ABSTRACT

PARKER, ALAN. Linear Predictive Coding with Multi-Pulse Excitation. (Under the direction of J.B. O'Neal, Jr.)

Linear Predictive Coding has been a widely used technique in speech processing. This dissertation analyzes a Linear Predictive Coding scheme with an excitation set consisting of multiple impulses. Equations are derived for the location and amplitude of the pulse excitations as well as the prediction coefficients used in the all-pole filter. The equations for the prediction coefficients are uncoupled from the input gain. The solution of the equations is guaranteed to minimize a total mean square error. Using these equations an iterative algorithm is presented which converges to a stationary point of the total error function. The algorithm is tested for data generated by a multi-pulse model and is able to detect all the underlying parameters of this model. Finally, the algorithm is used in synthetic speech production. The results indicate an improved performance over standard LPC.

LINEAR PREDICTIVE CODING

WITH

MULTI-PULSE EXCITATION

by

Alan Parker

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

RALEIGH

1 9 8 4

APPROVED BY:

_____ _____

_____ _____

_____

Chairman of Advisory Committee

BIOGRAPHY

Alan Parker was born in Slough Bucks, England, on November 30, 1959. He obtained his high school diploma from Joseph Wheeler High School in Marietta, Georgia in June 1977. He entered Georgia Institute of Technology and received a Bachelor of Science degree in Applied Mathematics in December 1980.

The author continued his studies at Georgia Institute of Technology obtaining a Master of Science degree in Electrical Engineering in June 1981. From June 1981 to December 1981 the author was employed as a Design Engineer for Scientific Atlanta. The author entered the doctoral program at North Carolina State University in January 1982. The author completed his doctoral program leading to the Degree of Doctor of Philosophy in Electrical Engineering in May 1984.

The author is married to Kate Ferriter. Ms. Ferriter graduated from Georgia Institute of Technology in December 1981 with a Bachelor of Science in Industrial Engineering.

TABLE OF CONTENTS

1 INTRODUCTION

1.1 Background

Production of speech at low bit rates has been of considerable interest in speech research. The bit rates required for different qualities of speech is summarized in Fig. 1.1 [1]. At the low end one of the most powerful techniques to date has been that of Linear Predictive Coding (LPC). It is capable of producing intelligible speech at bit rates of 4.8 kBits and below. It measures specific correlation parameters of a speech segment and uses these parameters in a model to synthesize the speech waveform. It is known that by increasing the bit rate for LPC that dramatic improvement of speech quality is not obtained. Atal [2] has suggested that the reason for this is the highly inflexible way present day LPC systems are excited. Presently, LPC systems use one impulse per pitch period to excite the filter. This was under the assumption that there existed only one excitation during each pitch period. Holmes [12] has demonstrated the possibility of more than one excitation during a pitch period; that is, before and after glottal closure. In doing so, Holmes has given the physical basis for a multi-pulse excitation LPC system. This implies that by increasing the bit rate (to describe more excitations and gains) an improved speech quality over standard LPC can be obtained while still maintaining low bit rates. Fig. 1.2 shows a multi-pulse model for LPC. The filter coefficients for multi-pulse are not the same as for standard LPC. These coefficients need to be derived based on the multi-pulse

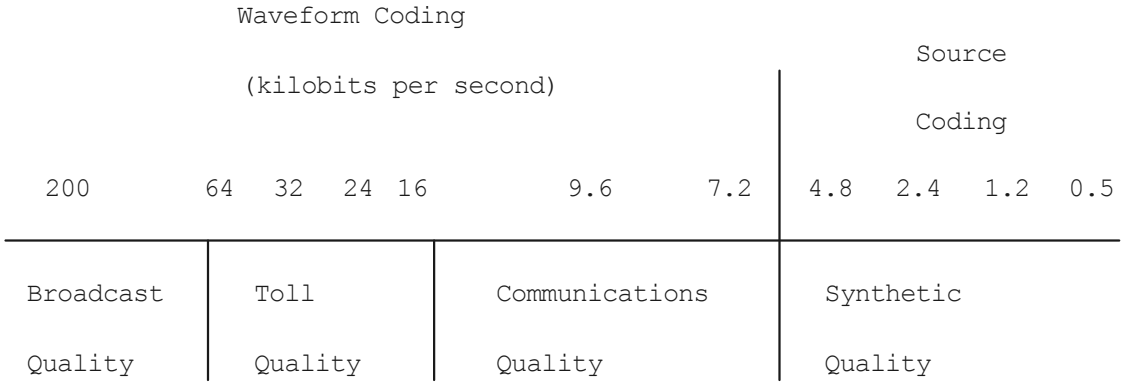

Figure 1.1 Bit Rate Required for Different Qualities of Speech

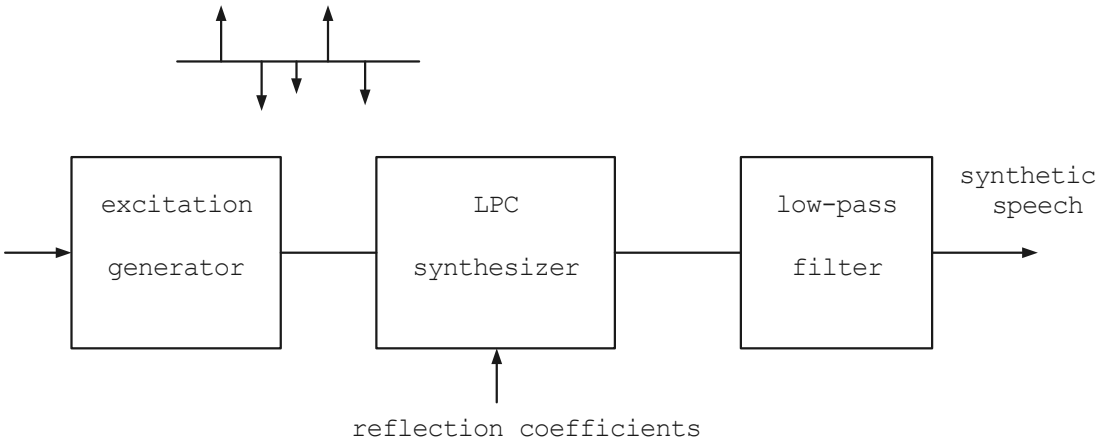

Figure 1.2 A Multi-Pulse Model for LPC

model and the minimization of some error. Singhal and Atal [15] derive a set of filter coefficients for the multi-pulse case; however, these coefficients are coupled with the input gain and are not guaranteed to yield a minimization of the mean-squared error. Atal and Singhal [15] have demonstrated that synthetic speech generated with a multi-pulse model results in a significantly improved speech quality over standard

LPC. Atal [2] derives the pulse positions using an analysis-by-synthesis method of sequential impulse extraction. In this research, an improved set of reflection coefficients is obtained which are guaranteed to minimize a mean-squared error. Also, a mathematical derivation for the optimal pulse locations is presented. In summary for multi-pulse the parameters to be determined are

- Pulse Positions
- Pulse Amplitudes
- Reflection Coefficients

Multi-Pulse will in general have the following properties

- A higher bit rate than LPC
- An improved quality over LPC
- No voice switch as in LPC
- No pitch detector as in LPC

1.2 Review of Research Methodology

The principles of LPC are reviewed in Chapter 2. Also discussed are the problems of synthetic speech production as they relate to multi-pulse and LPC.

In Chapter 3 multi-pulse LPC is discussed. Included is a new model for speech production along with the derivation of the pulse locations, amplitudes and reflection coefficients for this model. Chapter three is also concerned with frequency-weighted error.

Chapter 4 presents the multi-pulse algorithm and goes through an example of it in detail. Convergence of the algorithm is also discussed. In Chapter 5 simulations using the multi-pulse algorithm are performed. Multi-pulse data is generated and the algorithm is used to detect the parameters of the generating model. Also the robustness of the algorithm is tested as noise is added to both the input and output data.

Synthetic speech production is presented in Chapter 6. The LPC and multi-pulse predictors are compared when given real speech data. The bit rate for standard LPC and multi-pulse are derived. A new algorithm for transmitting the pulse positions is presented. This algorithm is compared in bit rate with the theoretically best algorithm. In addition the LPC distance measure is introduced as a comparison between multi-pulse and LPC.

Chapter 7 summarizes the results of the multi-pulse algorithm. Possible future applications of multi-pulse are discussed. Included in these are applications to seismic processing, speech recognition, and echo cancellation.

Presented in the appendix is the derivation of the reflection coefficients for the case where the pulse positions and amplitudes are known. The program written in the C programming language which was used to calculate the multi-pulse parameters is presented. Finally, the convergence of the algorithm for the general case is given.

2 LINEAR PREDICTIVE CODING

2.1 Introduction

Linear predictive coding has been a widely used technique in speech analysis and synthesis. Its power lies in its ability to provide accurate estimates of the speech parameters such as pitch, formants, and spectra. In the production of synthetic speech LPC introduces a mechanical quality or warble into the received speech. This may be due to the inaccurate way the LPC is excited. An idealized model for LPC is shown in Fig. 2.1. The excitation for LPC generally consists of white noise for unvoiced sections of speech and a quasi-impulse train for voiced sections of speech. The voiced-unvoiced switch is set into one position depending on the section of speech to be synthesized. An example of voiced and unvoiced speech is shown in Fig. 2.2. The reflection coefficients determine the behavior of the LPC synthesizer.

This chapter presents only those principles of LPC that will be used in comparison to multi-pulse. A complete discussion of LPC along with the assumptions involved in its derivation can be found in the references [3-11].

2.2 Principles of LPC

The filter representing all the effects of the vocal tract can be written as [8]

$$H(z) = \frac{G}{1 - \sum_{k=1}^{p} a_k z^{-k}} \tag{2.1}$$

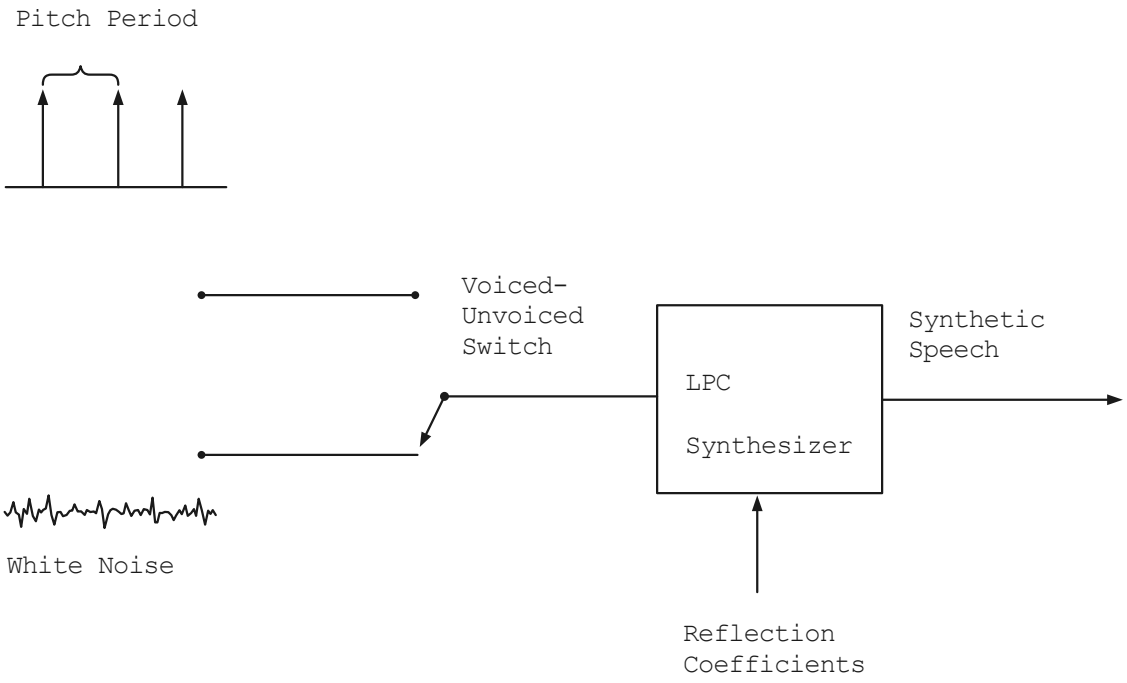

Figure 2.1 Model of LPC Speech Production

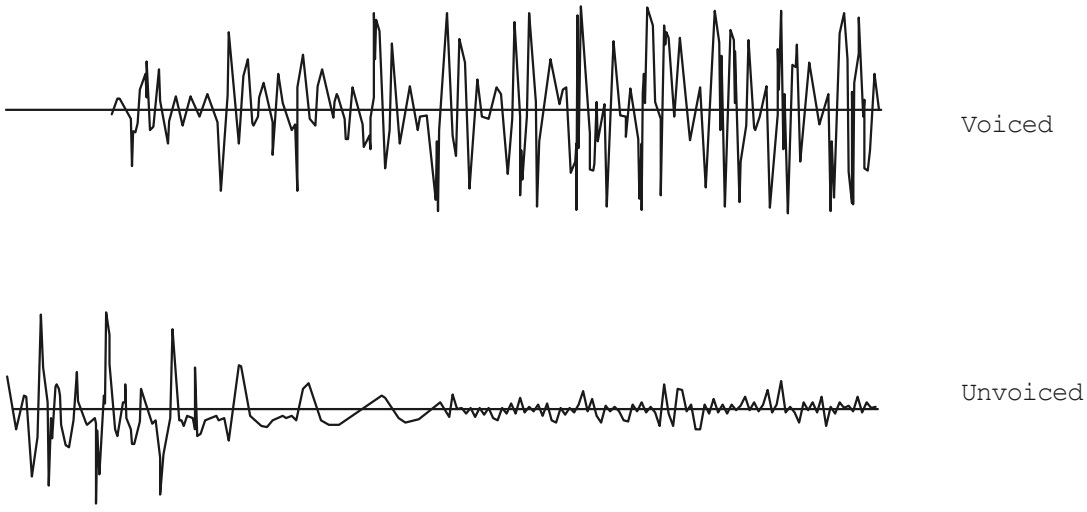

Figure 2.2 Example of Voiced and Unvoiced Speech

This yields the difference equation

$$s(n) = \sum_{k=1}^{p} a_k s(n-k) + Gu(n) \tag{2.2}$$

where $u(n)$ is the excitation. LPC assumes that during voiced sections of speech there is only one excitation per pitch period. If this is true then (2.2) can be written as

$$s(n) = \sum_{k=1}^{p} a_k s(n-k) + G\delta(n) \qquad 1 \leq n \leq N \tag{2.3}$$

where N is the block size. To obtain LPC parameters the estimate

$$\hat{s}(n) = \sum_{k=1}^{p} \alpha_k s(n-k) \tag{2.4}$$

is formed. The α_k's are chosen to minimize

$$E = \sum_{k=1}^{N} e^2(k) \tag{2.5}$$

with

$$e(n) = s(n) - \hat{s}(n) = s(n) - \sum_{k=1}^{p} \alpha_k s(n-k) \tag{2.6}$$

Taking

$$\frac{\partial E}{\partial \alpha_j} = 0 \tag{2.7}$$

and defining

$$\phi(i,k) = \sum_{m=1}^{N} s(m-i)s(m-k) \tag{2.8}$$

Standard matrix techniques produce [7]

$$\sum_{k=1}^{p} \alpha_k \phi(i,k) = \phi(i,0) \tag{2.9}$$

or in matrix notation

$$\Phi \vec{\alpha} = \psi \tag{2.10}$$

Φ is a $p \times p$ matrix containing elements $\phi(i,k)$ and ψ is a vector containing elements $\phi(i,0)$. It should be noted that equation (2.10) is a useful equation if there is only one excitation per pitch period and it can be accurately estimated. Otherwise, equation (2.4) will not be a good estimator. The solution of equation (2.10) can be done efficiently using the Cholesky Decomposition Technique [8] used in the next section. The procedure used in this section to determine the α_i's is known as the covariance method. The three basic methods of LPC analysis are

1. The covariance method
2. The autocorrelation method
3. The lattice method

These methods are thoroughly discussed in the literature [9,10,11].

2.3 Solution for the Covariance Method

To solve equation (2.10) one must use the fact that when Φ is invertible it is positive definite [8]. Then Φ can be written as

$$\Phi = ABA^T \tag{3.1}$$

with A a lower triangular matrix. It follows that

$$A_{ij}b_j = \phi(i,j) - \sum_{k=1}^{i-1} A_{ik}b_k A_{jk} \qquad 1 \leq j \leq i-1 \tag{3.2}$$

and

$$b_i = \phi(i,i) - \sum_{k=1}^{i-1} A_{ik}^2 b_k \qquad i \geq 2 \tag{3.3}$$

with initial condition

$$b_1 = \phi(1,1) \tag{3.4}$$

where ϕ is defined by equation (2.8). Once the b's and a's are obtained and noting that

$$\Phi\vec{\alpha} = ABA^T\vec{\alpha} = \Psi \tag{3.5}$$

letting

$$X = BA^T\vec{\alpha} \tag{3.6}$$

then

$$AX = \Psi \tag{3.7}$$

and

$$A^T\vec{\alpha} = B^{-1}X \tag{3.8}$$

From (3.7) one obtains

$$X_i = \psi_i - \sum_{j=1}^{i-1} A_{ij} X_j \qquad 2 \leq i \leq p \tag{3.9}$$

with initial condition

$$X_1 = \psi_1 \tag{3.10}$$

and from (3.8) one obtains

$$\alpha_i = X_i / B_i - \sum_{j=i+1}^{p} A_{ji} \alpha_j \qquad 1 \leq i \leq p-1 \tag{3.11}$$

with initial condition

$$\alpha_p = X_p / B_p \tag{3.12}$$

After solving equations (3.2), (3.3) and (3.4) for B and A one uses equations (3.9) and (3.10) to determine X and then the α's follow from equations (3.11) and (3.12).

