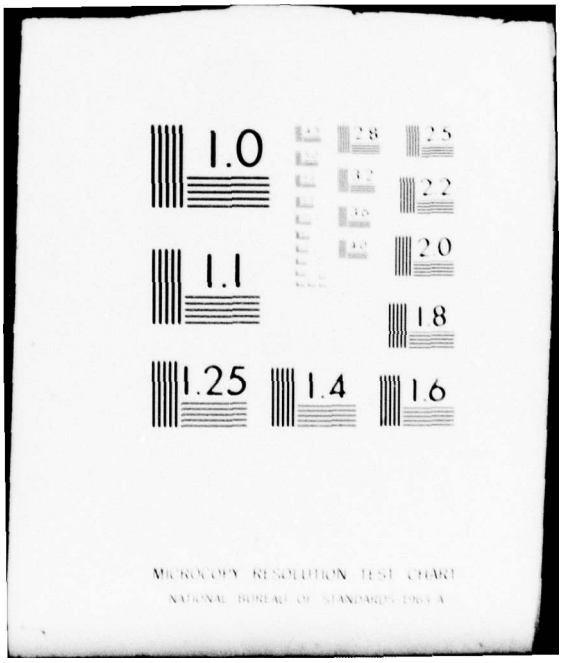
AD-A068 749 UTAH UNIV SALT LAKE CITY DEPT OF COMPUTER SCIENCE F/G 17/2
ESTIMATION OF THE PARAMETERS OF AN AUTOREGRESSIVE PROCESS IN TH--ETC(U)
DEC 78 W J DONE N00173-77-C-0041
UNCLASSIFIED UTEC-CSC-79-021 NL

1 OF 3
AD A068749


1.0
1.1
1.25
1.4
1.6
2.0
1.8
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

Estimation of the Parameters of an Autoregressive Process in the Presence of Additive White Noise

WILLIAM JOHN DONE

UNIVERSITY OF UTAH

LEVEL II

DECEMBER 1978
UTEC-CSc-79-021
COMPUTER SCIENCE, UNIVERSITY OF UTAH
SALT LAKE CITY, UTAH 84112

ON THE PARAMETERS OF AN AUTOREGRESSIVE PROCESS IN THE PRESENCE OF ADDITIVE WHITE NOISE
WILLIAM J

(6)

ESTIMATION OF THE PARAMETERS OF AN AUTOREGRESSIVE PROCESS IN THE PRESENCE OF ADDITIVE WHITE NOISE

(10) by

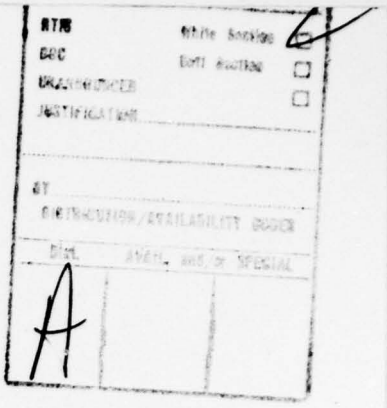
William John Done

(12) 220 p.

(11) December 1978

(14) UTEC-CSc-79-021

This research was supported by the Advanced Research Projects Agency of the Department of Defense under Contract No. (15) N00173-77-C-0041



ABSTRACT

Applications of linear prediction (LP) algorithms have been successful in modeling various physical processes. In the area of speech analysis this has resulted in the development of LP vocoders, devices used in digital speech communication systems. The LP algorithms used in speech and other areas are based on all-pole models for the signal being considered. With white noise excitation to the model, the all-pole LP model is equivalent to the autoregressive (AR) model.

With the success of this model for speech well established, the application of LP algorithms in noisy environments is being considered. Existing LP algorithms perform poorly in these conditions. Additive white noise severely effects the intelligibility and quality of speech after analysis by an LP vocoder.

autoregressive (AR)

It is known that the addition of white noise to an AR process produces data that can be described by an autoregressive moving-average (ARMA) model. The AR coefficients of the ARMA model are identical to the AR coefficients of the original AR process. This dissertation investigates the practicality of this model for estimating the coefficients of the original AR process. The

mathematical details for this model are reviewed. Those for the autocorrelation method (LP) algorithm are also discussed.

Experimental results obtained from several parameter estimation techniques are presented. These methods include the autocorrelation method for LP and a Newton-Raphson algorithm which estimates the ARMA parameters from the noisy data. These estimation methods are applied to several AR processes degraded by additive white noise. Results show that using an algorithm based on the ARMA model for the data improves the estimates for the original AR coefficients.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In the analysis of physical processes, one of the first steps taken is the development of a mathematical model which is representative of the process. Some examples of successful models for physical processes are those presently being used in the analysis of speech. One especially useful model is that based on all-pole linear prediction (LP). LP algorithms are important in both major areas of concern in digital speech communications:

1) high quality synthetic speech and
2) low bit rate communications systems.

Unfortunately, few physical processes can be measured without error. In many cases where measurement error is insignificant, the desired signal is corrupted by some other noise source. Since parameters of the model are to be inferred from the data, the estimation algorithm must be robust if it is to be useful in noisy situations. That is, the estimation algorithm should produce acceptable parameter estimates from data degraded by the types of noise expected in the system. This should be accomplished over a wide range of signal-to-noise ratios (SNR's). Most of the evaluations of LP algorithms, however, have been

performed with high quality speech inputs having minimal background noise. When noise is added to a speech signal prior to analysis, the intelligibility and quality of the synthetic speech generated by the LP system are degraded. The addition of noise causes problems in four areas:

1) silence detection,

2) voiced/unvoiced determination,

3) pitch period calculation if voiced, and

4) identification of the LP coefficients.

McAulay [26] has addressed the first three problems. This research is concerned with problem 4), the identification of the coefficients of the all-pole model when the degradation is due to additive white noise.

With white noise excitation, the all-pole LP model for speech is identical to the autoregressive (AR) model discussed in many texts. The research presented here specifically deals with a model for an AR process plus white noise. The data resulting from the addition of white noise to an AR process is an autoregressive moving-average (ARMA) process. The moving-average (MA) component of the model is equivalent to an all-zero specification for a system. In this dissertation, the model for an AR process plus white noise will be referred to as the AR-to-ARMA transformation model. A detailed description of this model is presented in Chapter 3. The most significant feature of this model emphasized here is the following: if the AR

process to be identified is degraded by additive white noise, the AR parameters of the data are identical to those of the original AR model.

The addition of the white noise introduces MA parameters. Parameter estimation methods must take into account the presence of the MA characteristics of the data. Intuitively, an analysis system based on the ARMA model should be more robust in white noise environments than linear predictive coding (LPC) systems, which deal only with AR parameters. This robustness arises because the model explicitly accounts for this kind of noise. ARMA estimation procedures, however, are more difficult to implement than AR estimation methods. The MA portion of the model introduces nonlinear relationships. Solutions usually involve iterative schemes. Also, use of these methods requires significant modifications of the LP analysis procedure, even though the AR parameters are the goal of each method.

The primary objective of this research is to determine the applicability of the AR-to-ARMA transformation model in estimating the parameters of the desired AR process. Intuitively, algorithms based on this model should perform better in white noise environments than the LP algorithm, which ignores the MA component of the data. The possible benefit of parameter estimation procedures derived from the AR-to-ARMA transformation is improved operating

characteristics of the system in white noise conditions. Specifically, the objectives of this research are:

1) Illustrate the effect of additive white noise on the AR coefficient estimates produced by the autocorrelation method of LPC.
2) Test several estimation procedures based on the AR-to-ARMA transformation on data generated from known ARMA models.
3) Apply the most promising methods to data generated by adding white noise to known AR models. This will be done for several AR models over a wide range of SNR's.
4) Compare the results of 3) with those obtained from LPC.
5) Identify areas for future work.

There are some restrictions placed on the scope of this work. These qualifications are made to reduce the complexity of the model for the noisy data and to emphasize the estimation of the LP coefficients. First, only additive white noise will be considered. In that case, the AR coefficients of the data are identical to the AR coefficients of the desired AR signal. If the noise is non-white but can be described by an ARMA model, the AR coefficients of the data are no longer equal to those of the original AR process. An additional estimation stage, based on nonlinear relationships, would be required for the non-white noise environment.

Second, if the original AR process is of order q, that

is, there are q AR coefficients, it is assumed that q is known for this AR(q) model. Otherwise, q must be estimated from the noisy data along with the coefficients. The emphasis here is meant to be on the estimation of the coefficients.

The third restriction is that q will be limited in value to four or less for most tests. The parameter q is restricted to these small values because the variance of the parameter estimates tends to increase as the number of parameters in the model increases. Also, the computational requirements for some of the ARMA estimation algorithms are large, resulting in long experiments on the general purpose computer used in this research. Demonstration of the performance of estimation algorithms based on the transformation model for these low order processes should be sufficient to indicate the advantages and disadvantages of that approach.

Finally, as stated in objectives 2) and 3) above, tests are performed on known ARMA models. This implies that all data analyzed are synthetic in the sense that all processes are generated from specified models using approximately white noise sequences as the excitation. This has definite advantages over tests performed on data from unknown ARMA models. First, there is the confidence that the data actually comes from an ARMA process. Second, the parameter estimates can be compared directly with the

parameters of the generating model. Third, experimental specifications such as SNR's can be given with greater certainty. Also, in those algorithms requiring initial estimates for the parameters, the parameter values of the generating model can be used. This removes any uncertainties due to initial estimates from the experiments, which are primarily concerned with identifying the AR parameters using the AR-to-ARMA transformation model. It must also be stated that the computational requirements of various algorithms are not considered in this research. No algorithm is dismissed simply because it requires a higher computational load than algorithms already in common use.

