulse response of the clean speech vector were specified and the constraints imposed by the side information became tighter. As more coefficients were transmitted, the errors in the signal came primarily from the phase estimate.

The Wiener filter displayed diminishing returns from additional side information. Appropriate LP models for speech usually use ten to twelve coefficients. Larger amounts of side information did not appear to improve the performance of the Wiener filter. For example, the differences between the output of filters using 12 and 64 LP coefficients were difficult to hear in casual listening. This behavior is reasonable considering the frequency domain interpretation of the Wiener filter. It attenuates frequences where the power spectrum of the noise is greater than the power spectrum of the signal and passes frequencies where the power spectrum of the signal is greater than the power spectrum of the noise. Thus, the frequency response of the Wiener filter is defined by the location and strength of the peaks in the power spectrum of the signal, corresponding to the formant frequencies of the speech. The formants can be accurately described by the tenth to twelfth order LP models initially used. The higher-order LP models do not change the locations of the peaks in the estimated power spectrum enough to affect the output of the Wiener filter.

The final set of experiments compared the performance of the POCS-based algorithms and the Wiener filter as the SNR was varied. The algorithms were tested with twelve coefficients of side information. The two estimation algorithms produced distinct artifacts when applied to the noisy speech data. The Wiener filter was a lowpass filter for most of the frames where speech energy was strong. Many of the speech vectors had the majority of their energy in the lower frequency bands and little energy in the high frequency bands. The Wiener filter passed low frequencies almost unattenuated, but reduced the high frequencies significantly. The system based upon the Wiener filter generally muffled the speech and the noise. It reduced the noise audibility and worked without noticeable distortion down to SNRs of 20dB. Below this level, the noise power was strong enough to make the low pass filtering muffle the speech and produce audible distortions in the signal.

The ML side information algorithms had a different effect on the signal. They were more successful at preserving the high-frequency content of the speech signal. The original speech was not as muffled in the POCS-based estimate as it was in the Wiener

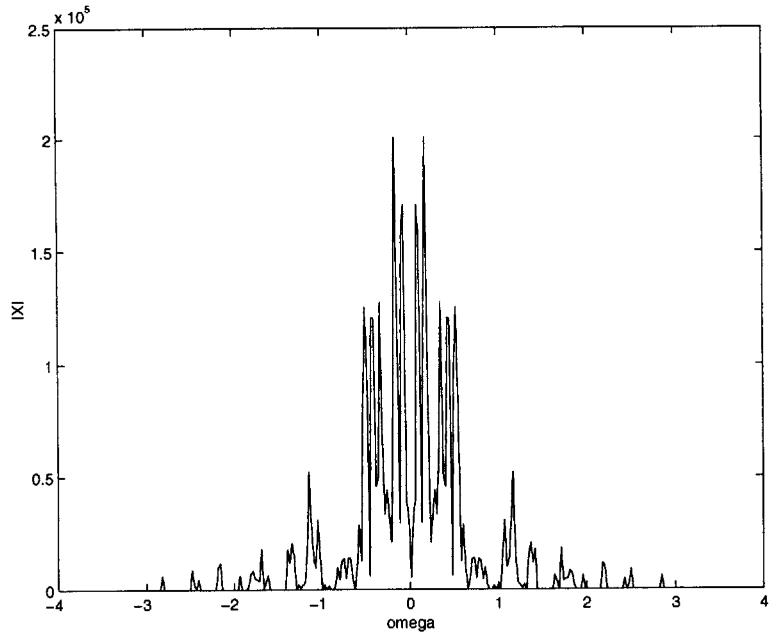


Figure 3-5: DFT Magnitude of a speech window

filter estimate. The POCS-based estimators, however, produced musical artifacts in the background at SNRs below 10 dB. The tones were generally more annoying to listeners than the muffled quality of the Wiener-filtered speech. The musical tones arose from the second projection in the POCS algorithms. The $P_2$ step often left sharp peaks in the frequency spectrum that were not zeroed. The peaks produced coherent tones which were noticeable to the human ear despite having low power. The frequencies of peaks shifted between frames, producing the chiming quality of the background interference.