2.4 Synthetic Speech Production using LPC

One possible use of LPC is in the production of synthetic speech. A model of LPC speech production is shown in Fig. 2.3

During voiced sections of speech the output of the filter is given as

$$\hat{s}(n) \; = \; \sum_{i=1}^{p} \alpha_i \hat{s}(n-i) \tag{4.1}$$

Hence, in synthetic speech production the error is

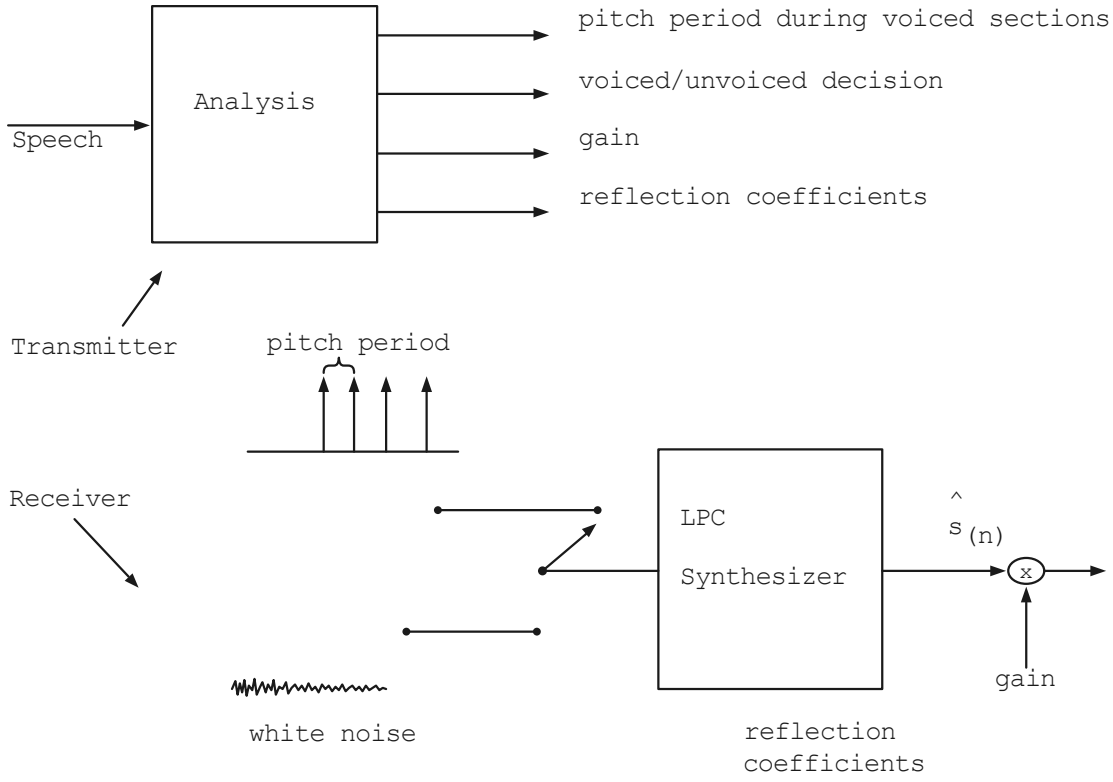

Figure 2.3 LPC Speech Production

$$e_s(n) \;=\; s(n) - \hat{s}(n) \;=\; s(n) - \sum_{i=1}^{p} \alpha_i \hat{s}(n-i) \tag{4.2}$$

One needs to recall that the α's are not chosen to minimize

$$E_s \;=\; \sum_{n=1}^{N} e_s^2(n) \tag{4.3}$$

but to minimize

$$E \;=\; \sum_{n=1}^{N} e^2(n) \tag{4.4}$$

with

$$e(n) \;=\; s(n) - \sum_{i=1}^{p} \alpha_i s(n-i) \tag{4.5}$$

So the α's are not chosen optimally for synthetic speech production. Minimization of equation (4.3) would result in the following equation for the α's

$$\sum_{n=1}^{N} s(n)\hat{s}(n-k) \;=\; \sum_{i=1}^{p}\sum_{n=1}^{N} \alpha_i \hat{s}(n-i)\hat{s}(n-k) \tag{4.6}$$

It is not clear that the α's can be determined from equation (4.6) since $\hat{s}(n)$ is not known and depends on the α's. This is one of the reasons that the error in equation (4.4) is minimized. So the α's are chosen to give the best (mean-square) prediction estimate

$$\hat{s}(n) \;=\; \sum_{i=1}^{p} \alpha_i s(n-i) \tag{4.7}$$

At the receiver these α's are used in

$$\hat{s}(n) \;=\; \sum_{i=1}^{p} \alpha_i \hat{s}(n-i) \tag{4.8}$$

Now

$$\hat{s}(n) = s(n) - e(n) \tag{4.9}$$

so at the receiver one gets

$$\hat{s}(n) \;=\; \sum_{i=1}^{p} \alpha_i [s(n-i) - e(n-i)] \tag{4.10}$$

which can be written as

$$\hat{s}(n) \;=\; \sum_{i=1}^{p} \alpha_i s(n-i) \;-\; \sum_{i=1}^{p} \alpha_i e(n-i) \tag{4.11}$$

so the synthetic speech is the prediction estimate (for which the α's are optimized) plus an autoregressive error sequence given by

$$e(n) \;=\; -\sum_{i=1}^{p} \alpha_i e(n-i) \tag{4.12}$$

Because of equation (4.12) it is important that the LPC parameters are chosen to be stable. If the α's do not result in a stable filter then a small initial error sequence could result in a very large error. That is, the impulse response can be a positive exponential for an unstable filter. The stability of the α's needs to be tested. The transfer function of the all-pole filter is

$$[H(z)]^{-1} = 1 - \sum_{i=1}^{p} \alpha_i z^{-i} \tag{4.13}$$

which can be written as

$$[H(z)]^{-1} = (1 - z^{-1}s_1)(1 - z^{-1}s_2)\cdots(1 - z^{-1}s_p) \tag{4.14}$$

where s_i is in general complex. For stability

$$|s_i| < 1 \qquad 1 \le i \le p \tag{4.15}$$

An efficient test for stability [10] in terms of the α's is as follows. Form the sequence k_i

$$
\begin{aligned}
k_i &= \alpha_i^{(i)} \\
\alpha_j^{(i-1)} &= \frac{\alpha_j^{(i)} + \alpha_i^{(i)}\alpha_{i-j}^{(i)}}{1 - k_i^2} \qquad 1 \le j \le i-1
\end{aligned}
\tag{4.16}
$$

where $i = p, p-1, \ldots, 1$. Initially, $\alpha_j^{(p)} = \alpha_j$. The test for stability is then given by

$$
|k_i| < 1 \qquad i = 1, \ldots, p \tag{4.17}
$$

Also, in practice, the α's should not be transmitted over the channel[8]. The reason for this is if the α's are a stable set then in many cases the quantized version $Q[\alpha]$ does not form a stable set. In other words, small fluctuations in the α's can result in large fluctuations in the pole locations. Generally, the k_i's are transmitted as defined by equation (4.16) since after quantization the k_i's will still result in a stable filter.

2.5 Conclusion

The technique of LPC has been presented in this chapter. The covariance equations were derived and an efficient means of solving them was presented. The next chapter deals with a new model for a multi-pulse excitation LPC.

3 DERIVATION OF MULTI-PULSE LPC

3.1 Introduction

As mentioned in the first chapter one of the reasons that present day LPC systems sound so mechanical may be because of the lack of a more sophisticated excitation set. One promising approach to the enhancement of low bit rate speech is reformulating the excitation in LPC as multiple impulses per block rather than one impulse per pitch period. Previous researchers have examined the multiple-pulse excitation problem by sequential impulse extraction [2,15]. That is, over a fixed block of speech the best single impulse excitation location (and amplitude) is found, then the best second impulse excitation is found conditioned on the first ,etc... until the final Nth excitation conditioned on the $N-1$ others is found. However, this method does not necessarily result in the optimal set of N impulses (i.e., that which produces the lowest mean square error between the actual and the modelled speech) since the introduction of succeeding excitations could alter the optimal location of the previously calculated impulses.

In this chapter a new method is presented for simultaneously extracting the optimal N impulse excitation locations and amplitudes from a given block of speech. This avoids the iterative recursions of the sequential approach previously considered. This method is similar to that of Jain [19] in that it uses the prediction residual to determine the pulse locations; however, in this work the pulse locations are chosen

to minimize a total mean-square error which, to the author's knowledge, is not the case in Jain's work. Another difference is in the solution for the prediction coefficients. Jain uses the prediction coefficients given by equation (9.1.7) while this work uses those in equation (4.8).

The method of multi-pulse used in this chapter is similar to pole-zero estimation. The multi-pulse algorithm estimates an all-pole filter and an input sequence (where zeros may or may not be present). The major difference between the two approaches is the assumption made on the input sequence. Typically, in pole-zero estimation [25,26,27,28] the input is assumed to be white noise or periodic impulses. In the multi-pulse case the input is assumed to be any sequence which is only non-zero at I points. The parameter I is discussed in later chapters. The next section presents the multi-pulse model.

3.2 The Multi-Pulse Model

Assume speech can be modelled as

$$s(n) = \sum_{i=1}^{p} a_i s(n-i) + x(n) + u(n) \qquad 1 \le n \le N \tag{2.1}$$

where $s(n)$ is the speech sample at time n, $x(n)$ is the excitation; a_i, $1 \le i \le p$ are the prediction coefficients and $u(n)$ is assumed to be white. The multi-pulse model will attempt to replicate this model and is given as

$$\hat{s}(n) = \sum_{i=1}^{p} \alpha_i s(n-i) + \hat{x}(n) \tag{2.2}$$

where $\hat{s}(n)$ is the approximation of $s(n)$; α_i, $1 \le i \le p$, are the estimates

of the a_i and $\hat{x}(n)$ is an estimate of $x(n)$. From this an error sequence is defined as the actual speech value minus the predicted speech sample

$$e(n) = s(n) - \hat{s}(n)$$

or

$$e(n) = s(n) - \sum_{i=1}^{p} \alpha_i s(n-i) - \hat{x}(n), \qquad 1 \leq n \leq N \tag{2.3}$$

The error to be minimized will be over the block of length N. The total error E will be defined as

$$E = \sum_{k=1}^{N} e^2(k) \tag{2.4}$$

In the multi-pulse model the input (excitation) will consist of I impulses over the block of length N. These impulses will have amplitude and positions given by

$$\hat{x}(n) = \sum_{i=1}^{I} b_i \delta(n - p_i) \qquad \begin{cases} 1 \leq n \leq N \\ 1 \leq p_i \leq N \end{cases} \tag{2.5}$$

where b_i is the amplitude of the ith impulse and p_i is its position. The discrete delta function is defined as

$$\delta(k) = \begin{cases} 1, & k = 0 \\ 0 & \text{otherwise} \end{cases}$$

The error sequence can then be written as

$$e(n) = s(n) - \sum_{i=1}^{p} \alpha_i s(n-i) - \sum_{i=1}^{I} b_i \delta(n - p_i) \tag{2.6}$$

3.3 Derivation of the Pulse Locations and Amplitudes

This section will give a mathematical derivation of the pulse positions. The error chosen to minimize was

$$E = \sum_{n=1}^{N} e^2(n) \tag{3.1}$$

with $e(n)$ given as

$$e(n) = s(n) - \sum_{i=1}^{p} \alpha_i s(n-i) - \sum_{j=1}^{I} b_j \delta(n-p_j) \tag{3.2}$$

The parameters α_i, b_j, and p_j are defined in the previous section. The prediction residual, $f(n)$, is defined as

$$f(n) = s(n) - \sum_{i=1}^{p} \alpha_i s(n-i) \tag{3.3}$$

For optimality,

$$\frac{\partial E}{\partial b_j} = 0 \tag{3.4}$$

This results in

$$\sum_{n=1}^{N} e(n) \frac{\partial e(n)}{\partial b_j} = 0 \tag{3.5}$$

Using equation (3.2) one obtains

$$\frac{\partial e(n)}{\partial b_j} = -\delta(n-p_j) \tag{3.6}$$

Equations (3.6) and (3.5) result in

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

Evaluating the summation yields

$$e(p_j) = 0 \tag{3.8}$$

From equations (3.3) and (3.2) it can be seen that

$$e(p_j) = f(p_j) - b_j \tag{3.9}$$

Hence, equation (3.8) becomes

$$b_j = f(p_j) \tag{3.10}$$

Using equations (3.1), (3.2), and (3.10) the total error E can be written as

$$E = \sum_{n=1}^{N} [\ f(n) - \sum_{j=1}^{I} f(p_j)\delta(n-p_j)\ ]^2 \tag{3.11}$$

If for the moment the reflection coefficients are fixed then the error defined by equation (3.11) can be thought of as a function of the pulse positions $p_1, p_2, \ldots, p_I$. If the pulse positions $p_1, \ldots, p_I$ actually corresponded to the optimal pulse locations then

$$E(p_1, p_2, \ldots, p_I) \leq E(s_1, s_2, \ldots, s_I) \tag{3.12}$$

where $s_1, s_2, \ldots, s_I$ correspond to any pulse positions. In particular, let

us move the pulse at position p_1 to a new position (where there is not a pulse already) $p_1 + \ell$. Then equation (3.12) would become

$$E(p_1, p_2, \ldots, p_I) \leq E(p_1 + \ell, p_2, \ldots, p_I) \tag{3.13}$$

with

$$p_j \neq p_1 + \ell \qquad j = 2, \ldots, I \tag{3.14}$$

Rewriting equation (3.13) using equation (3.11) one obtains

$$\begin{aligned} &\sum_{n=1}^{N} [\, f(n) - \sum_{j=2}^{I} f(p_j)\delta(n - p_j) \quad - f(p_1)\delta(n - p_1)\,]^2 \\ &\quad \leq \sum_{n=1}^{N} [\, f(n) - \sum_{j=2}^{I} f(p_j)\delta(n - p_j) \quad - f(p_1 + \ell)\delta(n - p_1 - \ell)\,]^2 \end{aligned} \tag{3.15}$$

defining

$$X(n) \;=\; f(n) - \sum_{j=2}^{I} f(p_j)\delta(n - p_j)$$

then equation (3.15) can be written as

$$\begin{aligned} &\sum_{n=1}^{N} [\, X(n) - f(p_1)\delta(n - p_1)\,]^2 \\ &\quad \leq \sum_{n=1}^{N} [\, X(n) - f(p_1 + \ell)\delta(n - p_1 - \ell)\,]^2 \end{aligned} \tag{3.16}$$

squaring both sides and noting that

$$X(p_1) = f(p_1)$$

and

$$X(p_1 + \ell) = f(p_1 + \ell)$$

one obtains

$$f^2(p_1) \geq f^2(p_1 + \ell)$$

which is

$$|f(p_1)| \geq |f(p_1 + \ell)| \tag{3.17}$$

Equation (3.17) is the equation for determining the pulse positions. If a pulse is at an optimal position p_1 and there is no pulse at position $p_1 + \ell$ then this equation must hold.

The function $f(n)$ determines the optimal locations and amplitudes of the pulses. The procedure is demonstrated as follows:

Graph the function $|f(n)|$ for $1 \leq n \leq N$. In this example four impulses will be used ($I = 4$). The block length will be eleven ($N = 11$). Fig. 3.1 shows the function that will be used in this example.

The first impulse is to be located at position 5 where $|f(n)|$ is a maximum. It will have $f(5) = \pm 6$. The second and third impulse are to be located at positions six and eight respectively where the function achieves its next largest value. The fourth impulse can be located at position 4, 7, 9 or 10 and the reduction in the error will be the same. For LPC the error is

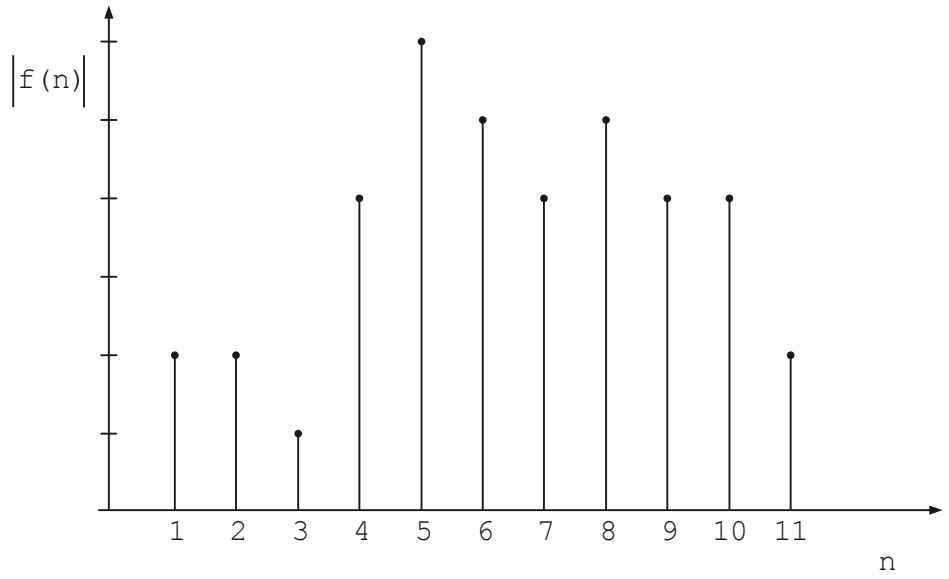

Figure 3.1 Function for calculating impulse locations

$$\sum_{k=1}^{N} e^2(k) = \sum_{k=1}^{11} f^2(k) = \sum_{k=1}^{11} |f(k)|^2 =$$

$$4 + 4 + 1 + 16 + 36 + 25 + 16 + 25 + 16 + 16 + 4 = 163$$

For multi-pulse the error is

$$\sum_{k=1}^{N} e^2(k) - \sum_{i=1}^{I} b_i^2 = 4 + 4 + 1 + 16 + 16 + 16 + 4 = 61$$

In conclusion, once you have obtained the function $|f(n)|$ the pulse locations and amplitudes follow from it. It is important to note that $|f(n)|$ is dependent on the α_i's. The above procedure is only valid once the α_i's are known. The procedure to determine the α_i's is discussed in the next section.

3.4 Derivation of the Prediction Coefficients for the Multi-Pulse Case

As in the derivation of the pulse locations and amplitudes the prediction coefficients will be chosen so that the total error (2.4) is minimized. To minimize E with respect to α_n we take the partial

$$\frac{\partial E}{\partial \alpha_n} = 0, \quad n = 1, \ldots, p$$

This results in

$$\sum_{k=1}^{N} \Big(s(k) - \sum_{i=1}^{p} \alpha_i s(k-i) - \sum_{j=1}^{I} b_j \delta(k-p_j) \Big) s(k-n) = 0 \qquad n = 1, \ldots, p$$

or

$$\sum_{k=1}^{N} s(k)s(k-n) - \sum_{k=1}^{N} \sum_{i=1}^{p} \alpha_i s(k-i)s(k-n) - \sum_{j=1}^{I} b_j s(p_j - n) = 0 \qquad (4.1)$$

where the δ function has been removed. Now by replacing b_j in (4.1) by its optimal value from the previous section one can write (4.1) as

$$\sum_{\substack{k=1 \\ k \neq p_1, \ldots, p_I}}^{N} s(k)s(k-n) = \sum_{\substack{k=1 \\ k \neq p_1, \ldots, p_I}}^{N} \sum_{i=1}^{p} \alpha_i s(k-i)s(k-n) \qquad n = 1, \ldots, p \quad (4.2)$$

Standard LPC yields

$$\sum_{k=1}^{N} s(k)s(k-n) = \sum_{k=1}^{N} \sum_{i=1}^{p} \alpha_i s(k-i)s(k-n) \qquad n = 1, \ldots, p \qquad (4.3)$$

which is the same as (4.2) without the restriction on the k's. Defining

$$\phi(i,k) = \sum_{n=1}^{N} s(n-i)s(n-k)$$

one can write (4.3) as

$$\phi(n,0) = \sum_{i=1}^{p} \alpha_i \phi(n,i) \qquad n = 1, \ldots, p \tag{4.4}$$

or in matrix notation $\Phi\vec{\alpha} = \Psi$ where Φ is positive semi-definite (all its eigenvalues are greater than or equal to zero). So when Φ is invertible one obtains

$$\vec{\alpha}_{LPC} = \Phi^{-1}\psi \tag{4.5}$$

Likewise for multi-pulse one can define

$$\Phi'(i,k) = \sum_{\substack{n=1 \\ n \neq p_1, \ldots p_I}}^{N} s(n-i)s(n-k) \tag{4.6}$$

and (4.2) can be written as

$$\Phi'(n,0) = \sum_{i=1}^{p} \alpha_i \Phi'(n,i) \qquad n = 1, \ldots, p \tag{4.7}$$

or in matrix notation $\Phi'\vec{\alpha} = \Psi'$ and for multi-pulse

$$\vec{\alpha}_{MP} = \Phi'^{-1}\Psi' \tag{4.8}$$

when Φ' is invertible. The next section will show that Φ' is positive semi-definite as Φ is in LPC.

3.5 The Multi-Pulse Matrix Φ'

To show that a matrix is positive semi-definite one must show its eigenvalues λ are greater than or equal to zero. Let λ be an eigenvalue of Φ' then there exists an $x \neq 0$ such that

$$\Phi' x = \lambda x$$

it follows that

$$x^T \Phi' x = x^T \lambda x = \lambda x^T x$$

or

$$\lambda = \frac{x^T \Phi' x}{x^T x}$$

hence

$$(\lambda \geq 0) \Leftrightarrow (x^T \Phi' x \geq 0)$$

It will be shown that $x^T \Phi' x \geq 0$ for all x hence all eigenvalues are non-negative. Let

$$x^T = [x_1, x_2, \ldots, x_p]$$

$$x^T \Phi' x = \sum_{i=1}^{p} \sum_{j=1}^{p} x_i x_j \Phi'(i,j)$$

$$x^T \Phi' x = \sum_{i=1}^{p} \sum_{j=1}^{p} x_i x_j \{ \sum_{\substack{m=1 \\ m \neq p_1, \ldots p_I}}^{N} s(m-i) s(m-j) \} \tag{5.1}$$

Consider

$$(x_1 s(m-1) + x_2 s(m-2) + \ldots + x_p s(m-p))^2 \geq 0$$

This can be written as

$$\sum_{i=1}^{p}\sum_{j=1}^{p} x_i x_j s(m-i)s(m-j) \geq 0 \tag{5.2}$$

and it follows immediately from this relationship that (5.1) is greater than or equal to zero. This proves that Φ' is positive semi-definite. This is important because it implies when Φ' is invertible the coefficients obtained by (4.8) correspond to a minimum of the error function for those pulse positions $p_1, \ldots, p_I$. This is demonstrated in 4.3. The next section derives the multi-pulse coefficients in a different manner.