Data presented in this dissertation show the degradation in the LPC parameter estimates that results from adding increasing levels of white noise to an example frame of speech. Several estimation procedures are then applied to noisy data generated from known AR models. The results for the AR(1), AR(2), and AR(4) processes analyzed show that two of the estimation methods tested yield AR parameter estimates that are better than those obtainable from the autocorrelation method of LPC. The improvement in the estimates is evident at SNR's through 0 dB. One of the methods is a Newton-Raphson implementation of a conditional maximum likelihood technique. This procedure simultaneously estimates both the AR and MA parameters from

the data. The second method is similar to LPC in the type of operations involved, but takes into account the MA component of the data, where the LPC method does not. The estimates produced by these two algorithms demonstrate the validity of the AR-to-ARMA transformation model. These algorithms are less susceptible to white noise degradation than LPC and are thus more robust estimation procedures. The comparisons of these estimation algorithms and the demonstration of the practicality of the transformation model are the primary contributions of this research.

Chapter 2 presents the results of a literature search into the topic of estimating the parameters of AR processes in the presence of noise. References for discussions of LPC algorithms and the AR-to-ARMA transformation model are given. Several sources for parameter estimation algorithms are also provided. The mathematical details for the AR process plus white noise are given in Chapter 3. The LPC estimation algorithm is discussed, as are those algorithms which take advantage of the transformation model. Chapter 4 contains descriptions of the various experiments performed and the data obtained from those tests. A summary of the work performed and the conclusions derived from this research are presented in Chapter 5. That chapter also lists areas for future work. Several appendixes provide detailed explanations of some of the material in Chapter 3.

CHAPTER 2

LITERATURE REVIEW

Introduction

The use of stochastic models for the analysis of discrete time domain series is important in many areas of interest. Examples of these applications include analysis of economic time series, seismic data, and more recently, discrete speech waveforms. The reader is referred to references given in Makhoul [23] and Box and Jenkins [10] as sources for discussions on the theory of time series analysis and possible applications. In a paper published in 1971 [3], Atal and Hanauer describe a system which models speech as an autoregressive process. Generation of a synthetic speech sequence from the AR parameters is proposed in that paper. This caused much activity in applying the method of time series analysis to speech and eventually resulted in the development of linear prediction vocoders, devices designed to apply LP algorithms to the analysis of speech. Linear prediction, the expression most commonly used in speech analysis to describe AR modeling, is quite successful in its application to discrete speech waveforms. As pointed out in Chapter 1, however, the presence of noise in signals analyzed using LP algorithms

has a detrimental effect on their performance.

A brief discussion of the LPC technique for speech analysis is presented to describe previous efforts at improving the operation of linear prediction when noisy speech must be used. Given a sequence of speech samples $s(k)$, $k = 0,\ldots, N-1$, estimates of the autocorrelation function $R_{ss}(k)$ of $s(k)$ are obtained from

$$R_{ss}(k) = \sum_{i=0}^{N-1-k} s(i)\, s(i+k) \ , \tag{2.1}$$

for $k = 0, \ldots, q$. In (2.1) q is the order of the AR process for the speech modeled by

$$s(k) = - \sum_{i=1}^{q} a(i)\, s(k-i) + \varepsilon(k) \ , \tag{2.2}$$

with the $\{a(i)\}_1^q$ the AR parameters and $\varepsilon(k)$ a white noise process. Estimates for the $\{a(i)\}_1^q$ are obtained by solving the Yule-Walker equations

$$\sum_{i=1}^{q} a(i)\, R_{ss}(i-k) = -R_{ss}(k) \ , \tag{2.3}$$

$k = 1, \ldots, q$. Expression (2.3) represents a system of q equations with q unknowns. The estimates $\{a(i)\}_1^q$--along with gain, pitch period, and voiced/unvoiced estimates--are used to construct a synthetic speech waveform. This brief development is based on what is commonly referred to as the autocorrelation method of LPC speech analysis [23]. In that method $s(k)$ is usually windowed prior to analysis. The primary aspect of this procedure that should be noted

is the need to estimate the autocorrelation function $R_{ss}(k)$ from the data $\{s(k)\}_0^{N-1}$.

The above discussion of LPC analysis will clarify the presentation of several approaches to parameter estimation in the presence of noise. These methods attempt to correct the autocorrelation function of the noisy data so that (2.3) might be used to estimate the AR parameters. The following part also contains summaries of work that has described and quantified the degradation caused by additive noise in LP systems. The reader is next referred to several sources describing the AR-to-ARMA transformation model. Using algorithms based on this model requires the identification of the parameters of autoregressive moving-average processes with respective orders of q and p, ARMA(q,p). This dissertation will develop the ideas presented in this latter modeling technique. Several papers concerning possible nonlinear estimation procedures will be reviewed in a section on parameter estimation. That section also lists algorithms that are applicable to ARMA parameter estimation. The reader is then referred to three previous works which deal with the estimation of parameters from MA(1) and ARMA(1,1) processes. The reader is also referred to sources for discussions of the pre-filtering approach to noise suppression.

Linear Prediction Literature

In 1976 at the IEEE International Conference on Acoustics, Speech, and Signal Processing, Yegnanarayana [42] reported on the effects of noise and distortion in parameter estimation in speech signals. Two topics from that report important to this research concern the possible distortion introduced by pre-filtering and four possible methods for dealing with additive noise. The pre-filtering referred to is that which is necessary to avoid aliasing prior to digitization of the speech signal. If the anti-aliasing filter introduces a sharp roll-off at the Nyquist frequency, this tends to increase the possibility of ill-conditioning in the autocorrelation matrix used in the Yule-Walker equations (2.3). This also holds for pre-filters meant to suppress the noise, a disadvantage that might occur with a pre-filtering approach to parameter estimation.

The four possible procedures to compensate for additive noise given (and criticized) by Yegnanarayana are as follows:

1) Correct the short time power spectrum of the observed data $x(k)$ by subtracting the power spectrum of the noise. The problem with this approach is that the short time power spectrum of the noise, containing random variations, may not be cancelled by subtracting the average noise power spectrum.

2) Whiten the noise component by pre-filtering. The distortions possible from pre-filtering have already been discussed.

3) Extract the parameters by analyzing only those sections of the spectrum corresponding to a high SNR (as a function of frequency). This technique introduces the more complicated selective linear prediction analysis method [24], requiring modification of the parameter extraction stage, and fails to use information about the AR process contained in those frequency ranges that are ignored.

4) Noise effects can be reduced by using a second order filter discussed in [18]. This filter, based on the first two autocorrelation coefficients, would correct only the gross spectral distortions of the noise.

At the same IEEE conference in 1976, Sambur and Jayant [33] presented preliminary results on the effects of white noise and differentially quantized speech on LPC synthesis procedures. To measure the distortion caused by inaccurate identification of the AR coefficients, the authors used a distance measure proposed by Itakura [18]. This metric is said to indicate where spectral matching errors, which occur because of failure to identify the AR parameters, begin to be statistically or perceptually significant. In [33] and [34], Sambur and Jayant indicate that the degradation resulting from white noise is more severe than

that resulting from the two types of differentially quantized speech. Their results for white noise degradation also illustrate that perceptually significant variations occur at a signal-to-noise ratio of about 28 dB. The signal-to-noise ratio is defined as $\sum s^2(k) / \sum n^2(k)$, the summation being over the entire duration of the speech input.

The brief development of LPC given in the introduction to this section indicates the possible approach of correcting the autocorrelation function of the input data so that it matches $R_{ss}(k)$, the autocorrelation function of the signal $s(k)$. If the noisy data $x(k)$ is given by

$$x(k) = s(k) + n(k) \ , \tag{2.4}$$

then the autocorrelations of $s(k)$ and $x(k)$ are related by

$$R_{ss}(k) = R_{xx}(k) + R_{nn}(k) - R_{xn}(k) - R_{nx}(k) \tag{2.5}$$

$$= R_{xx}(k) - R_{nn}(k) - R_{sn}(k) - R_{ns}(k) \ . \tag{2.6}$$

Then if estimates for the noise autocorrelation and signal-noise cross-correlation are available, the autocorrelation $R_{ss}(k)$ of the original signal may be estimated. This approach is appealing, since the standard LPC algorithms can be used once $R_{ss}(k)$ has been obtained by some additional operations.

In the development of a word spotting system based on the calculation of LPC parameters, Christiansen [11] proposed the following approach as one possible method of

dealing with noisy speech. If $\hat{R}_{ss}(k)$ indicates the approximation for $R_{ss}(k)$ that will be used in (2.3) to obtain the AR coefficients, a reasonable expression for $\hat{R}_{ss}(k)$ might be

$$\hat{R}_{ss}(k) = R_{xx}(k) - R_{nn}(k) \ . \tag{2.7}$$

Equation (2.7) derives from (2.6) with the following assumptions:

1) $R_{nn}(k)$ is obtained by averaging the autocorrelation function over intervals containing no speech activity;
2) $s(k)$ and $n(k)$ are uncorrelated, that is, $R_{sn}(k) = R_{ns}(k) = 0$ for all k.

This approach did not work in the word spotting system of [11]. Results indicate that the LPC algorithms failed, due to violation of assumptions 1) and 2) above. The effect of these assumptions is illustrated in Chapter 4.