An example of the sharp peaks in frequency is shown in figure 3.5 This figure shows one windowed segment of data after 100 iterations of the projection algorithm. The spectrum contains many sharp peaks in the high frequencies. The peaks do not correspond to frequencies where the clean speech possesses significant energy. They therefore sound like coherent tones in the background of the speech. The frequencies of the peaks change from frame to frame, producing the chiming artifacts heard in the output of the ML based algorithm.

# Chapter 4

# Single-Channel Enhancement Algorithms

As an auxiliary study, the POCS-based estimator was modified for use as a single channel enhancement algorithm. The algorithm was implemented with LP parameters estimated from the noisy speech instead of exact LP parameters transmitted on a side channel. This single channel enhancement algorithm was compared with iterative Wiener filtering.

## 4.1 Iterative Wiener Filtering

One of the standard problems in speech enhancement is estimating clean speech from a single channel transmitting only noisy data. A wide variety of approaches to this problem have been attempted [7]. The iterative Wiener filter was chosen for testing because it estimates the LP parameters of the speech in addition to estimating the speech waveform. The iterative Wiener filter is a special case of the Expectation-Maximization (EM) algorithm. The EM algorithm calculates an estimate of the underlying LP parameters of the speech from the observation of the noisy data [3], [9]. Generally, EM algorithms determine a set of parameters underlying a random process through two iterated steps. The first step produces an estimate of the unknown parameters using noise corrupted data. Once the parameters are known, a better estimate of the clean data can be made. From this data, a second estimate of the parameters is formed. This procedure is iterated, eventually converging to an estimate of the clean data and its parameters. It can be shown that the EM algorithm converges to a local maximum of the likelihood function relating the clean data and the underlying parameters [3].

The problem of single sensor speech enhancement can be mapped to an EM algorithm. In this mapping, the LP coefficients are the underlying parameters, the clean speech is the clean data, and the noisy speech is the corrupted data. An EM-based procedure estimates of the LP parameters and the clean speech. In the first step, the LP parameters for the noisy speech are generated. Using these estimates, an approximate power spectrum is generated and an approximate Wiener filter is calculated. The noisy speech is filtered and the steps are repeated. The next estimate of the LP parameters is calculated from the output of the Wiener filter. Another Wiener filter is calculated using these LP parameters. The noisy speech is filtered again, generating a second estimate of the clean speech. This iteration is repeated until some convergence condition is met.

## 4.2 Adaptation of the Side Information Algorithm

In the experiments described in Chapter 3, two competing algorithms were compared. The first algorithm used the exact LP side information to model the power spectrum of the speech and used a Wiener filter to generate an estimate. The second algorithm used the LP parameters in the POCS iteration. This comparison was repeated using the estimated LP parameters generated by the iterative Wiener filter.

The iterative Wiener filter described in section 4.1 produced estimates of the LP parameters and the speech. Based on the recommendations of previous work on the algorithm [6], the Wiener filter was iterated three times. In the first two iterations, the filtering was performed in the frequency domain. The final estimate of the speech was constructed by the time-varying Wiener filter described in section 3.5. The estimates of the LP parameters generated by the iterative Wiener filter were used in place of exact side information in the POCS algorithm. The algorithm was implemented in exactly the same fashion as the POCS algorithm described in section 3.4.

Both the iterated Wiener filtering algorithm and the side information algorithm were tested on speech at SNRs below 10 dB. The estimates from both algorithms sounded rough in casual listening. The output from the POCS-based algorithm, however, was thought to be less distorted than the estimate formed by the iterative Wiener filter. The added processing of the side information algorithm provided a modest improvement in the quality of the speech estimate. These results indicate that the POCS algorithm can produce speech estimates comparable to those from accepted single channel speech enhancement techniques.

# Chapter 5

# Conclusions

This thesis examines the noise cancellation problem when side information describing the clean speech is available. In certain situations, a reliable side channel between the signal source and receiver is open, but a cost constraint may prevent the channel from having adequate bandwidth to transmit the entire speech signal. Since the bandwidth of the side channel is limited, a compressed version of the clean speech is sent to the receiver. There, the noisy signal and the side information are combined to estimate the speech.