3.6 Alternate Derivation of the Multi-Pulse Coefficients

Equation (4.2) will be derived using a different method which will give an interpretation of its meaning. Let us assume one is using the LPC process where the prediction estimate is

$$\hat{s}(n) = \sum_{i=1}^{p} \alpha_i s(n-i) \tag{6.1}$$

and the α_i's need to be chosen to minimize a particular error. For LPC the error is

$$E = \sum_{n=1}^{N} e^2(n) \tag{6.2}$$

with

$$e(n) = s(n) - \hat{s}(n) \tag{6.3}$$

Now for the moment let us assume that one is not concerned with the error at I particular positions and therefore defines an error to be minimized as

$$E_M = \sum_{\substack{n=1 \\ n \neq p_1, p_2, \ldots, p_I}}^{N} e^2(n) \tag{6.4}$$

with $e(n)$ as in equation (6.3) or

$$e(n) = s(n) - \sum_{i=1}^{p} \alpha_i s(n-i) \tag{6.5}$$

Taking the appropriate partial and setting it equal to zero one obtains

$$\sum_{\substack{k=1 \\ k \neq p_1, \ldots p_I}}^{N} s(k)s(k-n) = \sum_{\substack{k=1 \\ k \neq p_1, \ldots p_I}}^{N} \sum_{i=1}^{p} \alpha_i s(k-i)s(k-n) \qquad n = 1, \ldots, p \tag{6.6}$$

which is identical to equation (4.2). So the multi-pulse coefficients are the LPC coefficients where one is not concerned with how large the error function is at positions $p_1, p_2, \ldots, p_I$. These multi-pulse coefficients do not attempt to predict the speech at those positions. The error defined by equation (6.4) is a specific case of a time-domain weighted error which in general could be written as

$$E = \sum_{n=1}^{N} e^2(n)w(n) \tag{6.7}$$

From equation (6.7)

$$\frac{\partial E}{\partial \alpha_j} = \sum_{n=1}^{N} \frac{\partial}{\partial \alpha_j}[e^2(n)w(n)] \tag{6.8}$$

$$= \sum_{n=1}^{N} [\, 2e(n) \frac{\partial e(n)}{\partial \alpha_j} \cdot w(n) + e^2(n) \frac{\partial w(n)}{\partial \alpha_j} \,]$$

The window $w(n)$ will be a function of the positions only as given in (6.11). Throughout this dissertation, the pulse positions and prediction coefficients will be treated as independent variables; that is, when one is calculating the prediction coefficients the pulse positions will be fixed and when one is calculating the pulse positions the prediction coefficients will be fixed. Then, it follows that

$$\frac{\partial w(n)}{\partial \alpha_j} \; = \; 0 \tag{6.9}$$

Setting the partial in equation (6.8) to zero and using equation (6.9) one obtains

$$\sum_{k=1}^{N} s(k)s(k-n)w(k) \; = \; \sum_{k=1}^{N} \sum_{i=1}^{p} \alpha_i s(k-i)s(k-n)w(k) \qquad n = 1, \ldots, p \tag{6.10}$$

where for the multi-pulse the particular error weighting can be seen to be

$$w(n) \; = \; \begin{cases} 0 & n = p_1, p_2, \ldots, p_I \\ 1 & \text{otherwise} \end{cases} \tag{6.11}$$

In summary, multi-pulse can be seen to be equivalent to standard LPC where the error is weighted in a particular fashion in the time domain. The above statement only applies to the solution of the reflection

coefficients for multi-pulse. The next section is concerned with frequency-weighted mean square error.

3.7 Frequency-Weighted Error

Equation (4.2) for the reflection coefficients has been derived by minimizing a mean square error. Many applications call for the minimization of a frequency-weighted mean square error. The general multi-pulse equations will be derived for a particular class of frequency weighted filters. A block diagram of the standard multipulse set-up is shown in Fig. 3.2

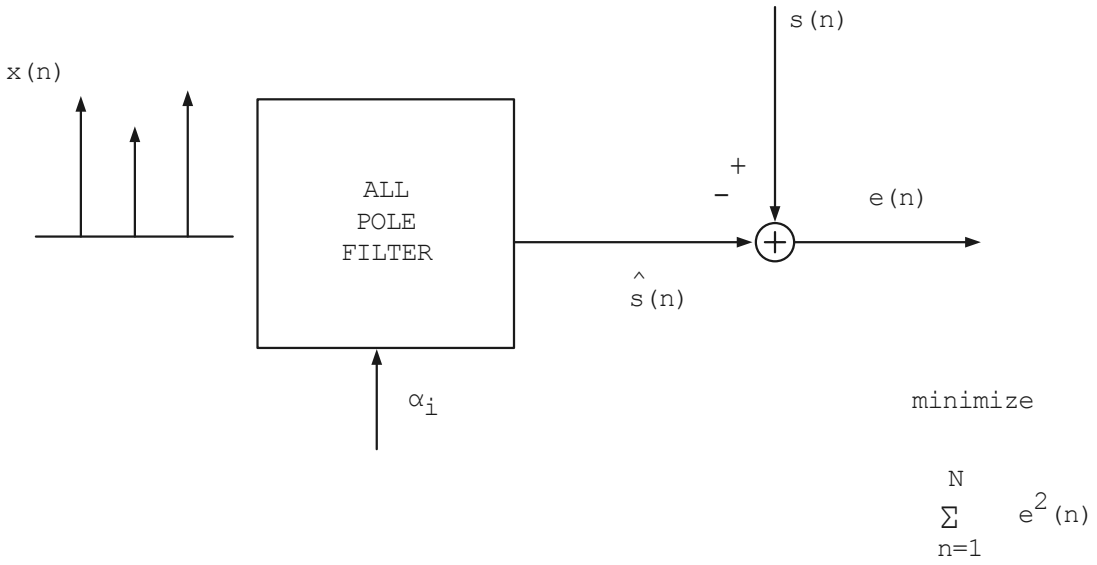

Figure 3.2 The Multi-Pulse Model

In Fig. 3.2 the error $e(n)$ is squared, summed and then minimized. If a frequency weighted filter is used then the set-up would be as shown in Fig. 3.3. In this figure $e_T(n)$ is the weighted error sequence. The new

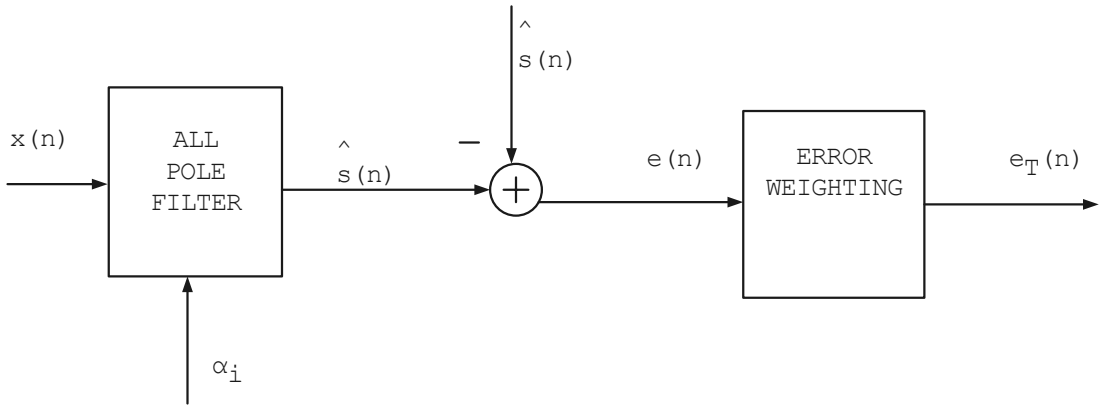

Figure 3.3 Multi-Pulse with Frequency-Weighted Error

problem is to minimize

$$E_T = \sum_{n=1}^{N} e_T^2(n) \tag{7.1}$$

where

$$e_T(n) = e(n) * h(n) \tag{7.2}$$

and h(n) is the impulse response of the frequency-weighted filter. Fig. 3.3 is equivalent to Fig. 3.4. Now, if the cascade of the all-pole filter and the frequency-weighted filter resulted in a new all-pole filter then Fig. 3.4 would become as shown in Fig. 3.5. The restriction that the combination of the LPC all-pole filter and the frequency-weighted filter result in an all-pole filter is not an excessive one. In fact, in the literature [2] one of the most frequent frequency-weighted filter used is

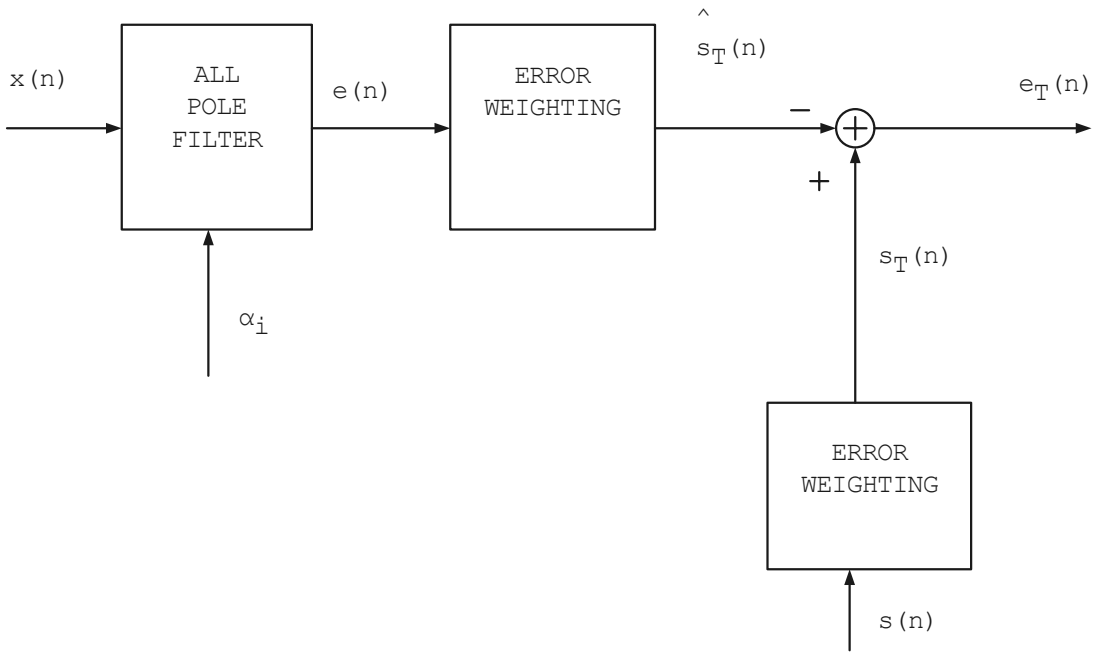

Figure 3.4 Alternate Form for Figure 3.3

$$W(z) = [\,1 - \sum_{k=1}^{p} \alpha_k z^{-k}\,]/[\,1 - \sum_{k=1}^{p} \alpha_k \gamma^k z^{-k}\,] \tag{7.3}$$

where γ is a number between zero and one. γ determines how one wants to de-emphasize the formant regions in the error spectrum. It can easily be seen from equation (7.3) that if this filter is used then the cascade will result in an all-pole filter. Now $e_T(n)$ can be minimized by using the multi-pulse equations on $s_T(n)$. After all, the multi-pulse algorithm will be able to determine the best impulsive excitation and all-pole filter to match a specific set of data.

3.8 Conclusion

This chapter develops a procedure for determining the optimal pulse locations and amplitudes for a given set of prediction coefficients. It avoids the previously used methods of sequential impulse extraction as it locates all positions and amplitudes globally once the prediction coefficients are obtained. In addition, for a given set of locations the equations for the prediction coefficients are given. The multi-pulse equations are put in a form which can be compared with the covariance equations of standard LPC. These results will be the basis of a proposed algorithm discussed in the next chapter. Also discussed was the solution to a minimization of a frequency-weighted mean square error for multi-pulse data. The next chapter presents an algorithm which locates the multi-pulse positions and determines the coefficients by the equations derived in this chapter.

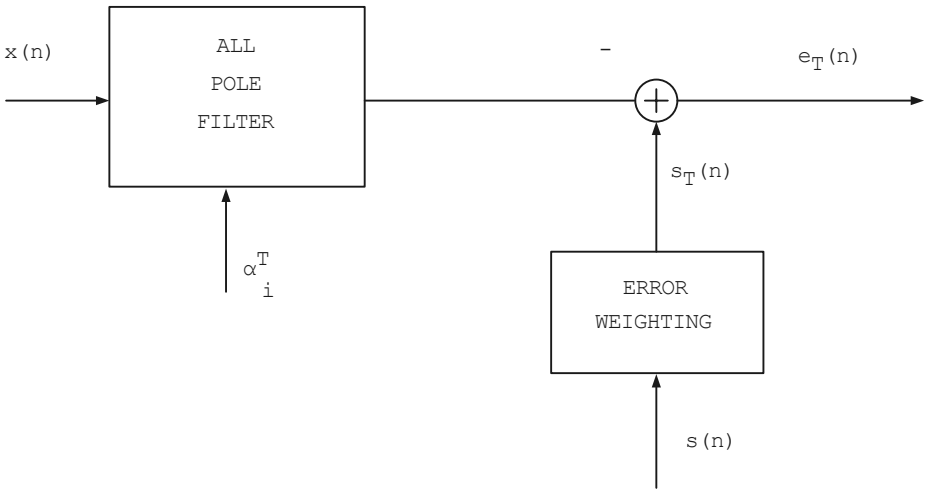

Figure 3.5 Frequency-Weighted Multi-Pulse Diagram

4 THE MULTI-PULSE ALGORITHM

4.1 Introduction

This chapter presents a multi-pulse algorithm using the results of the previous chapter. The algorithm is shown to monotonically decrease the error at each stage. Convergence is discussed and the algorithm is tested on true multi-pulse data.

4.2 The Multi-Pulse Algorithm

The algorithm for finding the reflection coefficients, pulse locations and pulse amplitudes is as follows:

1. Given the speech $s(n)$, $1 \leq n \leq N$ obtain α_{LPC} from (3.4.5).
2. Using these α_i's calculate the pulse positions by using the method discussed in 3.3.
3. Using these pulse positions calculate new α_i's using (3.4.8)
4. Continue? yes: go to 2

   no: stop.

The criterion for stopping at step 4 depends on the application. The next section demonstrates when the algorithm has converged.

4.3 Convergence of the Algorithm

This section defines convergence for the algorithm. Conditions on the speech segment are given which guarantee its convergence. Before convergence is defined a few preliminary definitions are introduced.

- $\vec{\alpha}_j = (\alpha_1, \ldots, \alpha_p)_j$ = the alphas at iteration j of the algorithm.
- $\vec{p}_j = (p_1, \ldots, p_I)_j$ = the positions at iteration j of the algorithm.
- $E(\vec{\alpha}_j, \vec{p}_j)$ = the total error at iteration j of the algorithm.

$$E(\vec{\alpha}_j, \vec{p}_j) = \sum_{\substack{n=1 \\ n \neq p_1, \ldots, p_I}}^{N} \left(s(n) - \sum_{i=1}^{p} \alpha_i s(n-i) \right)^2$$

where the positions and α's are from the vector $\vec{p}_j$ and $\vec{\alpha}_j$ respectively.

Iteration j is defined to mean that both step two and step three of the algorithm have taken place j times as shown in Fig. 4.1. Complete convergence of the algorithm is defined as follows

- The error sequence converges, i.e.

$$\lim_{j \to \infty} E(\vec{\alpha}_j, \vec{p}_j) = E_0.$$

- The α's converge

$$\lim_{j \to \infty} \vec{\alpha}_j = \vec{\alpha}_0$$

- The position vector converges

$$\lim_{j \to \infty} \vec{p}_j = \vec{p}_0$$

Some additional notation is necessary. As can be seen from Figure 4.1 at the beginning of each iteration the pulse positions are calculated based on the α's of the previous iteration. The notation $E(\vec{\alpha}_j, \vec{p}_{j+1})$ will be defined as the error after the first function of iteration j+1 is performed; that is, the error after the pulse positions are calculated but before the new α's are calculated based on those pulse positions. After iteration j the pulse positions $\vec{p}_j$ may not be the optimal pulse positions for $\vec{\alpha}_j$. The first function of iteration j+1 is to find the

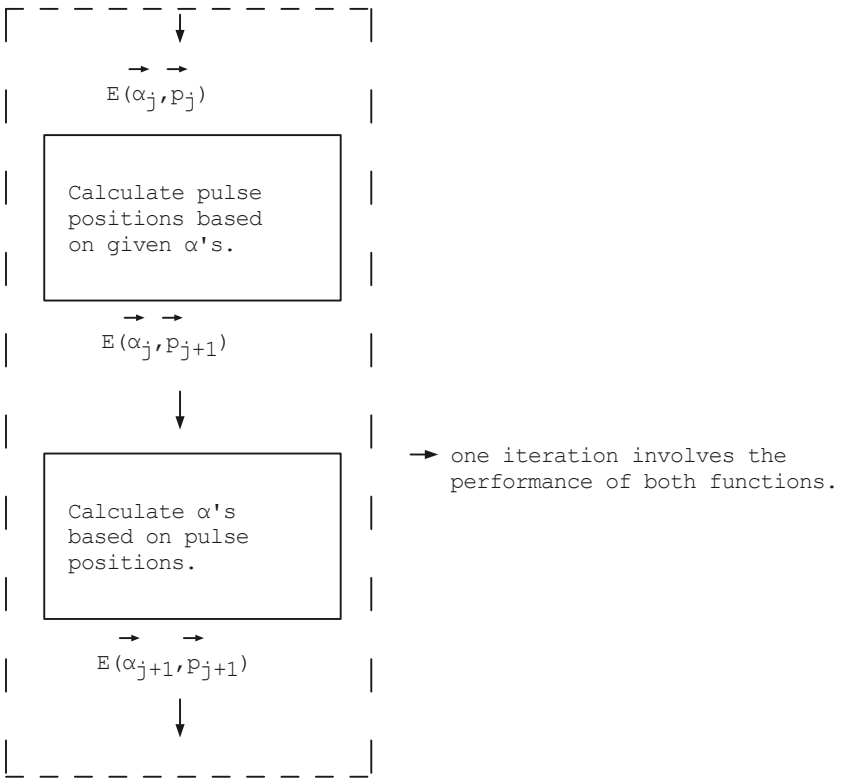

Figure 4.1 Definition of Iteration

optimal pulse positions for $\vec{\alpha}_j$. The method for doing this, by choosing the pulse locations to be those locations where the prediction residual has maximum magnitude, is an optimal one; that is, the algorithm finds a set of pulse locations for a fixed set of α's which minimizes the total error. The error then satisfies

$$E(\vec{\alpha}_j, \vec{p}_{j+1}) \leq E(\vec{\alpha}_j, \vec{p}_j) \tag{3.1}$$

because $\vec{p}_{j+1}$ contains the optimal pulse positions for the vector $\vec{\alpha}_j$.

The second function of each iteration is to calculate the α's based

on the given pulse positions. The α's satisfy

$$\Phi'\vec{\alpha} = \Psi' \tag{3.2}$$

where

$$\Phi' = \begin{bmatrix} \phi'(1,1) & \cdots & \phi'(1,p) \\ \vdots & & \vdots \\ \phi'(p,1) & \cdots & \phi'(p,p) \end{bmatrix}$$

with

$$\phi'(i,j) = \sum_{\substack{n=1 \\ n \neq p_1,\ldots,p_I}}^{N} s(n-i)s(n-j) \tag{3.3}$$

The matrix Φ' is positive semi-definite as proved in 3.3.5. That is, its eigenvalues are greater than or equal to zero. If it has a zero eigenvalue then Φ' is not invertible and there may be an infinite number of α's which satisfy equation (3.2). This is a highly undesirable situation since it is not clear which of these α's result in the minimization of the total error.