Atashroo [4] proposed a system for handling noisy speech that combines the pre-filtering and modeling approaches. Using a noncausal formulation, the transfer function for the Wiener filter $H(\omega)$ is

$$H(\omega) = \frac{\Phi_{sx}(\omega)}{\Phi_{xx}(\omega)} = \frac{\Phi_{ss}(\omega)}{\Phi_{ss}(\omega) + \Phi_{nn}(\omega)}$$

$$= 1 - \frac{\Phi_{nn}(\omega)}{\Phi_{xx}(\omega)} \ , \tag{2.8}$$

where $\Phi(\omega)$ indicates the power spectrum of the subscripted quantity. The power spectrum of the output, $\hat{\Phi}_{ss}(\omega)$, is

$$\hat{\Phi}_{ss}(\omega) = \Phi_{xx}(\omega)\ |H(\omega)|^2 = \Phi_{xx}(\omega)\ \left[1 - \frac{\Phi_{nn}(\omega)}{\Phi_{xx}(\omega)}\right]^2 . \quad (2.9)$$

$\Phi_{xx}(\omega)$ is estimated by averaging the spectra of short overlapping segments of data. $\Phi_{nn}(\omega)$ is estimated in like manner from speechless intervals. Once $\hat{\Phi}_{ss}(\omega)$ is computed from (2.9), its inverse Fourier transform will yield $\hat{R}_{ss}(k)$, which can be used in (2.3) to obtain estimates for the AR parameters. Note that (2.8) is obtained by assuming s(k) and n(k) are uncorrelated. Atashroo does not quantify the improvement possible with this method.

Common to the two previous techniques is the assumption that s(k) and n(k) are uncorrelated. Boll, in a system referred to as Predictive Noise Cancellation (PNC) [9], describes a system designed to approximate $R_{ss}(k)$ by estimating all of the terms on the right hand side of (2.5). PNC attempts to estimate these auto- and cross-correlation terms by filtering a secondary noise channel, n(k). The input for this channel is derived by averaging the characteristics of the noise when there is no speech activity. The filter H(z) is designed to minimize the error between its output u(k) and x(k), the noisy data. The method can be summarized in four statements:

1) Estimate the background noise n(k) and the noise characteristics during speechless intervals.
2) Estimate the noise-signal correlation filter H(z).
3) Modify the noisy speech autocorrelation function $R_{xx}(k)$

to obtain $\hat{R}_{ss}(k)$.

4) Calculate the final AR parameters using equation (2.3) with $\hat{R}_{ss}(k)$ replacing $R_{ss}(k)$.

Boll claims an improvement in SNR of 10 dB with this approach. The similarity of the PNC system to adaptive noise cancelling (ANC) systems should be noted [41].

In a recent paper [22], Lim and Oppenheim present four methods for estimating the parameters of an all-pole (AR) system degraded by additive white noise. The methods differ in the assumptions made about initial conditions for the parameters, data, and gain. Two of the methods are shown to be equivalent to the covariance and autocorrelation methods of LPC when there is no additive noise. When considering the noiseless case, three of the four methods result in linear operations in the estimation procedure, while the fourth method involves nonlinear relationships. When white noise is added to the desired signal, all of the procedures require the solution of nonlinear equations in the estimation stage. The authors propose two suboptimal methods involving only linear operations. Both methods are iterative and involve filtering the data to estimate the original all-pole signal. This is followed by an LPC estimation step to provide new estimates for the model parameters. The filtering--LPC process is repeated for each iteration. In one method, the filter used is a noncausal Wiener filter.

Results for synthetic data and speech data, at several SNR's, are presented.

ARMA Model Literature

The preceding discussions cover four possible approaches to extracting the AR parameters of a signal corrupted by noise. The three autocorrelation modification techniques qualify as modeling approaches in the sense that the relationships of (2.5) and (2.6) are used to obtain an estimate of $R_{ss}(k)$. The fourth technique, the iterative approach proposed by Lim and Oppenheim, is representative of the filtering approach to noise removal. If the additive noise is white, it is possible to use another model description. Walker [39] presents a discussion of the consequences of additive noise when analyzing time series. He points out that if s(k) is an AR(q) process

$$\sum_{i=0}^{q} a(i)\, s(k-i) = \varepsilon(k) , \tag{2.10}$$

a(0) = 1.0, and x(k) = s(k) + n(k) is the corresponding noisy process, the combination of these two equations gives

$$\sum_{i=0}^{q} a(i)\, x(k-i) = \sum_{i=0}^{q} a(i)\, n(k-i) + \varepsilon(k) \triangleq y(k) . \tag{2.11}$$

The autocorrelation function of y(k) is now a function of both the $\{a(i)\}_1^q$ and the additive noise, complicating the task of estimating the desired parameters of the system, the $\{a(i)\}_1^q$. It is Walker's belief that the laborious and

uninteresting calculations involved may be the reason for the neglect of this approach in time series analysis.

In their text on time series analysis [10], Box and Jenkins briefly discuss the effects of noise added to a general ARMA(q,p) process. If white, the noise will change the MA parameters only, leaving the AR parameters unchanged. The new time series is ARMA(q,r), where r = max(p,q). Thus, if the original process is AR(q), that is p = 0, then the new process created by adding white noise is ARMA(q,q). Box and Jenkins also discuss the effects of non-white additive noise. In that case the AR parameters are also changed.

The most extensive discussion to date on the development of this type of noise model for AR processes is due to Pagano [30]. He presents the extension of an AR(q) process to an ARMA(q,q) process as a result of the additive white noise. He also shows that the new process is actually an ARMA(q,q) process, not one in which the orders are less than q as a result of cancellation of factors from the AR and MA operators. Pagano then develops the nonlinear relationships, mentioned by Walker [39], between the $\{a(i)\}_1^q$, the SNR, and $R_{yy}(k)$, the autocorrelation function of the sequence y(k) defined by Walker in (2.11). Finally, he proposes a nonlinear regression technique through which estimates of the $\{a(i)\}_1^q$ can be obtained by taking advantage of the nonlinear relationships discussed

by Walker.

In a paper reviewing the applications of time series analysis [31], Parzen points out the necessary steps in applying the techniques available in time series analysis. One of the first steps is model conception, that is, selecting the model which is appropriate to the data being observed. As an example of this step, Parzen points out the possible use of the ARMA model for an AR process degraded by additive white noise, discussed by Pagano.

Tong [37] makes use of the extension of an AR(q) model to an ARMA(q,q) model when white noise is added in a procedure devised to aid in determining the order of an AR process corrupted by noise. He extends those results in a later paper [38] to special cases of additive noise that is correlated to the signal represented by the AR model.

Parameter Estimation Literature

The procedures proposed by the above sources and presented in detail by Pagano [30] at some point require the estimation of the parameters of an ARMA process, which is inherently more difficult than the estimation of AR coefficients. However, much work has been done on techniques for extracting ARMA parameters from time series. The techniques available include methods based on nonlinear operations and methods comprised of only linear operations. The first half of this section summarizes several papers which present algorithms for estimating the parameters of

an ARMA process. The second half of this section deals with parameter estimation techniques (not necessarily limited to ARMA processes) and modifications made to estimation procedures to improve convergence.

A presentation by Anderson [2] of ARMA parameter estimation algorithms based on the conditional maximum likelihood optimization of the normal likelihood function is one of the most thorough treatments of the subject. Anderson develops a matrix notation which facilitates writing the equations involved in the estimation. Algorithms are then developed along these divisions:

1) time domain versus frequency domain;
2) Newton-Raphson method versus the method of scoring (Gauss-Newton method);
3) parameter set 1 (AR coefficients, MA coefficients, and excitation sequence variance) versus parameter set 2 (AR coefficients and MA covariances).

After presenting the algorithms based on these eight possibilities, Anderson then briefly compares the methods and discusses some results found from Monte Carlo studies performed by other researchers. The contents of this paper are particularly useful in interpreting the matrix formulations for these algorithms common in the statistical literature.

Hannan [15] presents a three step procedure for estimating the parameters of an ARMA process. Spectral

factorization may be used in the estimation of the MA parameters, but if the requirements needed for factorization are not present, an alternative procedure is given. Hannan's technique, even though it produces asymptotically efficient estimates of the ARMA parameters, can be further modified to form an iterative procedure for improving the estimates of the ARMA parameters. Akaike [1] points out that Hannan's method is equivalent to a one-step Newton-Raphson iterative procedure for modifying the initial estimates to maximize the Gaussian likelihood function. The main limitation of the procedure, in Akaike's opinion, is the possible failure of the technique to improve the estimates because of poor initial estimates.

Another procedure for estimating ARMA parameters from a time series is given by Graupe, Krause, and Moore [13], which requires three steps involving only the solution of linear equations. The procedure is initiated by identifying the parameters of an equivalent AR(∞) process. Even though an infinite number of AR parameters is required to represent an ARMA(q,p) process in general, it is claimed that only a small number of these are necessary for the computation. From these initial AR(∞) parameters, two steps involving linear operations are required to obtain first the MA and then the AR parameters.

A fourth possibility for estimating the parameters of an ARMA process is represented in the approach given by

Steiglitz in [35]. Presented as a method for estimating the poles and zeros of the vocal tract, the procedure uses an algorithm developed for linear system identification [36]. This method requires knowledge of the input and output (possibly noisy) of the system. An iterative technique, using only linear operations, simultaneously estimates the coefficients of the pole and zero filters. This can be used directly on the signal, or on a minimum phase representation of the signal, obtained from homomorphic filtering.