Chapter 2 reviews the framework for estimation with side information developed by Barron, *et. al.* Section 2.1 considers the MAP, MMSE, and ML estimators for a generic side information signal. Each of these estimators is calculated in a similar fashion to their counterparts without side information. The remainder of the chapter describes algorithms to calculate ML and LLSE estimators using LP parameters or zero-phase impulse response coefficients as side information.

Implementations of the ML and the LLSE estimators using the LP side information and the ML estimator using the closely related zero-phase impulse response side information were compared. The approximate ML estimates for both the LP and zero-phase side information were generated through a POCS procedure. The LLSE estimator was implemented by a time-varying filter approximating the Wiener filter.

Comparisons between the three algorithms showed that the ML estimators preserved the high-frequency content of the original speech but also contained chiming noises. The approximate ML estimate based upon the LP side information sounded less pleasing than the exact ML estimate using the zero phase impulse response side information. The chiming tones were more noticeable in the LP-based estimates of the speech. The LLSE estimate sounded muffled but lacked the chiming noises. Generally, listeners rated the LLSE estimate most pleasant because the distracting background tones were absent from the Wiener-filtered speech.

Additionally, the side-information procedure was adapted to use as a single channel speech enhancement procedure. The traditional iterative Wiener filtering approach, a form of the EM algorithm, was implemented to form an estimate of the LP parameters of the speech directly from the noisy data. The estimated LP parameters and the noisy speech were input into the POCS algorithm. The side information processing improved the intelligibility of the speech over the results of the iterated Wiener filter.

While the performance of the speech enhancement algorithms is hard to measure, this work has shown that side information can improve the quality of enhanced speech. Over a restricted range of SNRs, the estimation algorithms considered in this thesis produced output judged to be superior to that of a single channel iterative Wiener filter. While these algorithms have not been compared with a wide variety of existing single channel enhancement algorithms, the preliminary result indicates that side information can provide an advantage over certain single channel noise reduction systems. In situations where the disturbances in a speech signal prevent a speech processing system from achieving its performance goals and cost constrains the amount of hardware that can be deployed to reduce the noise, side information algorithms may be a feasible solution.

In future work, the side information approach can be applied to other signals besides speech, or the algorithm based upon projection onto convex sets can be improved. Procedures to mitigate the chiming artifacts produced by these algorithms can be investigated. Additionally, different choices of side information can be explored. A different set of side information may provide better constraints on the speech estimates and could lead to algorithms that have less annoying artifacts.

# Bibliography


[1] Richard Barron, Charles Sestok, and Alan V. Oppenheim. Speech enhancement using spectral magnitude side information. In *ICASSP Proceedings*, 1998.

[2] James P. Boyle and Richard L. Dykstra. A method for finding projections onto the intersections of convex sets in Hilbert space. In *Lecture Notes in Statistics*, number 37. Springer-Verlag, 1985.

[3] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statisical Society, Series B*, 39(1):1–38, 1977.

[4] Edward A. Lee and David G. Messerschmitt. *Digital Communication.* Kulwer Academic Publishers, 1994.

[5] Jae S. Lim, editor. *Speech Enhancement.* Prentice-Hall, 1983.

[6] Jae S. Lim and Alan V. Oppenheim. All-pole modeling of degraded speech. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-26(3):197–210, 1978.

[7] Jae S. Lim and Alan V. Oppenheim. Enhancement and bandwidth compression of noisy speech. *Proceedings of the IEEE*, 67:592–601, 1979.

[8] J. D. Markel and A. H. Gray, Jr. *Linear Prediction of Speech.* Springer-Verlag, 1976.

[9] Todd K. Moon. The Expectation-Maximization algorithm. *IEEE Signal Processing Magazine*, pages 47–60, November 1996.

[10] Alan V. Oppenheim and Ronald Schaefer. *Discrete Time Signal Processing.* Prentice-Hall, 1989.

[11] Athanasios Papoulis. *Probability, Random Variables, and Stochastic Processes.* McGraw-Hill, 1991.

[12] Lawrence Rabiner and Ronald Schaefer. *Digital Processing of Speech Signals.* Prentice-Hall, 1978.