The assumption is made in this section that regardless of the iteration the matrix Φ' is invertible. In practice, this assumption has been displayed to be valid, especially when using quantized speech or speech in slightly noisy environments. Just as the covariance matrix in the solution of the normal equations for LPC is almost always invertible, Φ' in the multi-pulse case will almost always be invertible especially when the number of pulses removed is not large relative to the block size. In fact when I denotes the number of pulses per block and N denotes

the block size it has been found that for 600 different blocks of speech with an average of 5 iterations per block the Φ' matrix has always been invertible. The block size was N=160. The number of pulses were I=16 and I=32. So a total of 6000 Φ' matrices were calculated and every one of these were invertible. It is not required, however, that the matrix be always invertible. Section 9.3 presents a procedure which handles the case where Φ' has a zero eigenvalue. In this section we will deal with the case where Φ' is invertible.

If Φ' is invertible then the solution of equation (3.2) is unique and given by

$$\vec{\alpha} = \Phi'^{-1}\Psi' \tag{3.4}$$

Equation (3.2) yields a unique solution for the α's but we need to demonstrate that these α's result in a minimum of the total error. An appropriate way to do this is to look at the Jacobian matrix and use the second derivative test [21]. The Jacobian matrix is given as

$$J = \begin{bmatrix} \dfrac{\partial^2 E}{\partial\alpha_1\partial\alpha_1} & \cdots & \dfrac{\partial^2 E}{\partial\alpha_1\partial\alpha_p} \\ \vdots & & \vdots \\ \dfrac{\partial^2 E}{\partial\alpha_p\partial\alpha_1} & \cdots & \dfrac{\partial^2 E}{\partial\alpha_p\partial\alpha_p} \end{bmatrix}$$

These second order partials are evaluated at the critical point of E which is given already by equation (3.4). The second derivative test states that if these mixed partials are continuous and the matrix J is positive definite then the critical point is a relative strict minimum. In our case E is given as

$$E = \sum_{\substack{n=1 \\ n \neq p_1,\ldots,p_I}}^{N} e^2(n) \tag{3.5}$$

with

$$e(n) = s(n) - \sum_{m=1}^{p} \alpha_m s(n-m) \tag{3.6}$$

The first partial yields

$$\frac{\partial E}{\partial \alpha_i} = - \sum_{\substack{n=1 \\ n \neq p_1,\ldots,p_I}}^{N} 2e(n)s(n-i) \tag{3.7}$$

and the mixed partial is

$$\frac{\partial^2 E}{\partial \alpha_i \partial \alpha_j} \;=\; 2 \sum_{\substack{n=1 \\ n \neq p_1,\ldots,p_I}}^{N} s(n-i)s(n-j) \;=\; 2\Phi'(i,j) \tag{3.8}$$

The mixed partials are constants and therefore continuous. As can be seen from equation (3.8) the Jacobian matrix is given by

$$J \;=\; 2\Phi' \tag{3.9}$$

Since Φ' is positive definite then J is positive definite hence the alphas from equation (3.4) correspond to a strict relative minimum. Since there is only one stationary point the α's correspond to a global strict minimum. In conclusion, for a given set of pulse positions there exists a unique solution for the α's and this solution minimizes the total mean square error. The error sequence then satisfies

$$E(\vec{\alpha}_{j+1}, \vec{p}_{j+1}) \;\leq\; E(\vec{\alpha}_j, \vec{p}_{j+1}) \tag{3.10}$$

(3.10) along with (3.1) results in

$$E(\vec{\alpha}_{j+1}, \vec{p}_{j+1}) \leq E(\vec{\alpha}_j, \vec{p}_j) \tag{3.11}$$

The error sequence is a monotone decreasing sequence bounded below by zero and hence it converges. An important observation to be used later is

$$E(\vec{\alpha}_j, \vec{p}_{j+1}) > E(\vec{\alpha}_{j+1}, \vec{p}_{j+1}) \qquad \text{when } \vec{\alpha}_j \neq \vec{\alpha}_{j+1} \tag{3.12}$$

since the new alphas $\vec{\alpha}_{j+1}$ correspond to a strict minimum of the error function.

It is left to be proved that the α's converge and the position vector converges. Also, it is important to know when the algorithm has converged. The test for convergence of the algorithm is quite simple and will be presented first. The algorithm has converged at iteration $k+1$ when

$$\vec{p}_{k+1} = \vec{p}_k \tag{3.13}$$

that is, the positions at one iteration are identical to the positions at the next iteration. This is simple to see since $\vec{\alpha}_k$ was calculated based on $\vec{p}_k$. The alpha vector $\vec{\alpha}_{k+1}$ is calculated based on $\vec{p}_{k+1}$ and if $\vec{p}_{k+1}$ is equal to $\vec{p}_k$ then the alpha vector $\vec{\alpha}_{k+1}$ equals $\vec{\alpha}_k$. Likewise, $\vec{\alpha}_k$ was used to generate $\vec{p}_{k+1}$ and $\vec{\alpha}_{k+1}$ is used to generate $\vec{p}_{k+2}$. Since $\vec{\alpha}_k$ equals $\vec{\alpha}_{k+1}$ it follows that $\vec{p}_{k+2}$ equals $\vec{p}_{k+1}$. It follows that

$$\vec{p}_{k+j} = \vec{p}_k \qquad j = 1, 2, \ldots$$

and

$$\vec{\alpha}_{k+j} = \vec{\alpha}_k \qquad j = 1, 2, \ldots$$

What is happening is that the alphas are used to generate positions which in turn are used to generate the same alphas which in turn are used to generate the same positions ... Similarly an alternate test for convergence is

$$\vec{\alpha}_{k+1} = \vec{\alpha}_k \tag{3.14}$$

To prove that convergence occurs, one must show that equation (3.14) must hold for some k. Suppose

$$\vec{\alpha}_j \neq \vec{\alpha}_{j+1}$$

then from (3.12)

$$E(\vec{\alpha}_j, \vec{p}_{j+1}) > E(\vec{\alpha}_{j+1}, \vec{p}_{j+1}) \tag{3.15}$$

This implies that

$$\vec{p}_{j+m} \neq \vec{p}_j \quad m = 1, 2, \ldots \tag{3.16}$$

If there existed a k such that $\vec{p}_{j+k}$ equals $\vec{p}_j$ then it follows that

$$E(\vec{\alpha}_{j+k}, \vec{p}_{j+k}) = E(\vec{\alpha}_j, \vec{p}_j) \geq E(\vec{\alpha}_j, \vec{p}_{j+1}) > E(\vec{\alpha}_{j+1}, \vec{p}_{j+1})$$

But this is not possible since the error sequence is a monotone decreasing sequence; therefore, equation (3.16) is valid. The α's cannot change at every iteration because there are only a finite number of possible pulse positions. Each time the alphas change one pulse position will be excluded from the set of all future pulse positions. Hence the α's can change only a finite number of times. This implies that equation (3.14) must hold for some k. Once equation (3.14) holds the algorithm has completely converged. Section 9.3 discusses how to treat the case where the Φ' matrix has zero eigenvalues.

4.4 Rate of Convergence

Section 4.3 proves that the multi-pulse algorithm converges. The purpose of this section is to derive a meaningful bound on the time for its convergence. Implied in section 4.3 is that the maximum number of iterations for the algorithm is given by

$$\text{iter}_{\max} = \binom{N}{I} \tag{4.1}$$

where N is the block size and I is the number of pulses per block. A typical application might have N = 160 and I =16 for which

$$\text{iter}_{\max} = 4.06 \times 10^{21} \tag{4.2}$$

This bound is clearly not small enough to be of any practical use. This

section derives an expected time to converge, E_t, in terms of the number of pulses, I, and a probability, p, to be defined shortly. First it is useful to present some preliminary definitions.

From section 4.3 the algorithm converges at some iteration k to some position vector $\vec{p}_0$ given by

$$\vec{p}_0 = (v_1, v_2, \ldots, v_I) = \vec{p}_k \tag{4.3}$$

The algorithm is said to have "locked on" to a pulse position v_j at time ℓ if v_j is contained in each of the position vectors $\vec{p}_\ell, \vec{p}_{\ell+1}, \ldots, \vec{p}_k$. At time k, the time which the algorithm has converged, the algorithm has locked onto all pulse positions. Let p denote the probability that in going from one iteration to the next the algorithm locks onto at least one position not locked onto at the previous iteration. As an initial analysis, let us add the restriction that the algorithm can only lock onto one additional pulse position at each iteration. Let λ_j denote whether the algorithm has locked onto a pulse position at iteration j. That is,

$$\lambda_j = \begin{cases} 1 & \text{if the algorithm locks onto a new pulse at iteration } j \\ 0 & \text{otherwise} \end{cases}$$

The assumptions used in this section are

$$\Pr\{\lambda_j = 1\} = p \tag{4.4}$$

and

$$\Pr\{\lambda_j = 1 \mid \lambda_{j-1}, \lambda_{j-2}, \ldots, \lambda_1\} = \Pr\{\lambda_j = 1\} = p \tag{4.5}$$

Pr denotes the probability of the event occurring. Equation (4.4) implies that the ability of the algorithm to lock onto a pulse is independent of the iteration. Equation (4.5) implies that the ability of the algorithm to lock onto a pulse position at iteration j is independent of whether it locked onto a pulse position at iteration j-1,...,1. Equations (4.4) and (4.5) together imply that λ_j are independent Bernoulli trials. I pulses must be locked onto for the algorithm to converge. The number of iterations to obtain I successes in successive Bernoulli trials follows the Pascal distribution (also known as the negative binomial distribution) [23].

The probability that the algorithm has converged at step j is given by

$$\begin{aligned} p(j) &= \binom{j-1}{I-1} p^I (1-p)^{j-I} \quad j = I, I+1, \ldots \\ &= 0 \qquad\qquad\qquad\qquad \text{otherwise} \end{aligned} \tag{4.6}$$

The term $p^I(1-p)^{j-I}$ arises from the probability associated with exactly one outcome that has locked onto I pulses in j iterations. In order for this outcome to occur there must be I-1 pulses locked onto in j-1 iterations before the last iteration which locks onto the last pulse. The combination in equation (4.6) represents the number of arrangements satisfying this condition. The expected value μ of this process is [22]

$$\mu = \sum_{j=I}^{\infty} j \binom{j-1}{I-1} p^I (1-p)^{j-I} = I/p \tag{4.7}$$

Its variance σ^2 is given as

$$\sigma^2 = I(1-p)/p^2 \tag{4.8}$$

If the restriction that the algorithm can only lock onto one position at each iteration is removed then the expected value can only be reduced hence the time to convergence is upwardly bounded by μ. Thus the expected time of convergence satisfies

$$E_t \leq I/p \tag{4.9}$$

An assumption made in this section is that p does not change from iteration to iteration. It may be the case that the probability of locking into a pulse position at iteration one is greater than the probability of locking into a pulse position at iteration twelve. In the case of speech data this phenomenon was not observed. If p is close to one then the algorithm converges in an average of less than I iterations with a very low variance. The parameter p needs to be determined statistically. It may depend on the type of data given to the algorithm. Section 4.5 derives this probability parameter for speech data.

4.5 Statistical Analysis for the Convergence Rate

In this section an experiment is described which estimates p for speech data. First, speech is separated into m distinct blocks. For each block i the algorithm is performed. The number of iterations for the algorithm to converge on each block is k_i. After the algorithm has

converged on block i a parameter n_j^i is calculated as

$$n_j^i = \begin{cases} 1 & \text{if the algorithm locks onto a new pulse at iteration j} \\ 0 & \text{otherwise.} \end{cases}$$

An estimate X_i for block i is given as

$$X_i = \sum_{j=1}^{k_i} n_j^i \tag{5.1}$$

X_i is simply the number of successful times the algorithm locks onto a new pulse for block i. Finally, after all blocks are analyzed the sum X is formed as

$$X = \sum_{i=1}^{m} X_i \tag{5.2}$$

X can be written as

$$X = \sum_{i=1}^{m} \sum_{j=1}^{k_i} n_j^i \tag{5.3}$$

which can be written as

$$X = \sum_{i=1}^{n} \lambda_i \tag{5.4}$$

Each of the λ_i corresponds to a single iteration of the algorithm. The total number of iterations performed, n, is given by

$$n = \sum_{i=1}^{m} k_i \tag{5.5}$$

In equation (5.4) λ_i are assumed to be independent Bernoulli trials.

The independence means that the ability of the algorithm to lock onto a position at iteration k is independent of whether it locked onto a position at time k-1,...,1. This assumption was used in the previous section in determining the bound for the convergence of the algorithm. The probability density function of λ_i is

$$P(\lambda_i = 1) = p$$

$$P(\lambda_i = 0) = 1 - p$$

X is a sum of the random variables λ_i. If n is large then X is distributed normally with mean np and variance np(1-p) [23]. If a random variable Z is formed as

$$Z = (X - np)/[\{np(1-p)\}^{1/2}] \tag{5.6}$$

then $Z \sim N(0,1)$. The hypothesis will be tested

$$H_0 : p < p_0$$

$$H_1 : p > p_0$$

The acceptance and rejection regions are shown in Fig. 4.1. $Z_{\alpha/2}$ is the point which satisfies

$$P[\, Z > Z_{\alpha/2} \,] = \alpha/2 \tag{5.7}$$

This is known as the $100(1-\alpha)$ percent confidence estimate. If the experiment is performed M times then on the average $M(1-\alpha)$ of the

outcomes will lie in the acceptance region and $M\alpha$ outcomes will lie in the rejection region. In the experiment on speech n was 3000, X given by equation (5.4) was measured also to be 3000; that is, on every iteration the algorithm picked up at least one pulse. So the measured Z is

$$Z = [n(1-p)/p]^{1/2} \tag{5.8}$$

To solve for those p_0's which would be rejected one has

$$[n(1-p_0)/p_0]^{1/2} \; > \; Z_{\alpha/2} \tag{5.9}$$

which can be written as

$$p_0 \; < \; n/[\, n + (Z_{\alpha/2})^2 \,] \tag{5.10}$$

If $Z_{\alpha/2} = 7.0$ then $\alpha = 10^{-12}$. The rejected probabilities would then be

$$p_0 \; < \; 3000/3049 \; = \; .9839 \tag{5.11}$$

If the true value of p was less than .9839 then it follows that if this experiment were done 10^{12} times then on the average only one would satisfy $Z > 7.0$. So statistically one can say with $100(1 - 10^{-12})$ percent confidence that p is greater than .9839. If this value of p is put into the convergence bound then

$$E_t < 1.0163 \; I \tag{5.12}$$

The data sets used to measure p included voiced, unvoiced, and silence

portions of speech. For speech production the bound (5.12) should be sufficient. The algorithm was run for different values of I and N. When I was equal to five the number of iterations often required was close to five; however, when I was large the number of iterations was not as large as the value I. When I was 16 the algorithm typically converged within 8 iterations. The probability p did not change as a result of the number of pulses used. This probability was measured when I was 5,10,15,20, and 25. The value of N was 160.

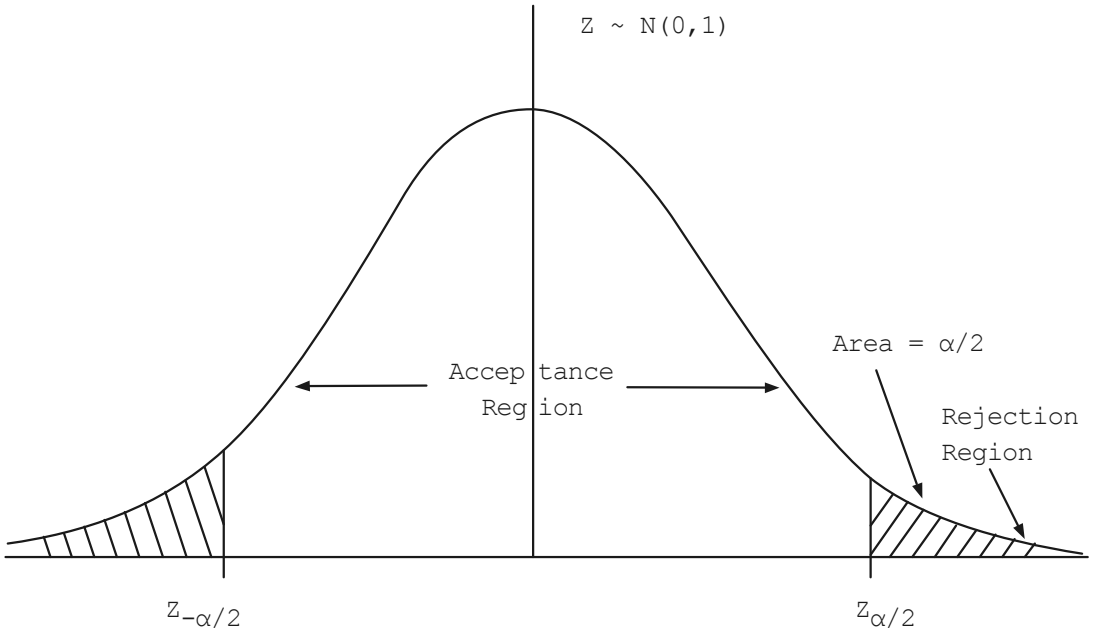

Figure 4.1 Acceptance and Rejection Regions

4.6 Perturbation of the Prediction Coefficients as a Result of Noise

In this section a bound is derived for the maximum change in the prediction coefficients due to the addition of white noise to the

original speech. This is the same scenario as in section 9.3. The equations for the solution of the prediction coefficients are given by

$$\Phi'\vec{\alpha} = \Psi' \tag{6.1}$$

If white noise is added to the speech segment then the equation results in

$$\Phi'_n\vec{\alpha}_n = \Psi'_n \tag{6.2}$$

The subscript n will denote that the speech segment is the noisy one. It follows from equations (9.3.8) and (9.3.9) that

$$\Phi'_n = \Phi' + \sigma I \tag{6.3}$$

and

$$\Psi'_n = \Psi' \tag{6.4}$$

In equation (6.3) σ is the variance of the white noise added to the speech segment. It follows from equations (6.4),(6.2), and (6.1) that

$$\begin{aligned} \vec{\alpha} - \vec{\alpha}_n &= \vec{\alpha} - (\Phi'_n)^{-1}\Psi' \\ &= \vec{\alpha} - (\Phi'_n)^{-1}\Phi'\vec{\alpha} \\ &= (I - (\Phi'_n)^{-1}\Phi')\vec{\alpha} \\ &= (\Phi'_n)^{-1}[\Phi'_n - \Phi']\vec{\alpha} \end{aligned} \tag{6.5}$$

From equation (6.3)

$$\Phi'_n - \Phi' = \sigma I \tag{6.6}$$

By using this in equation (6.5) one obtains

$$\vec{\alpha} - \vec{\alpha}_n = \sigma(\Phi'_n)^{-1}\vec{\alpha} \tag{6.7}$$

Equation (6.7) yields

$$\|\vec{\alpha} - \vec{\alpha}_n\| \leq \sigma\|(\Phi'_n)^{-1}\| \, \|\vec{\alpha}\| \tag{6.8}$$

The norm $\| \ \|$ is the ℓ_2 norm defined for a vector as

$$\|(x_1, x_2, \ldots, x_n)\| = \left[\sum_{i=1}^{n} x_i^2\right]^{1/2} \tag{6.9}$$

It is useful to use three theorems of functional analysis [24] which are

- Every linear operator defined on a finite-dimensional normed linear space is compact.
- If Φ is a compact, symmetric operator on a non-trivial Hilbert space H then $\|\Phi\| = |\lambda_{max}|$ where λ_{max} is the eigenvalue of Φ with largest absolute magnitude.
- If λ is an eigenvalue of an invertible matrix Φ then $1/\lambda$ is an eigenvalue of Φ^{-1}.