Kashyap and Nasburg [21] review several methods for estimating the parameters of multivariate autoregressive moving-average processes, including discussions of least squares methods and maximum likelihood methods. The authors also discuss numerical methods that might be used to obtain the parameter estimates. The Newton-Raphson (NR) method is discussed, but convergence problems that might be associated with this procedure are handled by using different initial guesses to start the algorithm. In this same paper Kashyap and Nasburg also review the algorithms developed by Durbin [12] and Walker [40]. It is stated that these methods are applicable in the univariate case, but may produce parameter estimates of questionable efficiency. Numerical results for various estimators are provided for one MA(1) process and one ARMA(1,1) process.

Using a state vector formulation, Gupta and Mehra [14]

discuss the numerical aspects of maximum likelihood estimates. Use of the NR method is discouraged, primarily because of convergence problems and computational drawbacks. The Gauss-Newton (GN) method (method of scoring) is stated to have somewhat better convergence properties. Several other numerical methods for parameter estimation are given, including a modified GN procedure and some suggestions for reducing the computational load.

Comparisons of several gradient methods for obtaining estimates of parameters involving nonlinear relationships are presented by Bard [5]. Gradient methods are of the form

$$\underline{\theta}_{i+1} = \underline{\theta}_i - \rho_i \underline{R}_i^{-1} \underline{g}_i \ . \tag{2.12}$$

In (2.12) $\underline{\theta}_i$ and $\underline{\theta}_{i+1}$ are the values of the parameter vector at the i^{th} and $i+1^{th}$ iterations, respectively, with $\underline{\theta}_i$ known. The vector $\underline{g}_i$ is the gradient of the cost function (e.g., likelihood, least squares), evaluated at $\underline{\theta}_i$. $\underline{R}_i$ is a matrix and ρ_i is a scalar, each evaluated at $\underline{\theta}_i$. The various gradient methods are characterized by the form of the matrix $\underline{R}_i$ and the strategy by which ρ_i is chosen. If $Q(\underline{\theta}_i)$ is the value of the cost function $Q(\underline{\theta})$ at the i^{th} iterate, then the estimation procedure seeks to find $\underline{\theta}_{i+1}$ such that

$$Q(\underline{\theta}_{i+1}) > Q(\underline{\theta}_i) \ , \tag{2.13}$$

if it is desired to maximize the cost function. As pointed

out by Bard, $\underline{R}_i$ should be positive definite to ensure (2.13) holds for $\underline{g}_i \neq 0$ and $\rho_i > 0$. Also, $\underline{R}_i$ is usually an approximation to $\underline{H}_i$, the Hessian of $Q(\underline{\theta})$ evaluated at $\underline{\theta} = \underline{\theta}_i$. The Hessian of $Q(\underline{\theta})$ is defined as the matrix of second partial derivatives of $Q(\underline{\theta})$ with respect to the elements of $\underline{\theta}$. If $\underline{R}_i = \underline{H}_i$ and $\rho_i = 1$, then one has the Newton-Raphson method. After evaluating $g(\underline{\theta})$ and $\underline{R}(\underline{\theta})$ at $\underline{\theta} = \underline{\theta}_i$, ρ_i is selected so that (2.13) is true. Bard then proceeds to describe several estimation procedures based on different choices for $\underline{R}_i$ and ρ_i. Using several of the most successful techniques, the author demonstrates their application to typical estimation problems and discusses the capabilities of the methods.

In his text on nonlinear parameter estimation [6], Bard describes various algorithms used to optimize some cost function of the parameters. He points out that the Newton-Raphson method is the only method which will reach the extremum in one iteration when the cost function is quadratic. Based on the one step convergence of the NR method for a quadratic function, Bard gives convergence rate efficiency factors for the various methods [6, p. 89]. For the NR method this factor is one, but the method may suffer from convergence problems. Bard also discusses methods for terminating estimation algorithms [6, p. 114].

Another survey of numerical techniques for optimizing a cost function is presented by Powell [32]. He discusses

steepest descent, direct search, and conjugate direction methods. Included in his discussion is a presentation of the NR method, which he states is still useful in many applications. Powell mentions that the most serious disadvantage of the NR method to many users is the need to evaluate the second derivatives of the cost function. Many of the techniques described by Powell have been developed to achieve fast convergence without explicitly evaluating the second derivatives. Powell gives recommendations for selecting an algorithm to optimize a given cost function. The suggestions are roughly based on the number of parameters, the availability of derivatives of the cost function, and whether or not the user wishes to evaluate the derivatives.

One of the techniques discussed in most of the preceding sources is attributed to Marquardt [25]. A disadvantage of the GN and NR methods is that the algorithms may fail to converge to the optimal solution if the initial guess does not fall into a small enough neighborhood of that solution. A criticism of some gradient methods is that, while the region of convergence is larger than that of the NR or GN methods, the rate of convergence is slower. Marquardt's method claims to combine the faster convergence of the GN method (when near the optimal solution) with the larger region of convergence for the gradient methods. If the iterative step in the GN

method is given by

$$\underline{\theta}_{i+1} = \underline{\theta}_i - \underline{R}^{-1}(\underline{\theta}_i)\, \underline{g}(\underline{\theta}_i) , \tag{2.14}$$

then Marquardt's method is given by

$$\underline{\theta}_{i+1} = \underline{\theta}_i - [\underline{R}(\underline{\theta}_i) + \lambda_i\, \underline{I}]^{-1} \underline{g}(\underline{\theta}_i) . \tag{2.15}$$

The scalar λ_i is automatically selected by the algorithm to ensure that $Q(\underline{\theta}_{i+1}) > Q(\underline{\theta}_i)$ when maximizing $Q(\underline{\theta})$. In (2.15) $g(\underline{\theta})$ is again the gradient of $Q(\underline{\theta})$, $\underline{R}(\underline{\theta})$ is a matrix and $\underline{I}$ is the identity matrix. As $\underline{\theta}_i$ approaches the optimal solution λ_i tends toward zero and the algorithm behaves like the GN method. However, if $\underline{\theta}_i$ is far from the optimal solution, λ_i will tend to be larger. When the $\lambda_i\, \underline{I}$ term dominates, then

$$\underline{\theta}_{i+1} \simeq \underline{\theta}_i - \lambda_i^{-1}\, \underline{g}(\underline{\theta}_i) , \tag{2.16}$$

which is the simplest expression for the gradient method. Marquardt's method is often recommended for nonlinear estimation problems [5], [32]. A disadvantage of this method is the increase in computations incurred by enclosing the iterative Marquardt method inside the iterative GN method.

Two useful texts on nonlinear parameter estimation are Ortega and Rheinboldt [29] and Beck and Arnold [7]. The text by Ortega and Rheinboldt provides extensive coverage of iterative methods for solving nonlinear parameter estimation methods, including the NR and GN methods, conjugate-direction methods, and gradient methods.

Convergence properties for the various techniques are discussed.

In Chapter 7 of their text, Beck and Arnold [7] describe several methods that might be used in parameter estimation. After presenting the GN method, the authors discuss several modifications of the GN method. These include Marquardt's method and the Box-Kanemasu interpolation method. The latter method is an algorithm which uses quadratic interpolation to select the parameter ρ_i in (2.12) such that (2.13) holds (if maximizing). Beck and Arnold present examples of some of the methods and compare their findings to those of Bard [5], [6].

A collection of papers on numerical techniques for unconstrained optimization, edited by Murray, is available in [27]. The papers contributed for this book cover in depth parameter estimation techniques that include direct search, conjugate-direction, quasi-Newton, and second derivative methods. Included in the topics is one paper on the problems related to optimization. There is also a discussion of the failures that can occur with any of the methods presented, causes of these failures, and what can be done to correct them.

The last three papers to be reviewed in this section on parameter estimation are all discussions of the Gauss-Newton and modified GN method of solving for the parameters to optimize a nonlinear function. Hartley [16]

and Hartley and Booker [17] discuss the optimization of the cost function $Q(\underline{\theta})$,

$$Q(\underline{\theta}) = \sum_{i=1}^{M} [z_i - f_i(\underline{\theta})]^2 \ . \qquad (2.17)$$

$\underline{\theta}$ is the parameter vector, the z_i are the known data values, and the $f_i(\underline{\theta})$, i = 1, ..., M, are M known functions mapping from the $\underline{\theta}$ parameter space to the observed data, z_i. These papers are useful in light of the formulation used by Pagano [30] to derive the nonlinear relationship between the AR model and the ARMA model resulting from the addition of white noise to the AR process. Jennrich [19] describes a modification of the GN method that may be useful in this type of work. The details for the GN and modified GN methods are included in Appendix B.

Previous Work on ARMA Estimation

This section presents several sources that provide information about the practical aspects of ARMA parameter estimation. In all cases, the work has been performed on small order processes, usually no more than second order. Box and Jenkins [10] provide the researcher with a thorough background in time series analysis as applied to ARMA modeling. Especially useful are the developments for the variances of parameter estimates.