It follows from these three theorems that

$$\|(\Phi'_n)^{-1}\| = 1/\lambda_{min} \tag{6.10}$$

where λ_{min} is the minimum eigenvalue of Φ'_n. Putting (6.10) into (6.8) one obtains

$$\|\vec{\alpha} - \vec{\alpha}_n\| \leq (\sigma/\lambda_{min})\|\vec{\alpha}\| \tag{6.11}$$

From equation (6.11) it follows

$$\|\vec{\alpha} - \vec{\alpha}_n\| \leq [\sigma/(\lambda_{min} - \sigma)]\|\vec{\alpha}\| \tag{6.12}$$

but $\lambda_{min} - \sigma$ is the minimum eigenvalue of Φ'. Finally, the following result is obtained

$$\|\vec{\alpha} - \vec{\alpha}_n\| \leq (\sigma/\lambda_{min})\|\vec{\alpha}\| \tag{6.13}$$

where λ_{min} is the minimum eigenvalue of Φ'. This result shows that small perturbations in the signal result in small perturbations in the alphas hence the algorithm is robust in the selection of the alphas.

4.7 Perturbation of the Total Error as a Result of Noise

In this section a bound is derived for the maximum change in the total error due to the addition of white noise. In the covariance method for LPC Rabiner and Schafer [8] show that the total error can be written as

$$E = \phi(0,0) - \vec{\alpha}^T\Psi \tag{7.1}$$

A similar expression for multi-pulse for the total error is

$$E = \phi'(0,0) - \vec{\alpha}^T\Psi' \tag{7.2}$$

The derivation of equation (7.2) is precisely the same as the derivation of (7.1) except that the summations have the restrictions on the pulse positions. If white noise with variance σ were added to the speech then equation (7.2) would become

$$E_n = \phi'(0,0) + \sigma - \vec{\alpha}_n^T \Psi' \tag{7.3}$$

It follows that

$$E - E_n = \sigma - (\vec{\alpha}^T - \vec{\alpha}_n^T)\Psi' \tag{7.4}$$

From this equation one obtains

$$|E - E_n| \leq \sigma + \|\vec{\alpha}^T - \vec{\alpha}_n^T\| \|\Psi'\| \tag{7.5}$$

The norm $\|\|$ is the ℓ_2 norm as defined in equation (6.9). Noting that

$$\|\vec{\alpha}^T - \vec{\alpha}_n^T\| = \|(\vec{\alpha} - \vec{\alpha}_n)^T\| = \|(\vec{\alpha} - \vec{\alpha}_n)\| \tag{7.6}$$

and from (6.13)

$$\|\vec{\alpha} - \vec{\alpha}_n\| \leq (\sigma/\lambda_{\min})\|\vec{\alpha}\| \tag{7.7}$$

Also,

$$\|\Psi'\| = \|\Phi'\vec{\alpha}\| \leq \lambda_{\max}\|\vec{\alpha}\| \tag{7.8}$$

Putting (7.7) and (7.8) into (7.5) yields

$$|E - E_n| \leq \sigma + \sigma(\lambda_{\max}/\lambda_{\min})\|\vec{\alpha}\|^2 \tag{7.9}$$

which can be written as

$$|E - E_n| \leq \sigma[1 + (\lambda_{\max}/\lambda_{\min})\|\vec{\alpha}\|^2] \tag{7.9}$$

This result shows that small perturbations in the signal result in small perturbations in the total error at each stage.

4.8 Testing the Algorithm

In order to test the algorithm, true multi-pulse data will be given to it and it will attempt to determine the prediction coefficients, the pulse amplitudes and the pulse locations. The data is generated as follows

$$s(n) = \sum_{i=1}^{6} a_i s(n-i) + x(n) \qquad 1 \leq n \leq 50 \tag{8.1}$$

Table 4.1 gives the values of the a_i's used for the generation.

Table 4.2 gives the values of the input sequence $x(n)$. If not specified in Table 4.2, $x(n)$ is assumed to be zero. The speech is shown in Table 4.3. The speech was arbitrarily chosen to be zero for the first six samples. As can be seen from equation (8.1) some initial data is required.

Table 4.1 Prediction Coefficients for Multi-Pulse Example

n	Prediction coefficient a_n
1	.85776
2	-.40311
3	-.35678
4	.35421
5	-.31809
6	-.23467

Table 4.2 Pulse Locations and Amplitudes for Multi-Pulse Example

n	Input $x(n)$	n	Input $x(n)$	n	Input $x(n)$
9	10.0	19	14.0	37	29.0
10	-10.0	20	-18.0	44	-20.0
11	23.0	21	34.0	45	28.0
14	19.0	23	142.0		
15	17.0	25	98.0		

When the algorithm is given the multi-pulse data from Table 4.3 it converges in three full iterations. The error at stage four of the algorithm is shown in Table 4.4 for each iteration.

The pulse positions at each iteration are shown in Table 4.5. In that table they are arranged in decreasing order of pulse magnitude. It can be seen from Table 4.2 that the positions obtained at iteration three of the algorithm are in the correct order. The algorithm, therefore, can detect a true multi-pulse model. At iteration three of the algorithm the pulse positions, pulse amplitudes, and prediction coefficients are exactly those that are given in Table 4.1 and Table 4.2. The algorithm

Table 4.3 Generated Multi-Pulse Data

n	data $s(n)$	n	data $s(n)$
1	0.000000	26	21.42594
2	0.000000	27	-38.25678
3	0.000000	28	-103.9352
4	0.000000	29	-106.5658
5	0.000000	30	-104.6697
6	0.000000	31	-64.73758
7	0.000000	32	-4.989212
8	0.000000	33	63.45269
9	10.00000	34	100.7484
10	-1.422400	35	97.99105
11	17.74882	36	64.18945
12	12.22981	37	47.86752
13	7.385102	38	-3.104567
14	10.38753	39	-57.08845
15	22.96221	40	-96.87094
16	11.89389	41	-85.42988
17	-8.199680	42	-45.24986
18	-21.56009	43	-.2811050
19	-2.335299	44	13.05438
20	-13.91531	45	69.40560
21	18.62132	46	88.25043
22	14.59551	47	77.40385
23	159.9328	48	21.38895
24	125.5301	49	-23.84401
25	147.5670	50	-50.57220

Table 4.4 Multi-Pulse Algorithm Error at each Iteration

Iteration	Error
1	2391.174
2	63.09375
3	0.000000

has been tested on many true models and has succeeded in detecting all the parameters correctly. In practice the algorithm has always converged in less than seven iterations. This includes those tests which were run on real speech data in the presence of noise.

Table 4.5 Pulse Positions for the Multi-Pulse Algorithm

iteration	pulse locations (largest pulse amplitudes first)
1	23 37 45 44 28 25 19 27 21 15 14 39 47
2	23 25 21 37 45 11 44 14 15 20 19 9 10
3	23 25 21 37 45 11 44 14 20 15 19 9 10

4.9 Conclusion

Presented in this chapter is a multi-pulse algorithm using the theory of chapter three. The algorithm is guaranteed to converge and in practice does so very quickly. The algorithm was tested on true multi-pulse data which was generated and it was able to detect all the parameters involved in that generation. The algorithm will be used in chapter six for synthetic speech production. In this chapter the algorithm used the covariance method to obtain the multi-pulse parameters. The

main thrust of the algorithm presented in this chapter is to implement the multi-pulse equations derived in Chapter three. There are many variations on the algorithm which would be useful depending on the particular application. The next chapter discusses variations in the multi-pulse parameters along with multi-pulse data with additive noise.

5 SIMULATIONS USING THE MULTI-PULSE ALGORITHM

This chapter investigates the accuracy of the algorithm presented in chapter four. Simulations are done when certain multi-pulse parameters are varied. Included will be a discussion on how the spacing of pulses, the number of pulses, the frame size, and noise on the data affect the convergence of the algorithm. Many multi-pulse models were generated and tested and two of them which are representative of these models are shown in Fig. 5.1 and Fig. 5.2. In these figures the output is generated by

$$s(n) = \sum_{i=1}^{p} \alpha_i s(n-i) + x(n)$$

where the LPC coefficients are shown in Table 5.1. The output sequence is $s(n)$ and the input sequence is $x(n)$. The algorithm was given the output data as shown in the two figures. In both cases it converged to the true location and amplitude of the impulses. The error at each iteration of the algorithm is shown in Table 5.2.

5.1 Spacing of the Pulses

A major concern with the algorithm was its ability to detect the input sets where the spacing was close or far apart. It turned out in simulation that it did not matter how the impulses were spaced. As can be seen in Fig. 5.1 the impulses are located close together and still the algorithm is able to detect their positions. Likewise, the algorithm detects the positions when the impulses are far apart as in Fig. 5.2. It

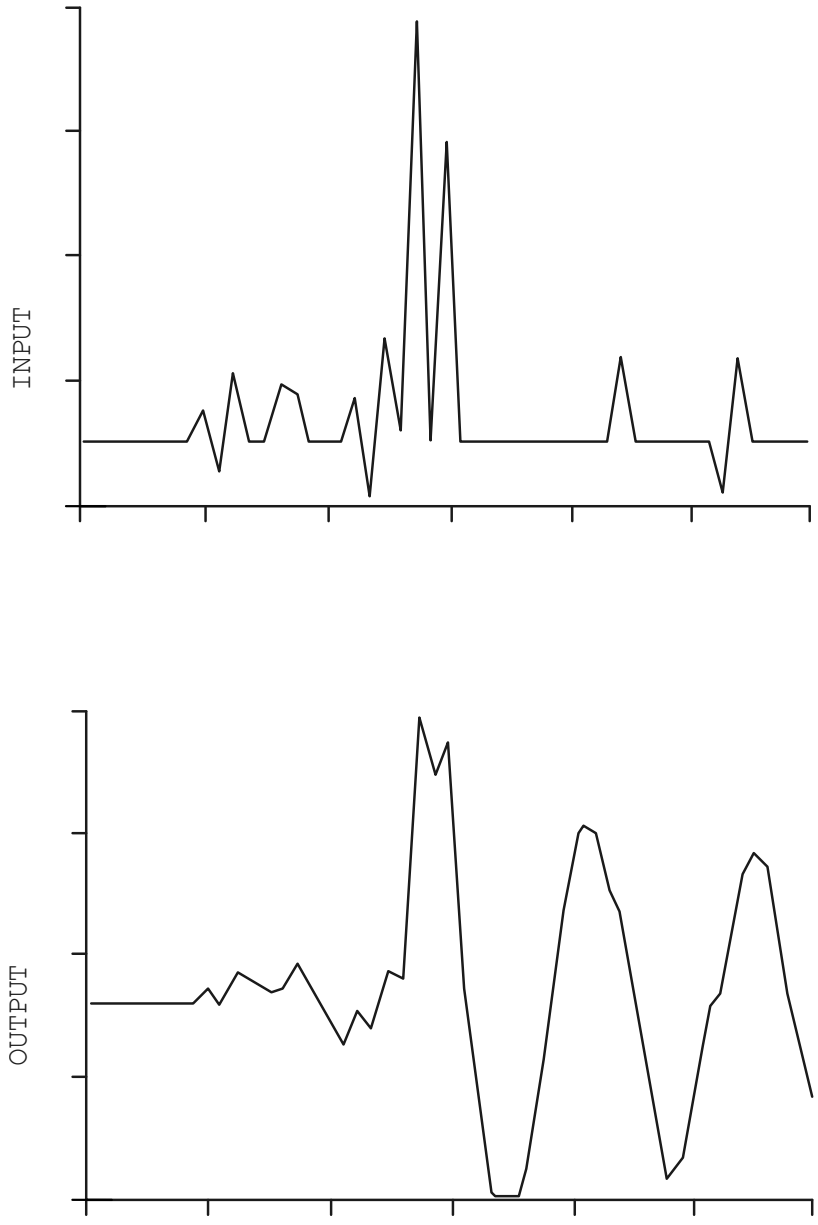

Figure 5.1 Multi-Pulse Generated Data

is important that the algorithm be able to detect impulses close together because the excitation set for speech comes in bursts of energy. In fact, the next chapter will show that in almost all cases of real speech the

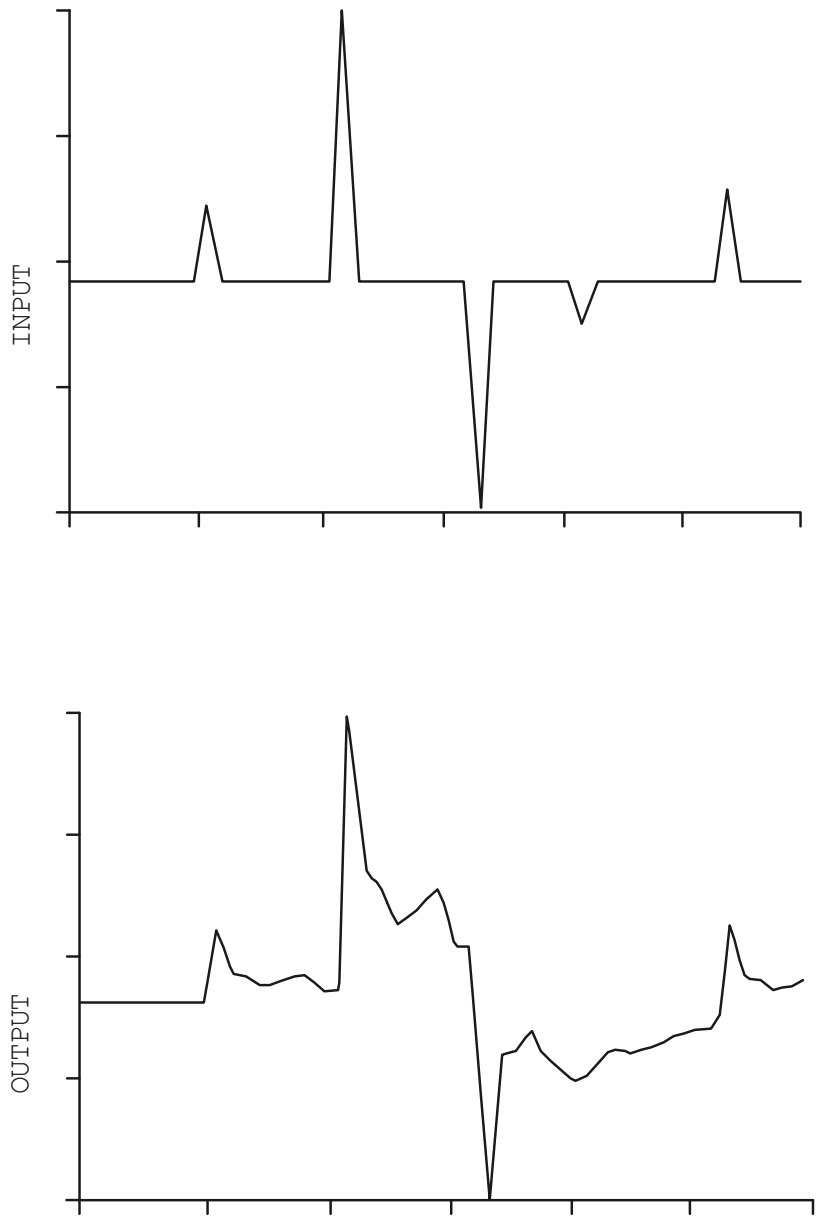

Figure 5.2 Multi-Pulse Generated Data

algorithm will locate pulse positions very close to each other.

5.2 The Frame Size and the Number of Pulses

Table 5.1 LPC Coefficients for Figures 5.1 and 5.2

α_i	Fig 5.1	Fig. 5.2
α_1	.85776	.4312
α_2	-.40311	.2015
α_3	-.35678	.0060
α_4	.35421	.1115
α_5	-.31809	.12682
α_6	-.23467	.092662
α_7	.00000	-.17475

Table 5.2 Error for Algorithm on Fig. 5.1 & 5.2

Iteration	Fig 5.1 Error	Fig 5.2 Error
1	$.2391174 \times 10^4$	$.1933516 \times 10^3$
2	$.6309375 \times 10^2$	0.0000000
3	0.0000000	

The frame size and the number of pulses are included in one section because they are so related. In this section N will denote the frame size and I will denote the number of pulses. As discussed in 4.5 the number of pulses are related to the convergence of the algorithm. As measured in 4.5 the frame size did not affect the convergence of the algorithm.

If I is approximately equal to N then the matrix Φ' has been measured in test cases to have zero eigenvalues. When this is the case a corrective procedure, such as the one in 9.4, needs to be used. In all test cases if I was small compared to N then the matrix Φ' has been invertible. In most applications I will be smaller than N as a large number of pulses will require a large amount of information to be

transmitted over the channel.

5.3 Noise on the Multi-Pulse Data

In this section noise is added to the multi-pulse data. In the previous sections it was shown that the algorithm could detect a true multi-pulse model. This section is concerned with how robust the algorithm is. Fig. 5.3 shows how the noise is added.

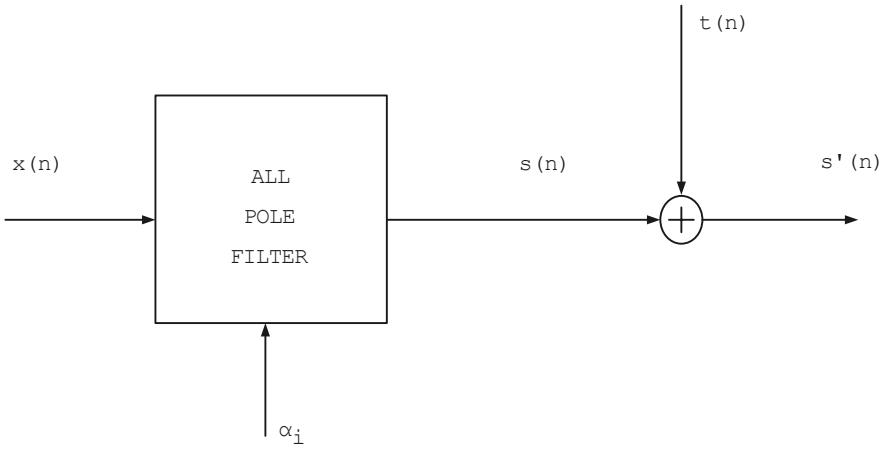

Figure 5.3 Multi-Pulse Data with Additive Noise

The noise sequence $t(n)$ is generated as follows:

$$t(n) = [\sum_{i=1}^{10} x_i(n)]/10 - 1/2 \tag{3.1}$$

where x_i is uniformly distributed on $[0,1]$. Then, $t(n)$ is multiplied by a constant γ so that $\gamma t(n)$ yields a specified signal to noise ratio defined as

$$\frac{S}{N} = \frac{\overline{s^2}}{\overline{n^2}} \tag{3.2}$$

with

$$\overline{s^2} = [\sum_{n=1}^{N} s^2(n)]/N \tag{3.3}$$

and

$$\overline{n^2} = [\sum_{n=1}^{N} t^2(n)]/N \tag{3.4}$$

The algorithm is given $s'(n)$ and from this data determines coefficients $\alpha_1, \ldots, \alpha_p$ and positions $p_1, \ldots, p_I$ and then the following sequence is formed

$$e(n) = \begin{cases} s(n) - \sum_{i=1}^{p} \alpha_i s(n-i) & n \neq p_1, p_2, \ldots, p_I \\ 0 & \text{otherwise} \end{cases} \tag{3.5}$$

Then, s(n) is graphed with e(n). The error e(n) is a measure of how accurately the algorithm can determine the prediction coefficients and pulse positions of the underlying model. Figures 5.4-5.11 show the results of additive noise on the signals in Fig. 5.1 and 5.2 with signal to noise ratios of 20.0,10.0,5.0, and 0.0 dB. For the signal in Fig. 5.2 the performance of the algorithm on the pulse positions is shown in Table 5.3. Similar results are obtained for the signal in Fig. 5.1. The results indicate that small deviations from the model do not result in a significant deviation in the parameters detected by the algorithm. This is crucially important since real speech, or data, is not generated by a true model. Looking at Figures 5.4-5.11 one can see that the resulting error sequence is not large compared to the amount of noise added.