There are two papers in which MA(1) and ARMA(1,1) models have been studied. Nelson [28] uses Monte Carlo methods to test several types of estimators on MA(1)

processes. The MA(1) process has a single MA parameter, b. Nelson's work considered the operation of the selected estimators for processes where b = $\pm$0.2, $\pm$0.5, and $\pm$0.8. One of Nelson's most interesting findings is the tendency for the maximum likelihood methods to perform best for the MA(1) processes with b of moderate magnitude, that is, b close to 0.5 in magnitude. Kashyap and Nasburg [21] use one MA(1) process and one ARMA(1,1) process to demonstrate some of the techniques presented in their paper on estimation methods. Anderson [2] discusses the findings of Nelson [28] and Kashyap and Nasburg [21]. The results of Nelson's work will be reviewed in more detail in Chapter 4.

For the benefit of the reader, two references to alternative techniques for extracting the parameters of a model from noisy data are given. Widrow, et. al., discuss the method of adaptive noise canceling [41], and Kailath [20] presents an overview of linear filtering theory. The bibliography of the latter is extensive and gives many references to topics in Wiener filtering and recursive Wiener and Kalman filtering. The approach taken when using noise suppression methods in conjunction with these two classes of estimation procedures is to restore the signal prior to estimating the parameters. The algorithms of the parameter estimation stage are then likely be unchanged from the algorithms used in the noiseless case.

CHAPTER 3

THEORY

Introduction

In this chapter the details of pertinent theory are presented. A review of linear prediction is given. Included in the review is a development showing the effects of white noise on the LP parameters determined by the linear prediction algorithm. Presented in a matrix formulation, the LPC algorithm discussed is that commonly referred to as the autocorrelation method. The LPC discussion is followed by the details of the AR-to-ARMA transformation model. In that section the generation of an ARMA(q,q) process from the addition of white noise to an AR(q) process is demonstrated, followed by a section on the first order AR process corrupted by white noise. This section is valuable because the low order of the model allows one to examine in detail the effects of adding the white noise to the AR(1) process. Many of the results in the next chapter are based on the analysis of this first order case. Succeeding sections discuss five parameter estimation methods that are considered as means to extract estimates of the AR parameters from the data. Following that is a discussion of the noncausal formulation for the

Wiener filter. A brief presentation of the nonlinear regression algorithm suggested by Pagano [30] is then given. The last section presents details on the noise sequences used in this work as the excitation sequences for the ARMA models and the additive noise sequence.

Linear Predictive Coding

If s(k) is a time series which can be modeled as a q^{th}-order autoregressive process, AR(q), then we have

$$s(k) = -\sum_{i=1}^{q} a_1(i)\, s(k-i) + \varepsilon(k) \ , \qquad (3.1)$$

$a_1(0) = 1.0$. The $\{a_1(i)\}_1^q$ are the AR parameters and $\varepsilon(k)$ is a zero mean white noise process. The formulation of the AR(q) process in (3.1) is identical to the q^{th}-order all-pole LP model. In the autocorrelation method of LPC analysis, the equations are much more compact if matrix notation is used. Refer to Makhoul [23] for additional background and a list of references for LPC development. The development of a notational convention for LPC using a matrix formulation can be found in Boll [8].

Using the autocorrelation method, the sequence s(k) has infinite extent but is nonzero only for $0 \leq k \leq N-1$, where N is the size of the analysis window. Form the (N+q) x 1 vector $\underline{s}$, where $\underline{s}$ is given by

$$\underline{s} = [s(0) \quad s(1) \cdots s(N-1) \quad 0 \cdots 0]^T \ . \qquad (3.2)$$

Using D as a delay operator for vector notation, $D^i\underline{s}$ is an

(N+q) x 1 vector with the sequence s(0) ... s(N-1) beginning at the $(i+1)^{th}$ position. The superscript i can take on the values 1, ..., q. For example,

$$D^1\underline{s} = [0 \quad s(0) \quad s(1) \cdots s(N-1) \quad 0 \cdots 0]^T ,$$

$$D^2\underline{s} = [0 \quad 0 \quad s(0) \quad s(1) \cdots s(N-1) \quad 0 \cdots 0]^T , \text{ and}$$

$$D^q\underline{s} = [0 \quad 0 \cdots 0 \quad s(0) \quad s(1) \cdots s(N-1)]^T .$$

Form the (N+q) x q matrix $\underline{H}_s$ by including as columns the $D^i\underline{s}$, i = 1, ..., q,

$$\underline{H}_s = [D^1\underline{s} \quad D^2\underline{s} \quad D^3\underline{s} \cdots D^q\underline{s}] . \tag{3.3}$$

If an error sequence $\underline{\varepsilon}$ is defined as

$$\underline{\varepsilon} = [\varepsilon(0) \quad \varepsilon(1) \cdots \varepsilon(N-1) \cdots \varepsilon(N+q-1)]^T , \tag{3.4}$$

then (3.1) can be written as

$$\underline{s} = -\underline{H}_s \, \underline{a}_1 + \underline{\varepsilon} . \tag{3.5}$$

The vector $\underline{a}_1 = [a_1(1) \quad a_1(2) \quad \ldots \quad a_1(q)]^T$ is formed from the prediction coefficients and the index k in (3.1) is confined to the interval $0 \leq k \leq N+q-1$. The subscript 1 indicates that these coefficients come from the application of the LP algorithm to s(k). The coefficients $\{a_2(k)\}_1^q$ which follow come from the application of the LP algorithm to x(k).

In LPC, the optimal distance measure is the minimum of the sum of squares of the elements of $\underline{\varepsilon}$, as a function of the $\{a_1(i)\}_1^q$. If the loss function L_ε is defined as

$$L_\varepsilon = \sum_{k=0}^{N+q-1} \varepsilon^2(k) = \underline{\varepsilon}^T \underline{\varepsilon} \ , \tag{3.6}$$

then the minimum of L_ε with respect to the $\{a_1(i)\}_1^q$ is to be found. Using vector calculus, we have

$$\frac{\partial L_\varepsilon}{\partial \underline{a}_1} = \frac{\partial}{\partial \underline{a}_1} [\underline{\varepsilon}^T \underline{\varepsilon}] = 2 \ \underline{\varepsilon}^T \frac{\partial \underline{\varepsilon}}{\partial \underline{a}_1} \ .$$

The minimum of L_ε is obtained by setting this expression equal to zero:

$$\underline{\varepsilon}^T \frac{\partial \underline{\varepsilon}}{\partial \underline{a}_1} = 0 \ .$$

From (3.5), $\partial \underline{\varepsilon} / \partial \underline{a}_1 = \underline{H}_s$ and $\underline{\varepsilon}^T \underline{H}_s = 0$, or

$$\underline{H}_s^T \ \underline{\varepsilon} = 0 \ . \tag{3.7}$$

Substituting (3.5) into (3.7) gives

$$\underline{H}_s^T \ \underline{s} + \underline{H}_s^T \ \underline{H}_s \ \underline{a}_1 = 0$$

or

$$\underline{H}_s^T \ \underline{H}_s \ \underline{a}_1 = -\underline{H}_s^T \ \underline{s} \ . \tag{3.8}$$

Note that the matrix $\underline{H}_s^T \underline{H}_s$ and the vector $\underline{H}_s^T \underline{s}$ are defined by

$$\underline{H}_s^T \ \underline{H}_s = \begin{bmatrix} R_{ss}(0) & R_{ss}(1) & \cdots & R_{ss}(q-1) \\ R_{ss}(1) & R_{ss}(0) & \cdots & R_{ss}(q-2) \\ \vdots & \vdots & \ddots & \vdots \\ R_{ss}(q-1) & R_{ss}(q-2) & \cdots & R_{ss}(0) \end{bmatrix} \tag{3.9a}$$

and

$$\underline{H}_s^T \underline{s} = \begin{bmatrix} R_{ss}(1) \\ R_{ss}(2) \\ \cdot \\ \cdot \\ \cdot \\ R_{ss}(q) \end{bmatrix} . \qquad (3.9b)$$

$R_{ss}(k)$ is given by

$$R_{ss}(k) = \sum_{i=0}^{N-1-|k|} s(i)\ s(i+|k|) . \qquad (3.10)$$

Equation (3.8) is a matrix equation representation for the Yule-Walker expressions

$$\sum_{i=1}^{q} a_1(i)\ R_{ss}(i-k) = -R_{ss}(k) , \qquad (3.11)$$

for $k = 1, \ldots, q$.

If the sequence $s(k)$ is contaminated by additive noise to produce the series

$$x(k) = s(k) + n(k) , \qquad (3.12)$$

and an AR(q) model is forced on the noisy data, similar results are obtained. The AR model forced on the noisy data is

$$x(k) = -\sum_{i=1}^{q} a_2(i)\ x(k-i) + e(k) , \qquad (3.13)$$

$a_2(0) = 1.0$. The $\{a_2(i)\}_1^q$ are the prediction coefficients and $e(k)$ is the resulting error sequence. If the matrix $\underline{H}_x$ and the vector $\underline{x}$ are formed from the data $x(0), \ldots, x(N-1)$ in a manner similar to $\underline{H}_s$ and $\underline{s}$, then the loss function L_e for the noisy data is

$$L_e = \sum_{k=0}^{N+q-1} e^2(k) = \underline{e}^T \underline{e} \ , \tag{3.14}$$

with $\underline{e} = [e(0) \quad e(1) \quad \ldots \quad e(N+q-1)]^T$. Minimizing L_e with respect to the $\{a_2(i)\}_1^q$ results in

$$\underline{H}_x^T \underline{H}_x \underline{a}_2 = -\underline{H}_x^T \underline{x} \tag{3.15}$$

as the expression defining the least squares estimate for the $\{a_2(i)\}_1^q$ defined in (3.13). The elements of the matrix $\underline{H}_x^T\underline{H}_x$ and the vector $\underline{H}_x^T\underline{x}$ are formed from the autocorrelation function of x(k) as in (3.10) with x(k) replacing s(k). The $\{a_1(i)\}_1^q$ represent the LPC coefficients determined from the undegraded signal, while the $\{a_2(i)\}_1^q$ are the LPC parameters obtained from noisy data, with no attempt made to eliminate the effects of additive noise.