In sections 4.6 and 4.7 theoretical bounds were derived for the change in the predictor coefficients and the total error due to the addition of white noise to the speech segment. These bounds are given by equations (4.6.13) and (4.7.9) as

$$\|\vec{\alpha} - \vec{\alpha}_n\| \;\leq\; (\sigma/\lambda_{\min})\|\vec{\alpha}\| \tag{3.6}$$

and

$$|E - E_n| \;\leq\; \sigma\{1 + (\lambda_{\max}/\lambda_{\min})\|\vec{\alpha}\|^2\} \tag{3.7}$$

The norm $\|\ \|$ is the ℓ_2 norm. The value σ is the variance of the white noise added to the speech segment. The alpha vector $\vec{\alpha}$ contains the prediction coefficients for the original speech segment. For the two speech segments concerned with here they are given in Table 5.1. The value E is the error made on the original speech segment. In this section E is zero as given in Table 5.2 since the original speech was generated in this chapter by a true multi-pulse model. The alpha vector $\vec{\alpha}_n$ contains the prediction coefficients measured for the noisy speech. The error made for the noisy speech is denoted by E_n. The largest and smallest eigenvalues of the Φ' matrix for the two speech segments are denoted as λ_{max} and λ_{min} respectively. Table 5.4 shows the eigenvalues of the Φ' matrix for speech segments I and II. For speech segment I the norm $\|\vec{\alpha}\|$ is given as

$$\|\vec{\alpha}\| = 1.143353 \quad \text{(Speech Segment I)} \tag{3.8}$$

Using the values of the eigenvalues from Table 5.4 for Speech Segment I in equations (3.6) and (3.8) yields

$$\|\vec{\alpha} - \vec{\alpha}_n\| \leq \sigma/(1.173432 \times 10^4) \quad \text{(Speech Segment I)} \tag{3.9}$$

and

$$|E_n| \leq 16.1851\sigma \quad \text{(Speech Segment I)} \tag{3.10}$$

These theoretical bounds are compared with the actual parameters measured in Table 5.6. for signal to noise ratios of 20.0, 10.0, 5.0, and 0.0dB. As can be seen from table 5.6 the bound given by (3.7) is a loose bound. The bound given by (3.6) is much tighter. The bound on the alphas was derived by invoking only one inequality (4.6.8) while the bound on the total error was derived by invoking four inequalities.

5.4 Noise on the Excitation Set

In this section noise will be added to the impulsive excitation set. Fig. 5.12 shows how the noise is added. The noise sequence is generated as in equation (3.1). The signal to noise ratio is calculated using $s(n)$ as the signal and $t'(n)$ as the noise. The results on Fig. 5.2 for signal to noise ratios of 20.0, 10.0, 5.0, and 0.0 dB are shown in Fig. 5.13-5.16. As seen in the figures the resulting error is much smaller than the noise added. For signal to noise ratios of 20, 10 and 5 dB the algorithm was able to locate all five pulse positions and for 0 dB it missed one position. As can be seen from the figures the algorithm works

better when additive noise is on the excitation set rather than the data itself.

5.5 Conclusion

Discussed in this chapter was the performance of the multi-pulse algorithm as certain parameters were varied. In addition, the algorithm was tested in the presence of noise. The addition of noise (> 5.0 dB) did not significantly affect the ability of the algorithm to detect the underlying parameters of the model.

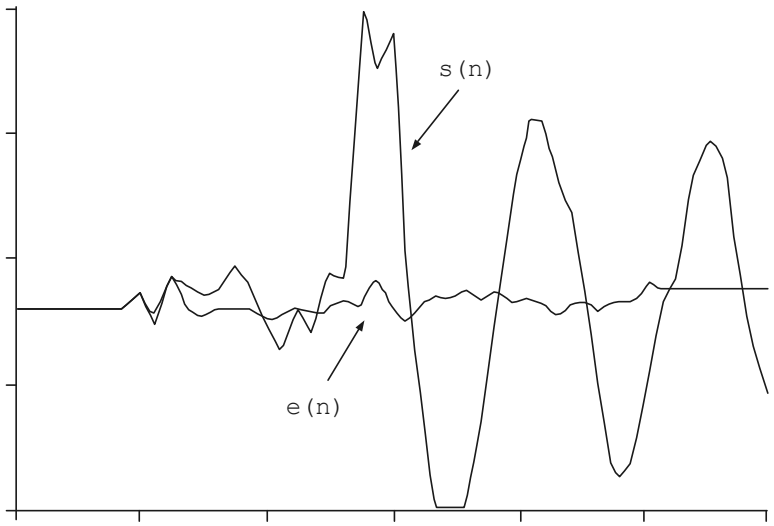

Figure 5.4 Data of Fig. 5.1 (S/N =20.0 dB)

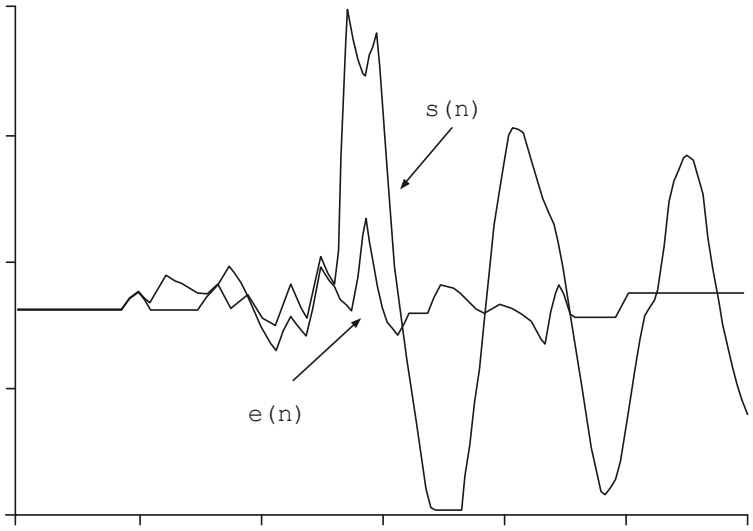

Figure 5.5 Data of Fig. 5.1 (S/N = 10.0 dB)

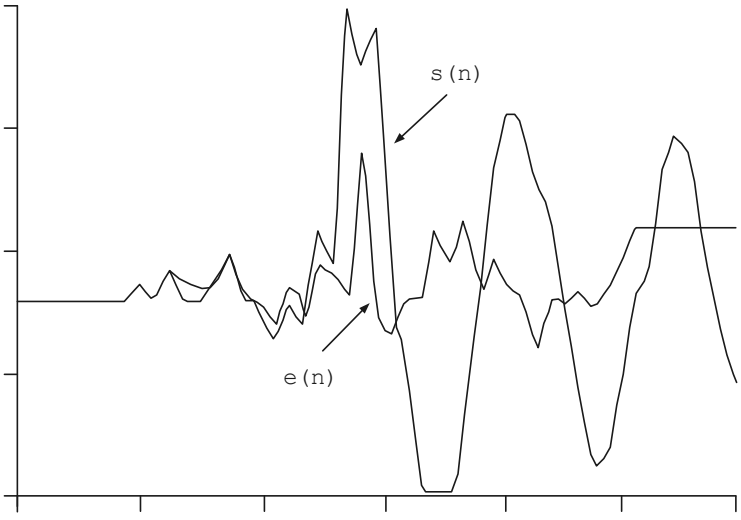

Figure 5.6 Data of Fig. 5.1 (S/N=5.0dB)

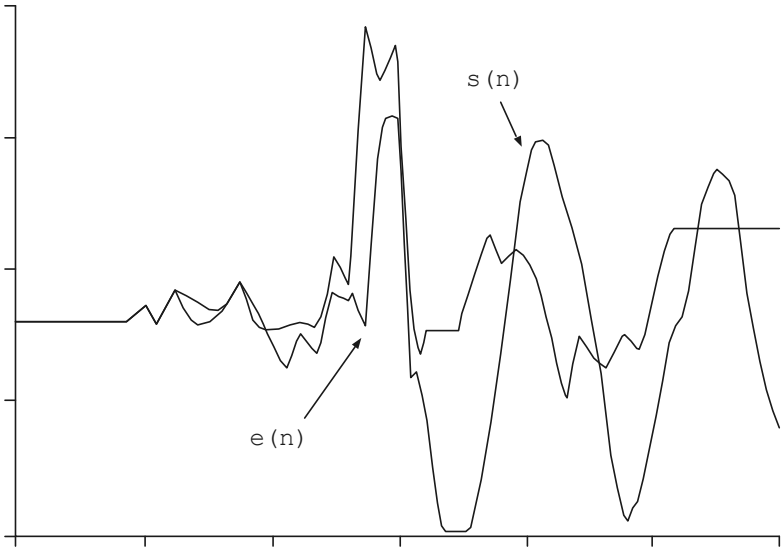

Figure 5.7 Data of Fig. 5.1 (S/N=0.0 dB)

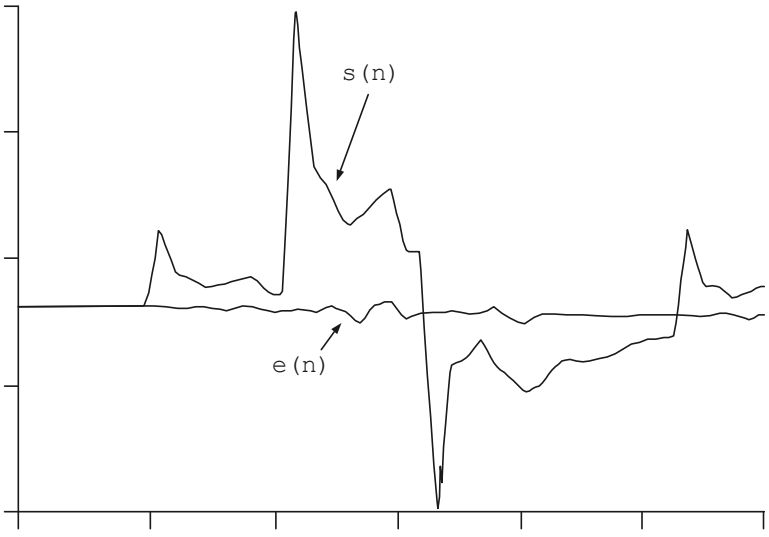

Figure 5.8 Data of Fig. 5.2 (S/N = 20.0 dB)

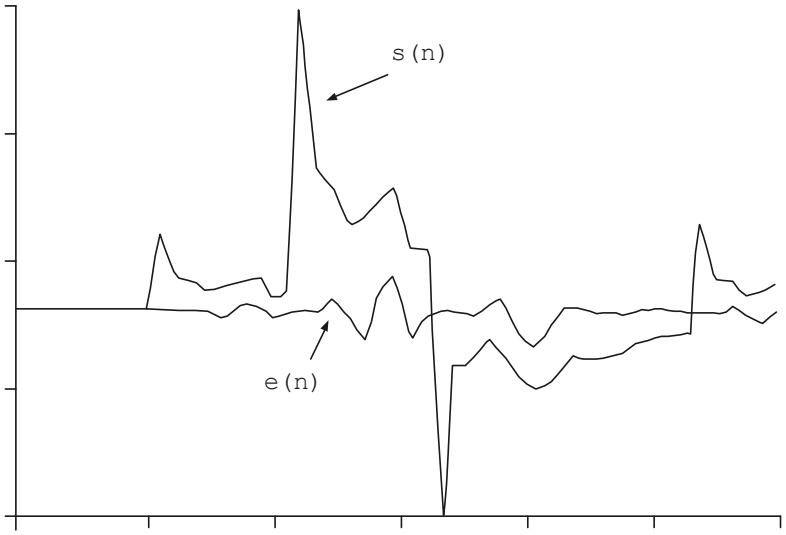

Figure 5.9 Data of Fig. 5.2 (S/N = 10.0 dB)

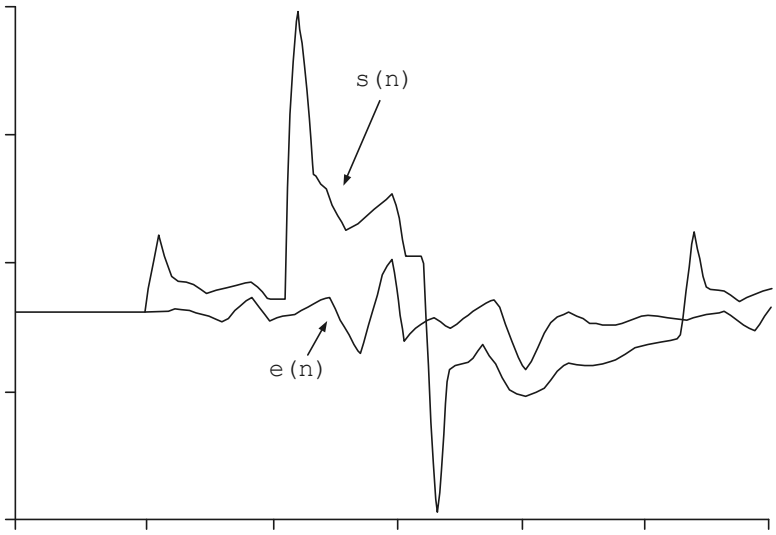

Figure 5.10 Data of Fig. 5.2 (S/N=5.0dB)

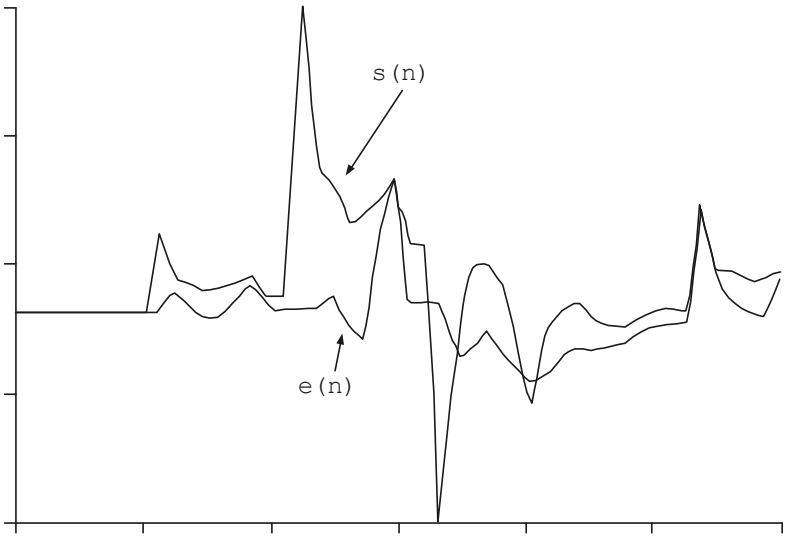

Figure 5.11 Data of Fig. 5.2 (S/N=0.0dB)

Table 5.3 Performance of the Algorithm for Signal in Fig. 5.2

S/N	Pulse Positions Accurate?
20.0	yes
15.0	yes
10.0	yes
5.0	no, missed one
0.0	no, missed two
-5.0	no, missed two

Table 5.4 The Eigenvalues of Φ' for Speech Segments I and II

	Speech Segment I	Speech Segment II
λ_1	$.1558461 \times 10^6$ ($\lambda_{\max}$)	$.2066472 \times 10^5$ ($\lambda_{\max}$)
λ_2	$.6239923 \times 10^5$	$.1416702 \times 10^5$
λ_3	$.3736242 \times 10^5$	$.1290627 \times 10^5$
λ_4	$.2125616 \times 10^5$	$.1328754 \times 10^5$
λ_5	$.1341647 \times 10^5$ ($\lambda_{\min}$)	$.1319526 \times 10^5$
λ_6	$.2349738 \times 10^5$	$.1275285 \times 10^5$
λ_7		$.9943853 \times 10^4$ ($\lambda_{\min}$)

Table 5.5 Perturbation of the Reflection Coefficients

$\|\vec{\alpha} - \vec{\alpha}_n\|$

S/N (dB)	Speech Segment I		Speech Segment II	
	Theoretical Bound	Measured Value	Theoretical Bound	Measured Value
20.0	.153050	.07830967	.011770	.005703469
10.0	1.53050	.370808	.1177695	.06002255
5.0	4.839862	.5718948	.372185	.1496651
0.0	15.304986	.7979619	1.176932	.2909281

Table 5.6 Perturbation of the Total Error

$$|E - E_n|$$

S/N (dB)	Speech Segment I		Speech Segment II	
	Theoretical Bound	Measured Value	Theoretical Bound	Measured Value
20.0	2.907×10^4	1.108×10^2	3.470×10^2	1.370
10.0	2.907×10^5	3.000×10^3	3.470×10^3	1.102×10^2
5.0	9.192×10^5	6.879×10^3	1.097×10^4	7.16×10^2
0.0	2.907×10^6	1.326×10^4	3.470×10^4	2.657×10^3

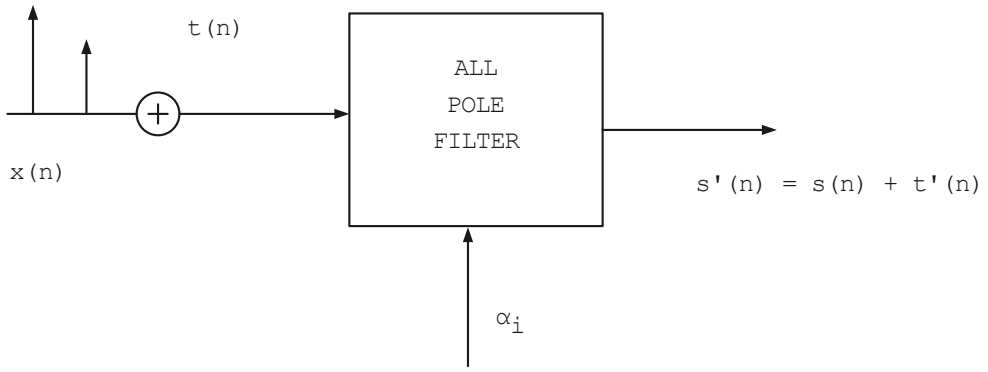

Figure 5.12 Noise on the Excitation Set

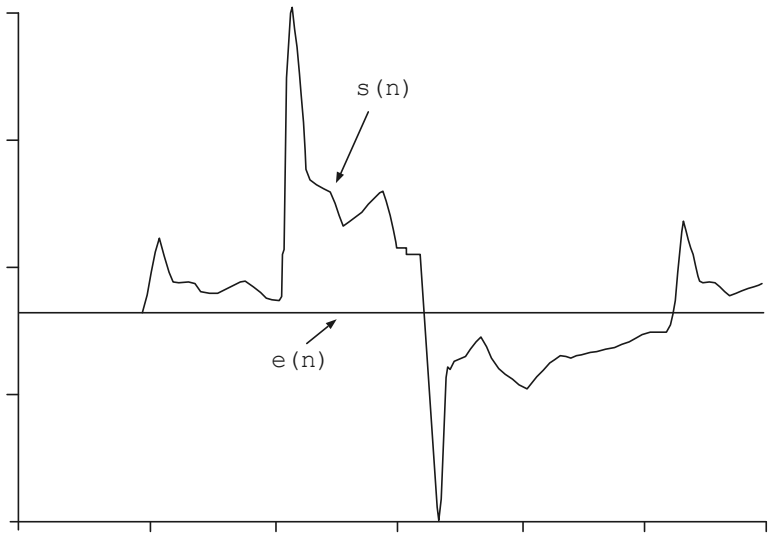

Figure 5.13 Data of Fig. 5.2 (S/N=20.0dB)

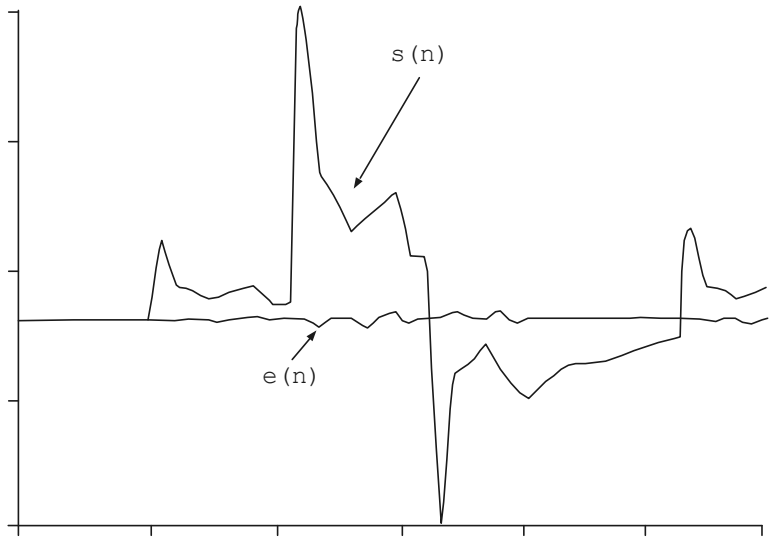

Figure 5.14 Data of Fig. 5.2 (S/N=10.0dB)