Constructing the matrix $\underline{H}_n$ and the vector $\underline{n}$ from the additive noise sequence n(k), the following relationships hold:

$$\underline{H}_x = \underline{H}_s + \underline{H}_n \ , \tag{3.16}$$

$$\underline{H}_x^T \underline{H}_x = \underline{H}_s^T \underline{H}_s + \underline{H}_n^T \underline{H}_n + \underline{H}_s^T \underline{H}_n + \underline{H}_n^T \underline{H}_s \ . \tag{3.17}$$

The $\underline{H}_n^T\underline{H}_n$ term is a matrix formed of the autocorrelation terms of n(k), and the terms $\underline{H}_s^T\underline{H}_n$ and $\underline{H}_n^T\underline{H}_s$ contain the cross-correlation terms between n(k) and s(k). If it can be assumed that s(k) and n(k) are uncorrelated, (3.17) becomes

$$\underline{H}_x^T \underline{H}_x = \underline{H}_s^T \underline{H}_s + \underline{H}_n^T \underline{H}_n \ . \tag{3.18}$$

With (3.12), (3.16), and (3.18) substituted into (3.15), we

have

$$[\underline{H}_s^T \underline{H}_s + \underline{H}_n^T \underline{H}_n] \underline{a}_2 = -[\underline{H}_s + \underline{H}_n]^T (\underline{s} + \underline{n}) \quad (3.19a)$$

$$= -[\underline{H}_s^T \underline{s} + \underline{H}_n^T \underline{n}] , \quad (3.19b)$$

where the assumption of uncorrelated signal and noise is used to reduce the right hand side of (3.19b) from (3.19a). Solving equations (3.8) and (3.19b) for $\underline{a}_1$ and $\underline{a}_2$, respectively, we obtain

$$\hat{\underline{a}}_1 = -[\underline{H}_s^T \underline{H}_s]^{-1} \underline{H}_s^T \underline{s} \quad (3.20)$$

and

$$\hat{\underline{a}}_2 = -[\underline{H}_s^T \underline{H}_s + \underline{H}_n^T \underline{H}_n]^{-1}[\underline{H}_s^T \underline{s} + \underline{H}_n^T \underline{n}] \quad (3.21)$$

as the least squares estimates for $\underline{a}_1$ and $\underline{a}_2$. The vector $\hat{\underline{a}}_2$ can be related to $\hat{\underline{a}}_1$ by pre-multiplying (3.19b) by $[\underline{H}_s^T \underline{H}_s]^{-1}$ to give

$$[\underline{I} + (\underline{H}_s^T \underline{H}_s)^{-1} \underline{H}_n^T \underline{H}_n] \hat{\underline{a}}_2 = -[\underline{H}_s^T \underline{H}_s]^{-1} \underline{H}_s^T \underline{s}$$

$$-[\underline{H}_s^T \underline{H}_s]^{-1} \underline{H}_n^T \underline{n}$$

$$= \hat{\underline{a}}_1 - [\underline{H}_s^T \underline{H}_s]^{-1} \underline{H}_n^T \underline{n}. \quad (3.22)$$

Solving (3.22) for $\hat{\underline{a}}_2$ gives

$$\hat{\underline{a}}_2 = [\underline{H}_s^T \underline{H}_s + \underline{H}_n^T \underline{H}_n]^{-1} \underline{H}_s^T \underline{H}_s \hat{\underline{a}}_1$$

$$- [\underline{H}_s^T \underline{H}_s + \underline{H}_n^T \underline{H}_n]^{-1} \underline{H}_n^T \underline{n} . \quad (3.23)$$

From (3.23) it is apparent that the addition of n(k) has degraded the $\hat{\underline{a}}_2$ in two ways:

1) a bias term $[\underline{H}_s^T \underline{H}_s + \underline{H}_n^T \underline{H}_n]^{-1} \underline{H}_n^T \underline{n}$ has been subtracted;

2) the relative magnitudes of the $\{\hat{a}_2(i)\}_1^q$ have been changed due to the matrix multiplying effect of the expression $[\underline{H}_s^T\underline{H}_s + \underline{H}_n^T\underline{H}_n]^{-1}\underline{H}_s^T\underline{H}_s$.

The results of equations (3.19b) through (3.23) are valuable in showing the distortion possible when noise is added to a sequence that is to be the input to an LPC system. These results are based on the explicit assumption that s(k) and n(k) are uncorrelated and fail to account for non-zero cross-correlation terms (the terms $\underline{H}_s^T\underline{H}_n$, etc.). It is this effect that is the primary hindrance in using the technique mentioned by Christiansen [11]. Results showing the distortion introduced by n(k) on the inverse spectrum derived from the $\{\hat{a}_2(i)\}_1^q$ and the effects of assuming that n(k) and s(k) are uncorrelated will be shown in Chapter 4.

ARMA Model Approach

Presented in this section are the details of the ARMA process which results from adding white noise to an AR process. The effects of additive white noise upon an AR(q) process are discussed in [10], [30], and [39]. The potential advantage of this approach is that it includes the noise effects explicitly in a more general model than the original AR(q) process. The model is developed on the following assumptions:

1) s(k) is a proper AR(q) sequence described by

$$\sum_{i=0}^{q} a(i)\, s(k-i) = \varepsilon(k) \ , \tag{3.24}$$

for $a(0) = 1$, $a(q) \neq 0$, and $q > 0$, with $\varepsilon(k)$ an independent, identically distributed (i.i.d.) $N(0,\sigma_\varepsilon^2)$ noise sequence and $s(k)$ stationary;

2) $s(k)$ is contaminated by $n(k)$ to form the observable data

$$x(k) = s(k) + n(k) \ , \tag{3.25}$$

where $s(k)$ and $n(k)$ are independent and $n(k)$ is an i.i.d. $N(0,\sigma_n^2)$ noise sequence

The model has q+2 parameters--$\{a(i)\}_1^q$, σ_ε^2, and σ_n^2. The data available for analysis to determine estimates of these parameters is the data sequence $x(0), \ldots, x(N-1)$.

Combining (3.24) and (3.25), we have

$$\sum_{i=0}^{q} a(i)\, x(k-i) = \sum_{i=0}^{q} a(i)\, n(k-i) + \varepsilon(k) \ . \tag{3.26}$$

A sequence $y(k)$ is defined as

$$y(k) = \sum_{i=0}^{q} a(i)\, x(k-i) \tag{3.27}$$

or

$$y(k) = \sum_{i=0}^{q} a(i)\, n(k-i) + \varepsilon(k) \ . \tag{3.28}$$

If $R_{yy}(k) = E[y(i)y(i+k)]$, it can be shown, using (3.28), that $R_{yy}(k) = 0$ for $|k| > q$. From (3.28), $y(k)$ is seen to be stationary. Combining this with the property that $R_{yy}(k) = 0$, $|k| > q$, shows $y(k)$ to be a moving average sequence MA(p), with $p \leq q$. Also from (3.28),

$R_{yy}(q) = \sigma_n^2 a(q) \neq 0$, by the hypothesis under assumption 1) above. As a result, y(k) is an MA(q) process, and there exists a sequence of random variables v(k), i.i.d. $N(0,\sigma_v^2)$ and constants $\{b(i)\}_1^q$ such that

$$y(k) = \sum_{j=0}^{q} b(j)\ v(k-j)\ , \tag{3.29}$$

b(0) = 1.0. Combining (3.27) and (3.29) gives

$$\sum_{i=0}^{q} a(i)\ x(k-i) = \sum_{j=0}^{q} b(j)\ v(k-j)\ . \tag{3.30}$$

Thus, the sequence x(k) can be viewed as an ARMA(q,q) process. While the original model has q+2 parameters--$\{a(i)\}_1^q$, σ_ε^2, and σ_n^2--the new model has 2q+1 parameters--$\{a(i)\}_1^q$, $\{b(i)\}_1^q$, and σ_v^2. From (3.29), we have

$$R_{yy}(k) = \sigma_v^2 \sum_{i=0}^{q-|k|} b(i)\ b(i+k)\ , \tag{3.31}$$

so the expanded parameter set could equivalently be expressed as $\{a(i)\}_1^q$ and $\{R_{yy}(i)\}_0^q$.

Using the definition for y(k) in (3.28),

$$R_{yy}(k) = \sigma_\varepsilon^2\ \delta(k) + \sigma_n^2 \sum_{i=0}^{q-|k|} a(i)\ a(i+k)\ ,$$

where $\delta(k) = \begin{cases} 1, & k = 0 \\ 0, & k \neq 0 \end{cases}$ is the Kronecker delta function.