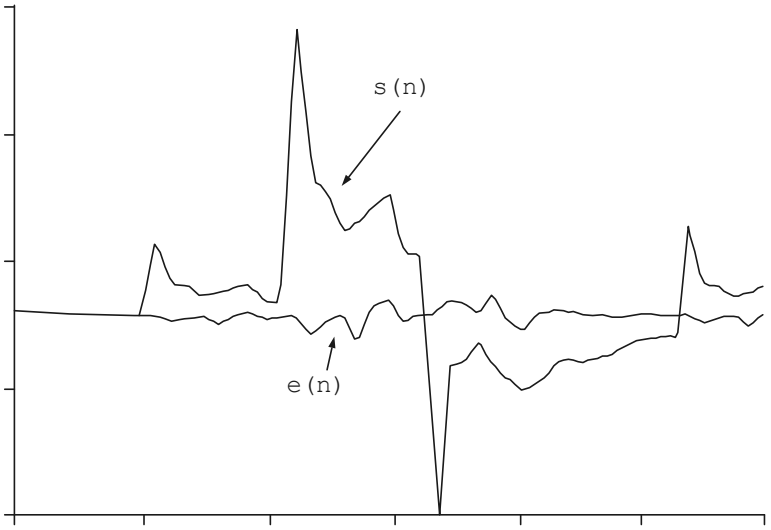

Figure 5.15 Data of Fig. 5.2 (S/N=5.0dB)

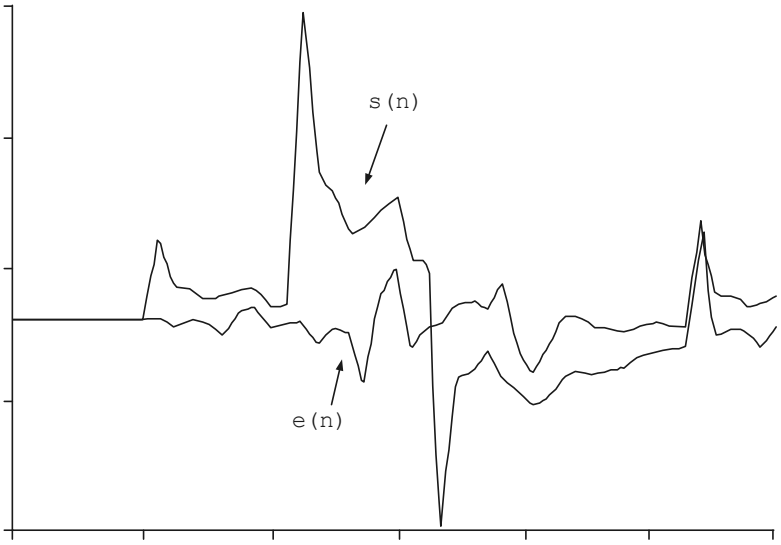

Figure 5.16 Data of Fig. 5.2 (S/N=0.0dB)

6 SYNTHETIC SPEECH PRODUCTION USING THE MULTI-PULSE ALGORITHM

6.1 Introduction

This chapter deals with the production of synthetic speech. The multi-pulse algorithm is given real speech data and from this it determines parameters to be used in a synthetic speech model. Also discussed is the performance of the prediction estimate when applied to speech.

6.2 The Prediction Estimate for Multi-Pulse and LPC

This section compares the performance of multi-pulse and LPC in terms of the prediction estimate. For LPC

$$\hat{s}(n) = \sum_{i=1}^{p} \alpha_i s(n-i) \tag{2.1}$$

For multi-pulse

$$\hat{s}(n) = \sum_{i=1}^{p} \alpha_i s(n-i) + \sum_{j=1}^{I} b_j \delta(n-p_j) \tag{2.2}$$

The error is then defined as

$$E = \sum_{n=1}^{N} e^2(n) \tag{2.3}$$

with

$$e(n) = s(n) - \hat{s}(n) \tag{2.4}$$

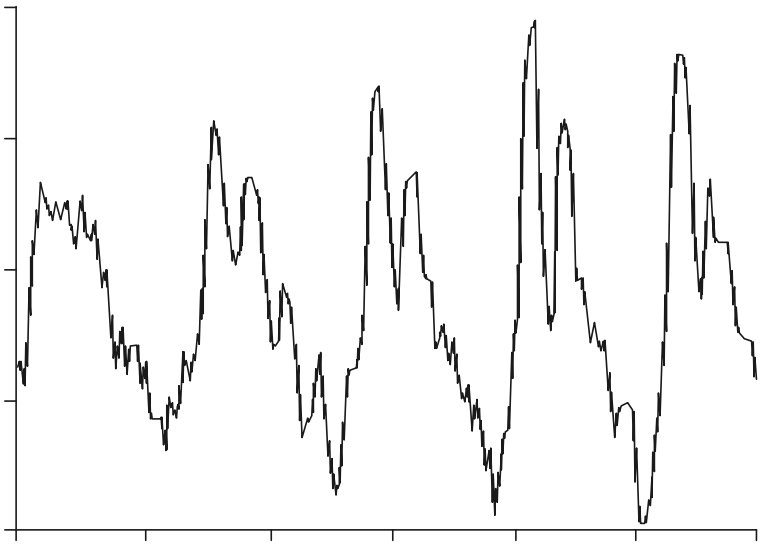

Figure 6.1 Speech Segment I (160 samples at 8khz)

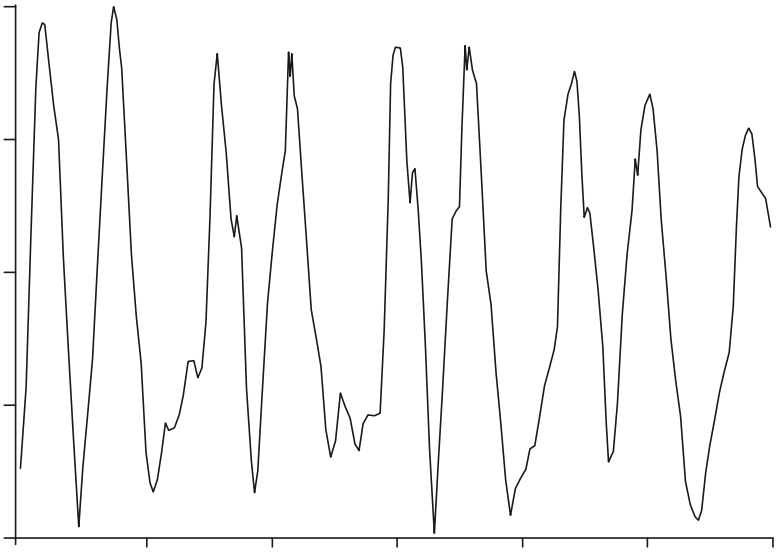

Figure 6.2 Speech Segment II (160 samples at 8 khz)

Table 6.1 S/N Calculations for Speech Segment I

		order of predictor				
		6	8	10	12	15
# of pulses	(LPC) 0	31.09	36.67	38.56	38.96	39.81
	5	33.64	41.28	43.70	44.40	46.28
	10	35.66	44.02	47.25	47.82	51.08
	15	37.36	46.44	49.71	51.02	53.53
	20	38.73	48.48	51.89	53.29	56.00
	25	40.02	50.47	54.28	55.11	58.24

Table 6.2 S/N Calculations for Speech Segment II

		order of predictor				
		6	8	10	12	15
# of pulses	(LPC) 0	24.99	26.74	27.45	28.17	28.27
	5	28.89	30.39	31.93	32.93	33.30
	10	31.52	32.86	34.22	35.11	35.92
	15	33.74	34.69	35.99	37.10	37.89
	20	35.36	36.45	37.86	38.45	39.60
	25	37.07	37.98	39.40	40.54	41.47

Many speech segments were tested and two which are representative of those are shown in Figures 6.1 and 6.2. For each speech segment the LPC and multi-pulse coefficients were obtained and the respective error functions were calculated using equation (2.3). Then the signal to noise defined as

$$\text{S/N} \;=\; 10 \log \frac{\Sigma\, s^2(n)}{\Sigma\, e^2(n)} \tag{2.5}$$

is calculated and tabulated for different numbers of predictor coefficients and different pulse positions. The results are tabulated in Table 6.1 and Table 6.2. The tradeoffs between higher order filters and number of pulses can be seen for these speech segments. As can be seen in Table 6.2 a 6th order multi-pulse with 5 pulse positions outperforms a 15th order LPC for that speech segment. Hence with multi-pulse there exists a possibility of reducing the order of the predictor thereby decreasing the number of computations required to calculate the multi-pulse parameters.

6.3 The Bit Rate for Standard LPC

The standard LPC system was discussed in chapter two. This section derives the bit rate necessary to operate the system when producing synthetic speech. The transmitter must send to the receiver the reflection coefficients, the gain, the pitch period and the voiced/unvoiced decision. The following parameters are now defined

(1) f_s : the sampling rate.

(2) N : the frame size.

(3) $f_N = f_s/N$: the number of frames per second.

(4) x_1 : # of bits appropriated for the gain.

(5) x_2 : # of bits for pitch period.

(6) x_3 : # of bits for each reflection coefficient.

(7) p : # of reflection coefficients.

The LPC bit rate LPC_{BR} is then given by

$$LPC_{BR} = f_N(px_3 + x_1 + x_2 + 1) \tag{3.1}$$

For the LPC system tested the following values were assigned

(1) $f_s = 8000$

(2) $N = 160$

(3) $f_N = 50$

(4) $x_1 = 10$

(5) $x_2 = 6$

(6) $x_3 = 10$

(7) $p = 12$

So the total bit rate is

$$LPC_{BR} = 6850 \text{ bits/sec.} \tag{3.2}$$

For multi-pulse the situation is more complex. In this case, the transmitter must send the pulse positions for each frame. Sending the pulse positions could require a lot of bits. The next section introduces a new method to code the pulse position and compares it with the theoretically optimal one.

6.4 Transmitting the Pulse Positions for Multi-Pulse

In this section the problem is to describe k pulse positions in a block of N samples with as few bits as possible. The algorithm is as

follows:

(1) Is there a pulse in the next 2^b samples ? Yes - go to (3). No - go to (2)

(2) Transmit a "0". Shift over 2^b samples. go to (1)

(3) Transmit a "1". Transmit b bits which define the position of the next pulse. Shift over to that pulse position. Go to (1).

The parameter to be determined is b. It will be chosen to minimize the maximum bit rate incurred in the above algorithm. In a block size of N there will be a "1" transmitted k times (one for each pulse). In the block size of N the maximum number of times a "0" can be transmitted is $N/2^b$. If a "0" is transmitted then no more information is sent. If a "1" is transmitted then b bits are sent to describe the position. So the maximum number of bits required for the transmission is

$$\text{Max}_B = kb + k + N/2^b = k(b+1) + N/2^b \tag{4.1}$$

If this function is minimized with respect to b then the optimal value of b is given as

$$b = [\ln(N * \ln(2)/k)]/\ln(2) \tag{4.2}$$

since b must be an integer one needs to check the value of equation (4.1) for the two closest integers to b when b is calculated via equation (4.2). This will suffice since the function defined by equation (4.1) is uni-modal. For example, let us attempt to find out how many bits are

required to describe 10 pulses in a block size of 160 samples using the above algorithm. The value of b from equation (4.2) is

$$b = 3.4712 \tag{4.3}$$

Looking at equation (4.1) for b=4 and b=3 the optimal is

$$\text{Max}_B = 60 \text{ bits} \tag{4.4}$$

and it holds this value for b=3 or b=4. The maximum number of bits required to transmit these pulse positions if the algorithm were not used is 160 bits. That is, send a stream of 160 bits with a "1" denoting a pulse in that position and a "0" denoting no pulse in that position. The algorithm as stated above yields a major improvement over that method. The minimum number of bits to send the pulse positions is given by

$$\text{min}_B = \log_2 \binom{160}{10} \approx 51 \tag{4.5}$$

By comparing equations (4.5) and (4.4) one can see that in this case the algorithm is at a bit rate relatively close to the optimum. The next section estimates the bit rate of a multi-pulse system using the results of this section.

6.5 The Bit Rate for Multi-Pulse

This section derives the bit rate for a multi-pulse system. Previous researchers have used 8 pulses every ten milliseconds which would correspond to 16 pulses every 160 samples. From the previous section in

order to describe 16 pulses every 160 samples it will take 84 bits. The parameters for multi-pulse transmission are

(1) f_s : the sampling rate

(2) N : the block size

(3) f_N : the number of blocks per second

(4) k : the number of impulses per block

(5) y_1 : the number of bits per frame for describing the positions

(6) p : the number of predictor coefficients

(7) y_2 : the number of bits to quantize each predictor coefficient

(8) y_3 : the number of bits to quantize each gain

The values chosen for a multi-pulse model were

(1) f_s = 8000

(2) N = 160

(3) f_N = 50

(4) k = 16

(5) y_1 = 84

(6) p = 8

(7) y_2 = 10

(8) y_3 = 3

The number of bits required for multi-pulse is

$$MP_{BR} = f_N(py_2 + ky_3 + y_1) \tag{5.1}$$

In the above example

$$MP_{BR} = 10.6 \text{ kBits} \tag{5.2}$$

The parameter y_3 is the number of bits to quantize each gain (and there are k of them in a block size of N). The gain for multi-pulse is a very slowly changing quantity. It in fact is chosen as the k largest values of the LPC error function and they turn out to be very close to each other in absolute value. The first bit of y_3 will determine the sign of the pulse excitation. The next two bits will be used in an adaptive delta modulation scheme to determine the magnitude of the pulse in terms of the magnitude of the pulse before it. In our example the number of predictor coefficients is eight which is smaller than the number used in the standard LPC example. This is because of the tradeoffs discussed earlier. Specifically, the more pulse positions one uses the less predictor coefficients needed to attain a specific signal-to-noise ratio. It has been found that the number of bits required for each predictor coefficient is essentially the same for multi-pulse and LPC. The next section discusses alternate ways of formulating the gain for the multi-pulse system.

6.6 Alternate Input Sequences for Multi-Pulse

One of the most disappointing aspects of speech production is that the quality of speech as perceived by the human ear is not necessarily a function of a mathematical measure. For example, there are cases where

speech segments having a signal-to-noise ratio of 25.0 dB sound much better than speech segments having a signal to noise ratio of 30.0 dB. This is true for most mathematical error measures. Hence, when it comes down to actually producing synthetic speech the major concern is not the minimization of some error. This is the case for present day LPC systems. The gain for the LPC system is usually assigned to be the energy in the error signal. This assignment is a suboptimal one but does in fact yield improved results over the case where the optimal gain is used. The same may be the case for a multi-pulse system. That is, a sub-optimal gain may yield a better sounding system. Unfortunately, the mathematical equations will give no insight into what this possible gain might be. Jain [19] has suggested a different gain from the one used in the thesis so far. The gain assigned by Jain is such that after the pulse has been excited at time t_0 the error incurred at that position is equal to the short time error of the previous positions. This is an intuitive way of keeping the error level constant throughout the production of the synthetic speech. In more detail the gain is as follows

- Compute $f(n)$ as
$$f(n) = s(n) - \sum_{i=1}^{p} \alpha_i s(n-i)$$
- If a pulse is at position n its magnitude is given as
$$u_n = f(n) - c v_{n-1}\, \mathrm{sign}(f(n))$$
- v is the short-time absolute value of the signal
$$v_n = q\, v_{n-1} + (1-q)|e_n|$$

The value of q was chosen as 0.95 and c took on the values 0.5 or 1.

This gain algorithm was employed in a multi-pulse system using the prediction coefficients calculated for multi-pulse in chapter 3. Listening tests do not indicate that the choice of gain via the above method is as good as the choice of the gain via the equations developed in chapter 3. In fact, at higher bit rates the performance of the above system did not improve as much as the other multi-pulse system. Jain is one of the few authors to realize the importance of the prediction residual in determining the pulse positions. Perhaps there is another way to sub-optimally define the gain for multi-pulse and obtain a better sounding speech segment. The extension from LPC to multi-pulse (in the way the gain is sub-optimally defined) does not yield an improved performance for the multi-pulse case. The next section deals with the LPC distance measure.

6.7 LPC Distance Measure

The LPC distance measure is a measure of the distance between the predictor coefficients obtained from two segments of speech. In the application here, one segment will be the original speech and the next will be the synthetically produced speech for either multi-pulse or LPC. The LPC distance measure used here [20] is

$$D(\hat{a}, a) = \log[(aRa^t)/(\hat{a}R\hat{a}^t)] \tag{7.1}$$

R denotes the correlation matrix of the synthetically produced speech.

The prediction coefficients of the speech are denoted by "a" while the prediction coefficients of the synthetically produced speech are

denoted by $\hat{a}$. Define E as

$$E = \hat{a} R \hat{a}^t \tag{7.2}$$

and $\tilde{E}$ as

$$\tilde{E} = a R a^t \tag{7.3}$$

Rabiner and Schafer [8, p.460] show that E is the energy in the error signal if the coefficients $\hat{a}$ were used to estimate the synthetic speech. Likewise, $\tilde{E}$ is the energy in the error signal if the coefficients a were used to estimate the synthetic speech. Since $\hat{a}$ was chosen to minimize the energy in the error signal it follows that

$$E \leq \tilde{E} \tag{7.4}$$

Using equations (7.2) and (7.3) it follows that (7.1) can be written as

$$D(\hat{a}, a) = \log \frac{\tilde{E}}{E} \tag{7.5}$$

For the frequency version of equation (7.5) one obtains [8, p.433]

$$D(\hat{a}, a) = \log \frac{\int_{-\pi}^{\pi} |S(e^{jw})|^2 |A(e^{jw})|^2 \, dw}{\int_{-\pi}^{\pi} |S(e^{jw})|^2 |\widetilde{A}(e^{jw})|^2 \, dw} \tag{7.6}$$

$S(e^{jw})$ is the Fourier transform of the synthetic speech. Also,

$$A(e^{jw}) = 1 - \sum_{k=1}^{p} \alpha_k \, e^{-jwk} \tag{7.7}$$

and

$$\widetilde{A}(e^{jw}) = 1 - \sum_{k=1}^{p} \hat{\alpha}_k \, e^{-jwk} \tag{7.8}$$

In equations (7.7) and (7.8) the alphas are the prediction coefficients for the speech and synthetically generated speech respectively. The LPC distance measure for multi-pulse and LPC is shown in Table 6.3 for 48 different blocks of speech. It can be seen from Table 6.3 that multi-pulse yields an improved LPC distance measure when compared to LPC.

6.8 Conclusion

This chapter has dealt with the production of synthetic speech. It has compared the prediction estimate of multi-pulse with the prediction estimate of LPC and shown that multi-pulse can yield a major improvement.

The bit rates for LPC and multi-pulse have been derived. A new algorithm was introduced to code the multi-pulse positions. The algorithm is extremely easy to implement in a real-time system and requires very little computations. Finally, the chapter introduced the LPC distance measure as a way of comparing LPC and multi-pulse. Multi-pulse outperformed LPC with this measure.