It is also possible to develop an expression for $R_{yy}(k)$ from (3.27). The three expressions for $R_{yy}(k)$ are summarized in (3.32):

$$R_{yy}(k) = \sigma_v^2 \sum_{i=0}^{q-|k|} b(i)\, b(i+k) \qquad (3.32a)$$

$$= \sum_{i=0}^{q} \sum_{j=0}^{q} a(i)\, a(j)\, R_{xx}(k+i-j) \qquad (3.32b)$$

$$= \sigma_\varepsilon^2\, \delta(k) + \sigma_n^2 \sum_{i=0}^{q-|k|} a(i)\, a(i+k)\ . \qquad (3.32c)$$

Thus, the addition of n(k) to s(k) produces the following relationships between the parameters:

1) equation (3.32a) gives the autocorrelation function $R_{yy}(k)$ for any MA(q) process;
2) in (3.32b) $R_{yy}(k)$ is in terms of the AR coefficients and $R_{xx}(k)$, the autocorrelation function of the data x(k), and is a valid expression for any MA autocorrelation function, given that the a(i) can be zero, i > 0, if x(k) is itself an MA process;
3) another definition for $R_{yy}(k)$ given in (3.32c) arises as a result of the noise model defined by (3.24) and (3.25);
4) the ARMA parameters $\{a(i)\}_1^q$ and $\{b(i)\}_1^q$ for the process x(k) are related through the autocorrelation function $R_{yy}(k)$, the relationship being expressed by (3.32a) and (3.32c).

A comparison of the ARMA model approach just described with a forced LPC fit of the data, represented by the solution of (3.21), shows two interesting facts. First, the forced LPC model, from a spectral point of view, must match the spectral characteristics of the data x(k) as

closely as possible. This spectral match includes those characteristics introduced by the noise. The flattening effect exhibited by forcing an all-pole LPC fit on a signal degraded by additive white noise will be illustrated in the next chapter. The second observation involves the assumption of the model form. If the original sequence s(k) is AR(q), then the addition of white noise results in an ARMA(q,q) process, x(k). This process is equivalently an AR(∞) process. The forced LPC fit is actually representative of the first step in the process discussed in [13] for estimating ARMA parameters, that is, underfitting the AR(∞) process. Using the technique given in [13], the ARMA model approach can then be viewed as a procedure by which the AR(q) and MA(q) parameters are estimated from the AR(∞) parameters.

With the development of the AR-to-ARMA transformation model complete, the processing steps required to use this algorithm for extracting the q AR parameters from noisy data are summarized:

1) because most ARMA estimation procedures require initial guesses for the parameters, a procedure that provides initial estimates for the parameters might be needed;
2) an algorithm suitable for estimating the AR and MA coefficients from a time series must be selected;
3) if it is desired to make use of the nonlinear regression stage to improve the AR parameter estimates,

as suggested by Pagano [30], the algorithm is terminated by that nonlinear regression procedure.

Of these three steps, the effort in this project has been directed toward the second step, the estimation of the ARMA parameters, concentrating on low order processes, especially the ARMA(1,1) model.

As will be pointed out in the next chapter, in testing the ARMA estimation algorithms and the feasibility of this model, synthetic data are used in all tests. These data are generated from a known AR or ARMA model with an approximately white noise excitation process. To avoid introducing the problems encountered in obtaining suitable initial parameter estimates into the ARMA parameter estimation stage, the coefficients used to generate the process are often used as the initial guesses. Thus, they represent the best possible guesses for the parameter values. Performing the experiments in this manner emphasizes the accuracy of the AR coefficient estimates. Where just AR or MA coefficients are being estimated, using zeros for all initial estimates yields good results. This is not possible where both AR and MA parameters are being estimated. Using zeros as initial estimates for the ARMA parameters leads to a singular matrix in the estimation algorithm. While using the model parameters as initial estimates is an unrealistic approach, it does place the emphasis on the validity of the transformation model and

the estimation algorithm being tested. A small number of tests using initial estimates arrived at through experimental methods showed that the primary difference in the experiment was the number of iterations required for the algorithm to converge to a final solution, not the final solution itself. Chapter 4 contains the results achieved using the transformation model to estimate the parameters of an ARMA process.

The AR(1) Process Plus White Noise

From the preceding section, if an AR(1) process is corrupted by additive white noise, the resulting data can be modeled as an ARMA(1,1) process. The AR(1) process s(k) is given by

$$s(k) + a\, s(k-1) = \varepsilon(k) , \tag{3.33}$$

where a is the single AR parameter. If n(k) is the additive white noise corrupting the AR process as in (3.25), after combining (3.25) and (3.33) we have

$$[x(k) - n(k)] + a\,[x(k-1) - n(k-1)] = \varepsilon(k)$$

or

$$x(k) + a\, x(k-1) = n(k) + a\, n(k-1) + \varepsilon(k) . \tag{3.34}$$

As before, define the sequence y(k) to be

$$y(k) = v(k) + b\, v(k-1) \tag{3.35a}$$

$$= x(k) + a\, x(k-1) \tag{3.35b}$$

$$= n(k) + a\, n(k-1) + \varepsilon(k) , \tag{3.35c}$$

where (3.35a) is the expression for an MA(1) sequence. In (3.35a), b is the single MA parameter and v(k) is an i.i.d.

$N(0,\sigma_v^2)$ white noise process that is the excitation sequence for the MA process y(k). The process v(k) is also the excitation sequence defined for the ARMA model resulting from adding n(k) to s(k), as described in the previous section. From (3.34) and (3.35a) we have

$$x(k) + a\, x(k-1) = v(k) + b\, v(k-1) , \tag{3.36}$$

the description for the ARMA(1,1) process x(k).

The equations in (3.35) give three ways of defining the sequence y(k). For each of the three expressions for y(k) there is the corresponding equation for the autocorrelation function $R_{yy}(k)$:

$$R_{yy}(k) = \sigma_v^2 \sum_{i=0}^{1-|k|} b(i)\, b(i+k) \tag{3.37a}$$

$$= \sum_{i=0}^{1} \sum_{j=0}^{1} a(i)\, a(j)\, R_{xx}(k+i-j) \tag{3.37b}$$

$$= \sigma_\varepsilon^2\, \delta(k) + \sigma_n^2 \sum_{i=0}^{1-|k|} a(i)\, a(i+k) , \tag{3.37c}$$

for k = 0 and 1, and a(0) = b(0) = 1. As pointed out in the discussion of (3.32), equation (3.37c) is unique to the ARMA model formed by adding n(k) to an AR(1) process.

Using (3.37a) and (3.37c), the generation of the MA coefficient by the addition of n(k) to s(k) is demonstrated for various signal-to-noise ratios. From (3.37c),

$$R_{yy}(0) = \sigma_\varepsilon^2 + \sigma_n^2\, (1 + a^2) , \tag{3.38a}$$

$$R_{yy}(1) = \sigma_n^2\, a , \tag{3.38b}$$

$$R_{yy}(k) = 0 , \quad |k| \geq 2 , \tag{3.38c}$$

with a(1) = a the single AR parameter. Alternatively, using (3.37a) with b(1) = b,

$$R_{yy}(0) = \sigma_v^2 (1 + b^2) , \tag{3.39a}$$

$$R_{yy}(1) = \sigma_v^2 b , \tag{3.39b}$$

$$R_{yy}(k) = 0 , \quad |k| \geq 2 . \tag{3.39c}$$

Equating the terms for $R_{yy}(0)$ and $R_{yy}(1)$ from (3.38) and (3.39) gives

$$\sigma_\varepsilon^2 + \sigma_n^2 [1 + a^2] = \sigma_v^2 [1 + b^2] ,$$

$$\sigma_n^2 a = \sigma_v^2 b .$$

Solving these expressions for b and σ_v^2, we obtain

$$b = \frac{1}{2 \sigma_n^2 a} \left([\sigma_\varepsilon^2 + \sigma_n^2 (1 + a^2)] \pm \right.$$

$$\left. [[\sigma_\varepsilon^2 + \sigma_n^2 (1 + a^2)]^2 - 4 \sigma_n^4 a^2]^{1/2} \right) \tag{3.40}$$

and

$$\sigma_v^2 = \sigma_n^2 \frac{a}{b} . \tag{3.41}$$

Note there are two possible values for the MA coefficient b. If the minus sign is used in (3.40), b_- will be used to symbolize the value of b. If the plus sign is used, b_+ will be written.

It is important that the parameters b and σ_v^2 possess certain properties. Appendix C shows the derivation of the

properties listed here:

1) b is real;

2) $b_+ = 1/b_-$, $|b_-| < 1$;

3) $\sigma_v^2 \geq 0$;

4) $\sigma_{v+}^2 = b_-^2 \, \sigma_{v-}^2$.

The minus and plus subscripts on the σ_v^2 term in 4) indicate whether b_- or b_+ is used to compute σ_v^2 in (3.41). Property 2) establishes that $b = b_-$ corresponds to a root inside the unit circle in the Z-domain. To demonstrate this, if

$$B(z) = 1 + b\, z^{-1} ,$$

then $B(z) = 0$ when $z = -b$. If the series generated by expanding $B^{-1}(z)$ is

$$B^{-1}(z) = \frac{1}{1 + b\, z^{-1}} = 1 - b\, z^{-1} + b^2\, z^{-2} - + \cdots ,$$

then the weights of the z^{-i} terms converge iff $|b| < 1$. Thus, $b = b_-$ corresponds to the convergent root. This is the invertibility property discussed by Box and Jenkins [10]. In designing an ARMA process it is appropriate to choose the MA parameters so that the MA operator satisfies the invertibility condition, that is, the roots of the MA operator polynomial lie inside the unit circle. For the MA(1) process, b_- and σ_{v-}^2 are the appropriate choices.