Table 6.3 LPC Distance Measure for LPC and Multi-Pulse

Speech Block	LPC	Multi-Pulse
1	2.676	1.926
2	2.506	1.569
3	3.868	4.833
4	4.466	1.399
5	3.686	1.187
6	3.372	3.349
7	3.080	1.967
8	3.251	3.553
9	4.088	1.703
10	5.106	3.414
11	4.402	2.887
12	4.463	1.893
13	4.767	2.175
14	3.286	1.812
15	3.806	3.010
16	4.423	3.607
17	4.945	4.177
18	4.459	2.501
19	5.015	2.928
20	4.649	3.030
21	4.807	3.140
22	5.163	3.546
23	3.068	1.734
24	2.497	2.646

Table 6.3 (continued)

Speech Block	LPC	Multi-Pulse
25	3.147	4.353
26	3.934	2.823
27	2.104	4.647
28	3.202	2.225
29	4.073	0.251
30	4.473	0.880
31	3.568	2.119
32	2.305	0.647
33	0.183	2.062
34	3.072	0.995
35	4.383	0.013
36	5.112	3.566
37	3.758	1.872
38	3.343	1.666
39	3.588	0.785
40	3.493	2.205
41	3.324	2.367
42	2.989	1.940
43	2.540	1.202
44	3.237	2.888
45	3.127	2.029
46	2.555	3.075
47	2.744	1.854
48	1.316	0.769

7 SUMMARY AND FURTHER RESEARCH

7.1 Summary

In this dissertation, a study of LPC with multi-pulse excitation was presented. The multi-pulse system was compared with an LPC system and bit rates were established for both of these. Multi-pulse performs better than LPC when allowed to run at 10.6 kBits. As discussed earlier, when the bit rates are the same there is not really a significant improvement over LPC.

New contributions by this dissertation include

- Solution of a set of equations for multi-pulse to determine the optimal prediction coefficients for a given set of pulse positions. These equations are uncoupled from the gain.
- A mathematical derivation of the optimal pulse locations for multi-pulse.
- A theoretical proof of convergence for the algorithm.
- An algorithm to transmit the pulse locations.

7.2 Applications of the Multi-Pulse Algorithm

One important application of the multi-pulse system is in seismic processing. In this application an explosion (producing a waveform) is produced at sea level and the return signal is measured. This return signal depends on the contours of the ocean floor and layers below that. At each discontinuity a portion of the signal is echoed back. If an

all-pole filter is used to describe the explosion then the problem can be modelled using multi-pulse. The pulse locations will correspond to locations where the signal is being echoed from, hence it will determine where the discontinuities in the layers are. Its ability to locate pulse positions close together will be useful in this application.

Another application is in echo cancellation. When a signal is comprised of an original signal plus an echoed version of this signal then the multi-pulse algorithm should be able to detect the original signal if the original signal can be adequately modelled by an all-pole filter. In this example, the multi-pulse algorithm will look for two pulse positions and amplitudes (the original excitation and its echo). Once it has determined the magnitude of the echo it can use this to remove the echo from the degraded signal.

Speech recognition is another possible application for multi-pulse. If the multi-pulse prediction coefficients are close when a person is saying the same sentence over and over then they could be used to verify a speaker based on stored information. As in LPC a multi-pulse distance measure could be defined and a threshold set up.

Alternate methods in speech production are also nice with multi-pulse. The multi-pulse predictor could be used in an adaptive predictive coder. This application is very nice because the error does not propagate at the receiver as in LPC. The error propagation at the receiver is one of the major causes of degradation in an LPC type system.

There are many other applications of the multi-pulse system.

Basically, whenever the multi-pulse model is valid the algorithm will be useful in determining its parameters.

8 REFERENCES

[1] Flanagan, J.L., et al., "Speech Coding," IEEE Trans. Commun., Vol. COM-27, pp. 710-737, April 1979.

[2] Atal, B.S. and J.R. Remde, "A New Model of LPC Excitation for Producing Natural-Sounding Speech at Low Bit Rates," Proc. ICASSP, Paris, France, pp. 614-617, 1982.

[3] Atal, B.S. and S.L. Hanauer, "Speech Analysis and Synthesis by Linear Prediction," J. Acoust. Soc. Amer., Vol. 50, pp. 637-655, Aug. 1971.

[4] Wakita, H., "Estimation of the Vocal Tract Shape by Optimal Inverse Filtering and Acoustic/Articulatory Conversion Methods," SCRL Monograph, No. 9, Speech Communication Research Laboratory, Santa Barbara, CA, 1972.

[5] Wakita, H., "Direct Estimation of the Vocal Tract Shape by Inverse Filtering of Acoustic Speech Waveforms," IEEE Trans., AU-21, pp. 417-427 ,1973.

[6] Morse, P.M. and K.V. Ingard, "Theoretical Acoustics," McGraw-Hill Book Co., New York, 1968.

[7] Markel, J.D. and A.H. Gray,Jr. "Linear Prediction of Speech," New York, Springer-Verlag, 1976.

[8] Rabiner, L.R. and R.W. Schafer, "Digital Processing of Speech Signals," Prentice-Hall, New Jersey, 1978.

[9] Atal, B.S., "Predictive Coding of Speech at Low Bit Rates," IEEE Trans. Commun., Vol. COM-30, No. 4, pp. 600-614, 1982.

[10] Makhoul, J., "Linear Prediction: A Tutorial Review," Proc. IEEE, Vol. 63, pp. 561-580, April, 1975.

[11] Atal, B.S., "On Determining Partial Correlation Coefficients by the Covariance Method of Linear Prediction," J. Acoust. Soc. Amer., Vol. 62, Suppl. 1, pp. 564, Fall 1977.

[12] Holmes, J.N., "Formant Excitation Before and After Glottal Closure," Conf. Rec. 1976, IEEE ICASSP, pp. 39-42, April, 1976.

[13] Atal, B.S. and M.R. Schroeder, "Predictive Coding of Speech Signals and Subjective Error Criteria," IEEE ICASSP, Vol. ASSP-27, pp. 247-254, June, 1979.

[14] M.R. Schroeder, B.S.Atal, and J.L. Hall, "Objective Measures of Certain Speech Signal Degradations Based on Properties of Human Auditory Perceptions," Frontiers of Speech Communication Research, edited by B. Lindblom and S. Ohman, London, Academic Press, pp. 217-229, 1979.

[15] Singhal Sharad and B.S. Atal, "Optimizing LPC Filter Parameters for Multi-Pulse Excitation," Proc. ICASSP, Boston, Mass., pp. 781-784, 1983.

[16] Flanagan, J.L., "Speech Analysis, Synthesis and Perceptions," 2nd Edition, Springer-Verlag, New York, 1972.

[17] Schroeder, M. R., "Vocoders: Analysis and Synthesis of Speech," Proc. IEEE, Vol. 54, pp. 720-734, 1966.

[18] Makhoul, J., "Spectral Analysis of Speech by Linear Prediction," IEEE Trans. on Audio and Electronics, Vol. Au-21, No. 3, pp. 140-148, June, 1973.

[19] Jain, V.K., "Simplified Algorithm for Multi-Pulse LPC Analysis of Speech," Proc. GlobeComm., pp. 789-793, Nov., 1983.

[20] Itakura, F. "Minimum Prediction Residual Principle Applied to Speech Recognition," IEEE TRANS. ASSP, Vol. ASSP-23, pp. 169-176, 1975.

[21] Bartle, R.G., "The Elements of Real Analysis, 2nd Edition," John Wiley & Sons, Inc., New York, 1976.

[22] Beyer, William H., "CRC Standard Mathematical Tables, 25th Edition," CRC Press Inc., West Palm Beach, Fla., 1978.

[23] Hines, W.W. and D.C. Montgomery, "Probability and Statistics in Engineering and Management Science, 2nd Edition," John Wiley & Sons, Inc., New York, 1980.

[24] Naylor, A.W. and G.R. Sell, "Linear Operator Theory in Engineering and Science, 2nd Edition," Springer-Verlag, New York, 1982.

[25] Steiglitz, Kenneth, "On the Simultaneous Estimation of Poles and Zeros in Speech Analysis," IEEE Trans. ASSP, vol. ASSP-25, pp. 229-234, 1977.

[26] Morikawa, H. and H. Fujisaki, "Adaptive Analysis of Speech Based on a Pole-Zero Estimation," IEEE Trans. ASSP, vol. ASSP-30, pp. 77-87, Feb., 1982.

[27] Miyanaga, Y., et. al., "A Speech Analysis Algorithm Which

Eliminates the Influence of Pitch Using the Model Reference Adaptive System," IEEE Trans. ASSP, vol. ASSP-30, pp. 88-96, Feb. 1982.

[28] Konvalinka, I.S. and M.R. Matausek, "Simultaneous Estimation of Poles and Zeros in Speech Analysis and ITIF-Iterative Inverse Filtering Algorithm," IEEE Trans. ASSP, vol. ASSP-27, pp. 485-491, Oct. 1979.

9 APPENDICES

9.1 Derivation of the Multi-Pulse Coefficients when Pulse Positions and Amplitudes are Known

This section derives an alternate set of multi-pulse equations for the case when the pulse positions and amplitudes are known. As before, the prediction estimate is

$$\hat{s}(n) = \sum_{i=1}^{p} \alpha_i s(n-i) + \sum_{j=1}^{I} b_j \delta(n-p_j) \tag{1.1}$$

and with

$$E = \sum_{n=1}^{N} e^2(n) \qquad e(n) = s(n) - \hat{s}(n) \tag{1.2}$$

From taking the partial of E with respect to the prediction coefficients and setting equal to zero one obtains

$$\sum_{n=1}^{N} (s(n) - \sum_{i=1}^{p} \alpha_i s(n-i) - \sum_{j=1}^{I} b_j \delta(n-p_j)) s(n-k) = 0 \tag{1.3}$$

which can be written as

$$\sum_{n=1}^{N} \sum_{i=1}^{p} \alpha_i s(n-i) s(n-k) = \sum_{n=1}^{N} s(n) s(n-k) - \sum_{j=1}^{I} b_j s(p_j - k) \tag{1.4}$$

Standard LPC yields the same equations except for the summation with b_j which is the correlation between the input and the speech. So if for LPC the equations are written in matrix form as

$$\Phi \vec{\alpha}_{LPC} = \Psi \tag{1.5}$$

then for multi-pulse they are

$$\Phi\vec{\alpha}_{MP} = \Psi - \Psi^0 \tag{1.6}$$

or

$$\vec{\alpha}_{MP} = \vec{\alpha}_{LPC} - \Phi^{-1}\Psi^0 \tag{1.7}$$

where Ψ^0 is a column vector with kth element

$$\psi_k = \sum_{j=1}^{I} b_j s(p_j - k) \tag{1.8}$$

These equations might be used in the application to LPC where the gain and pitch period(pulse locations) is known and an improved set of α's is desired. These equations would not be used for a pitch synchronous LPC system where the parameters are updated every pitch period.

9.2 Subroutine to Calculate Multi-Pulse Parameters

This section contains a program written in C programming language which was used to calculate the multi-pulse parameters.

```
/*****************************************************************
****    This subroutine calculates the multi-pulse
****    parameters for a given set of data.  The multi-pulse
****    coefficients and pulse positions will be passed back to
****    the caller.
*****************************************************************/

#include <stdio.h>
#include <math.h>

  multip(n,i20,iter,p,s,nn,dec,pos,alpha,code)
```

```
  float s[ ],alpha[ ];
  int n,i20,iter,p,nn,dec,code,pos[ ];

/*************************************************************
****    Meaning of Variables Passed to Subroutine.

****    n:        the number of data samples in the speech block
****    i20:      the number of impulses to be used.
****    iter:     the number of iterations for the algorithm.
****    p:        the number of predictor coefficients.
****    s:        the array containing the speech.  if s[x] is
                  passed to the algorithm, the algorithm will
                  look for the initial data in samples s[x],s[x+1],
                  ...,s[x+p-1] and the speech data in s[x+p],s[x+p+1],
                  ...,s[x+n+p-1].
****    nn:       the number of initial pulse positions.
****    dec:      the decrement for the number of pulse positions used in
                  for(i=nn;i>=i20;i-=dec) where i denotes the number of
                  pulse positions the algorithm is using at that particular
                  iteration.
****    pos:      the array containing the positions of the pulses. if
                  pos[y] is passed to the algorithm it will return pulse
                  positions pos[y+1],...,pos[y+i20].
****    alpha     the array containing the alphas.  If alpha[j] is passed
                  to the algorithm it will return alpha(1) in location
                  alpha[j+1],...,alpha(p) will be in location alpha[j+p].
****    code:     if code = 0 then the algorithm calculates the optimal
                  coefficients for the fixed locations passed to it.
*************************************************************/
{
#define begin {
#define end   }
#define then
float kp[1000],phi[20][20],sum,psi[20],sum1,sum2,sum3;
float d[20],v[20][20],y[20],e[1000],b[1000];
int i,j,j3,j4,k,k2;
for(i=nn;i>=i20;i-=dec)   /*start pulse position loop */
begin
for(k2=1;k2<=n;k2++)
        kp[k2]=1.0;   /*initialize correlation select function */
if(code==1)
   begin
      for(k2=1;k2<=i20;k2++)   /* if code =1 then use pulse positions */
        kp[pos[k2]]=0.0;       /* passed to the algorithm to define */
```

```
   end                          /* new correlation select function */
for(j4=1;j4<=iter;j4++)         /* begin iteration loop */
begin
for(j3=1;j3<=p;j3++)
   begin                    /* begin loop to calculate the multi-pulse */
      for(k=1;k<=j3;k++)    /* phi matrix. (identical to standard lpc */
        begin               /* but with kp multiplied in with the sum */
          sum=0.0;
          for(j=1;j<=n;j++)
            sum+=s[j-j3+p-1]*s[j-k+p-1]*kp[j];
          phi[j3][k]=sum;
        end
   end
for(j3=1;j3<=p;j3++)
   begin
      sum=0.0;              /* calculate the multi-pulse psi matrix */
      for(j=1;j<=n;j++)     /* similar to lpc */
        sum+=s[j+p-1]*s[j+p-j3-1]*kp[j];
      psi[j3]=sum;
   end
d[1]=phi[1][1];
for(j=2;j<=p;j++)
   v[j][1]=phi[j][1]/d[1];   /*calculate the multi-pulse v matrix */
for(j3=2;j3<=p-1;j3++)
   begin
      sum1=0.0;
      for(k=1;k<=j3-1;k++)
         sum1+=v[j3][k]*v[j3][k]*d[k];
      d[j3]=phi[j3][j3]-sum1;   /*calculate the multi-pulse d matrix */
      for(j=j3+1;j<=p;j++)
         begin
           sum2=0.0;
           for(k=1;k<=j3-1;k++)
             sum2+=v[j][k]*d[k]*v[j3][k];
           v[j][j3]=(phi[j][j3]-sum2)/d[j3];
         end
   end
sum3=0.0;
for(j=1;j<=p-1;j++)
   sum3+=v[p][j]*v[p][j]*d[j];
d[p]=phi[p][p]-sum3;
y[1]=psi[1];
for(j3=2;j3<=p;j3++)
   begin
      sum1=0.0;
```

```
      for(j=1;j<=j3-1;j++)   /*calculate the multi-pulse y matrix */
         sum1+=v[j3][j]*y[j];
      y[j3]=psi[j3]-sum1;
   end
alpha[p]=y[p]/d[p];
for(j3=p-1;j3>=1;j3--)
   begin
      sum2=0.0;
      for(j=j3+1;j<=p;j++)
         sum2+=v[j][j3]*alpha[j];   /*alphas for multi-pulse are finally */
      alpha[j3]=y[j3]/d[j3]-sum2;   /*defined in this loop */
   end
if(code==1) return;   /* if positions were specified you are done */
for(j=1;j<=n;j++)
   begin
      sum1=0.0;
      for(j3=1;j3<=p;j3++)   /* loop to calculate error function */
         sum1+=alpha[j3]*s[j+p-j3-1];
      e[j]=s[j+p-1]-sum1;
   end
for(k2=1;k2<=n;k2++)
   kp[k2]=1.0;
for(j=1;j<=i;j++)
   begin
      b[j]=0.0;
      for(k=1;k<=n;k++)
         begin
           if(fabs(*(e+k))>fabs(*(b+j))) then
              begin        /* loop to calculate the largest i values of */
                b[j]=e[k]; /* the error function which define the positions */
                pos[j]=k;  /* to be used in the next iteration */
              end
         end
      e[pos[j]]=0.0;
      kp[pos[j]]=0.0;
   end
end
end
}
```

9.3 Convergence of the Algorithm for the General Case

The assumption in the section 4.3 was that Φ' at each iteration of the

algorithm was invertible. There may be applications for the multi-pulse algorithm where the assumption is not a good one. This is especially true when a large number of pulses are used in a block of data. This section is concerned with the specific case where an eigenvalue of Φ' is found to be zero. By adding low-level white noise to the speech a new speech segment will be formed which results in a positive definite Φ'. Hence the algorithm will attempt to predict a new speech segment which is arbitrarily close to the original speech segment. The subscript n will denote that the speech segment is the noisy one. Suppose that $t(n)$ is a white-noise sequence and a new speech segment is formed as follows

$$s_n(n) \;=\; s(n) + t(n) \qquad n = 1, \ldots, N \tag{3.1}$$

For the original speech

$$\phi'(i,k) \;=\; \sum_{\substack{n=1 \\ n \neq p_1, \ldots, p_I}}^{N} s(n-i)\, s(n-k) \tag{3.2}$$

For the noisy speech

$$\phi'_n(i,k) \;=\; \sum_{\substack{n=1 \\ n \neq p_1, \ldots, p_I}}^{N} (s(n-i) + t(n-i))(s(n-k) + t(n-k)) \tag{3.3}$$

Equation (3.3) can be written as

$$\begin{aligned} \phi'_n(i,k) = \phi'(i,k) &+ \sum_{\substack{n=1 \\ n \neq p_1, \ldots, p_I}}^{N} t(n-i)s(n-k) + \sum_{\substack{n=1 \\ n \neq p_1, \ldots, p_I}}^{N} t(n-k)s(n-i) \\ &+ \sum_{\substack{n=1 \\ n \neq p_1, \ldots, p_I}}^{N} t(n-i)t(n-k) \end{aligned} \tag{3.4}$$

But

$$\sum_{\substack{n=1 \\ n\neq p_1,\ldots,p_I}}^{N} t(n-i)s(n-k) \;=\; 0 \tag{3.5}$$

and

$$\sum_{\substack{n=1 \\ n\neq p_1,\ldots,p_I}}^{N} t(n-k)s(n-i) \;=\; 0 \tag{3.6}$$

since the white noise is assumed to be uncorrelated with the speech.

Also

$$\sum_{\substack{n=1 \\ n\neq p_1,\ldots,p_I}}^{N} t(n-i)t(n-k) \;=\; \sigma\,\delta(i-k) \tag{3.7}$$

with

$$\sigma \;=\; \sum_{\substack{n=1 \\ n\neq p_1,\ldots,p_I}}^{N} t^2(n) \tag{3.8}$$

So equation (3.4) can be reduced to

$$\phi'_n(i,k) = \phi'(i,k) + \sigma\,\delta(i-k) \tag{3.8}$$

which implies

$$\Phi'_n = \Phi' + \sigma I \tag{3.9}$$

where I denotes a $p \times p$ identity matrix. If λ is an eigenvalue of Φ' then there exists an x such that

$$\Phi' x = \lambda x$$

and hence

$$\Phi'_n x = (\Phi' + \sigma I)x = \Phi' x + \sigma x = (\lambda + \sigma)x$$

so $(\lambda + \sigma)$ is an eigenvalue of Φ'_n. It is known that λ is greater than or equal to zero since Φ' is positive semi-definite. The variance σ is always greater than zero so the new matrix Φ'_n is positive definite and the results of the previous section can be used to approximate $s_n(n)$. If σ is very small then $s_n(n) \approx s(n)$. Hence, the parameters chosen to approximate $s_n(n)$ will also approximate $s(n)$. To use the above procedure it is not necessary to generate white noise and add it to the speech and from this proceed with the algorithm. Simply proceed with the algorithm as normal and if a zero eigenvalue is detected replace $\phi(i,i)$ with $(\phi(i,i) + \sigma)$ and continue. The addition of white noise to the speech simply shifts the eigenvalues of Φ' up by σ, the variance of the noise. Technically σ is not the true "variance" as the summation in equation (3.8) is not performed over all n. The above initial derivation was given so that one can understand that shifting the eigenvalues up by a constant is equivalent to adding white noise to the speech. The value of σ depends on the application.

This section has given a means of correcting a matrix Φ' which has a zero eigenvalue. By using this procedure the algorithm of section 4.3 can be used which is guaranteed to converge.