Given expressions for b and σ_v^2 in (3.40) and (3.41), the effect of various SNR's on these parameters will be demonstrated. If σ_s^2 is the variance of s(k), the AR(q) process, then σ_s^2 is given by

$$\sigma_s^2 = \sigma_\varepsilon^2/[1 + \sum_{i=1}^{q} a(i)\ \rho(i)]\ , \qquad (3.42)$$

where $\rho(i) = R_{ss}(i)/R_{ss}(0)$. For $\sigma_\varepsilon^2 = 1.0$ and $q = 1$, the process variance is $\sigma_s^2 = 1/(1 - a^2)$, since $\rho(1) = -a$. The SNR is

$$SNR = \frac{\sigma_s^2}{\sigma_n^2} = \frac{1}{\sigma_n^2\ (1 - a^2)}\ . \qquad (3.43)$$

Consider now the extreme cases where $SNR \to \infty$ and $SNR \to 0$. This can also be expressed as $\sigma_n^2 \to 0$ and $\sigma_n^2 \to \infty$, respectively. The results for the behavior of b_-, b_+, σ_{v-}^2, and σ_{v+}^2 as $SNR \to 0$ or ∞ are summarized in Table 3-1.

Using (3.40) and (3.41) to compute the MA(1) parameters b and σ_v^2, the effect of a changing SNR on these parameters is found in Tables 3-2, 3-3, and 3-4. For all of this data, σ_ε^2 is arbitrarily set at 1.0. In Table 3-2 the results are computed using 0.1 as the single AR parameter a. In Tables 3-3 and 3-4, a is 0.5 and 0.9, respectively. From the data in these tables, it is clear that as $SNR \to \infty$, the observed data x(k) approaches the desired AR(1) process since $\sigma_{v-}^2 \to \sigma_\varepsilon^2 = 1$ and $b_- \to 0$. When $SNR \to 0$, however, the observed data begins to resemble the additive white noise n(k). This is true because $b_- \to a$ and (3.36) becomes

$$x(k) + a\ x(k-1) = v(k) + a\ v(k-1) \qquad (3.44)$$

In the Z-domain this can be written as

Table 3-1

Behavior of Parameters at SNR Extremes

SNR	b_-	b_+	σ^2_{v-}	σ^2_{v+}
∞	0	∞	σ^2_ε	0
0	a	1/a	∞	∞

Table 3-2

Equivalent ARMA(1,1) Parameters at Various SNR's for AR(1) Process Plus White Noise, a(1) = 0.1, $\sigma_\varepsilon^2 = 1.0$

SNR	σ_n^2	b_-	σ_{v-}^2	b_+	σ_{v+}^2
∞	0	0	$\sigma_\varepsilon^2 = 1$	∞	0
60	$.10101 \times 10^{-5}$	$.10101 \times 10^{-6}$	$.10000 \times 10^{1}$	$.99000 \times 10^{7}$	$.10203 \times 10^{-13}$
40	$.10101 \times 10^{-3}$	$.10100 \times 10^{-4}$	$.10001 \times 10^{1}$	$.99010 \times 10^{5}$	$.10202 \times 10^{-9}$
20	$.10101 \times 10^{-1}$	$.99990 \times 10^{-3}$	$.10102 \times 10^{1}$	$.10001 \times 10^{4}$	$.10100 \times 10^{-5}$
0	$.10101 \times 10^{1}$	$.50126 \times 10^{-1}$	$.20151 \times 10^{1}$	$.19950 \times 10^{2}$	$.50632 \times 10^{-2}$
-20	$.10101 \times 10^{3}$	$.99010 \times 10^{-1}$	$.10202 \times 10^{3}$	$.10100 \times 10^{2}$	$.10001 \times 10^{1}$
-40	$.10101 \times 10^{5}$	$.99990 \times 10^{-1}$	$.10102 \times 10^{5}$	$.10001 \times 10^{2}$	$.10100 \times 10^{3}$
-60	$.10101 \times 10^{7}$	.10000	$.10101 \times 10^{7}$	$.10000 \times 10^{2}$	$.10101 \times 10^{5}$
$-\infty$	∞	.1	∞	$.10000 \times 10^{2}$	∞

Table 3-3

Equivalent ARMA(1,1) Parameters at Various SNR's for AR(1) Process Plus White Noise, $a(1) = 0.5$, $\sigma_\varepsilon^2 = 1.0$

SNR	σ_n^2	b_-	σ_{v-}^2	b_+	σ_{v+}^2
∞	0	0	$\sigma_\varepsilon^2 = 1$	∞	0
60	$.13333 \times 10^{-5}$	$.66667 \times 10^{-6}$	$.10000 \times 10^{1}$	$.15000 \times 10^{7}$	$.44444 \times 10^{-12}$
40	$.13333 \times 10^{-3}$	$.66656 \times 10^{-4}$	$.10002 \times 10^{1}$	$.15002 \times 10^{5}$	$.44437 \times 10^{-8}$
20	$.13333 \times 10^{-1}$	$.65577 \times 10^{-2}$	$.10166 \times 10^{1}$	$.15249 \times 10^{3}$	$.43718 \times 10^{-4}$
0	$.13333 \times 10^{1}$	.26795	$.24880 \times 10^{1}$	$.37321 \times 10^{1}$	.17863
-20	$.13333 \times 10^{3}$	.49507	$.13466 \times 10^{3}$	$.20199 \times 10^{1}$	$.33004 \times 10^{2}$
-40	$.13333 \times 10^{5}$	.49995	$.13335 \times 10^{5}$	$.20002 \times 10^{1}$	$.33330 \times 10^{4}$
-60	$.13333 \times 10^{7}$	.50000	$.13333 \times 10^{7}$	$.20000 \times 10^{1}$	$.33333 \times 10^{6}$
$-\infty$	∞	.5	∞	$.20000 \times 10^{1}$	∞

Table 3-4

Equivalent ARMA(1,1) Parameters at Various SNR's for AR(1) Process Plus White Noise, $a(1) = 0.9$, $\sigma_\varepsilon^2 = 1.0$

SNR	σ_n^2	b_-	σ_{v-}^2	b_+	σ_{v+}^2
∞	0	0	$\sigma_\varepsilon^2 = 1$	∞	0
60	$.52632 \times 10^{-5}$	$.47368 \times 10^{-5}$	$.10000 \times 10^{1}$	$.21111 \times 10^{6}$	$.22437 \times 10^{-10}$
40	$.52632 \times 10^{-3}$	$.47323 \times 10^{-3}$	$.10010 \times 10^{1}$	$.21131 \times 10^{4}$	$.22416 \times 10^{-6}$
20	$.52632 \times 10^{-1}$	$.43330 \times 10^{-1}$	$.10932 \times 10^{1}$	$.23079 \times 10^{2}$	$.20525 \times 10^{-2}$
0	$.52632 \times 10^{1}$	.62679	$.75573 \times 10^{1}$	$.15954 \times 10^{1}$	$.29690 \times 10^{1}$
-20	$.52632 \times 10^{3}$	.89143	$.53137 \times 10^{3}$	$.11218 \times 10^{1}$	$.42226 \times 10^{3}$
-40	$.52632 \times 10^{5}$	.89991	$.52637 \times 10^{5}$	$.11112 \times 10^{1}$	$.42627 \times 10^{5}$
-60	$.52632 \times 10^{7}$	.90000	$.52632 \times 10^{7}$	$.11111 \times 10^{1}$	$.42632 \times 10^{7}$
$-\infty$	∞	.9	∞	$.11111 \times 10^{1}$	∞

$$[1 + a\, z^{-1}]\, X(z) = [1 + a\, z^{-1}]\, V(z) \qquad (3.45)$$

Cancelling the common factor $[1 + az^{-1}]$, (3.45) becomes

$$X(z) = V(z)\ , \qquad (3.46a)$$

$$x(k) = v(k)\ , \qquad (3.46b)$$

and the data $x(k)$ is now a white noise sequence since $v(k)$ is the white noise excitation sequence for the equivalent ARMA model. To show that $x(k) \simeq n(k)$ as $SNR \to 0$, we examine (3.35c) and (3.36), which give

$$x(k) + a\, x(k-1) = n(k) + a\, n(k-1) + \varepsilon(k) \qquad (3.47)$$

when combined. In the Z-domain (3.47) becomes

$$[1 + a\, z^{-1}]\, X(z) = [1 + a\, z^{-1}]\, N(z) + E(z)\ . \qquad (3.48)$$

As $SNR \to 0$, $\sigma_n^2 \to \infty$ and the $[1 + az^{-1}]N(z)$ term dominates the right side of (3.48). Consequently, the left side of (3.48) can be approximated as

$$[1 + a\, z^{-1}]\, X(z) \simeq [1 + a\, z^{-1}]\, N(z)\ . \qquad (3.49)$$

Cancelling the common factor as before, we have

$$X(z) \simeq N(z)\ , \qquad (3.50a)$$

$$x(k) \simeq n(k)\ . \qquad (3.50b)$$

This result is intuitively appealing since it shows that the data $x(k)$ becomes more like the additive noise $n(k)$ as the SNR becomes poorer.

The ideas generated in this section are important to the work done in evaluating the ARMA noise model because much of the data found in Chapter 4 is based on the analysis of the AR(1)-to-ARMA(1,1) transformation model.

As will be seen in Chapter 4, two approaches are taken in analyzing the this model:

1) simulate the s(k) + n(k) degradation by computing the MA(1) parameters b_- and σ^2_{v-} and generating directly the resulting ARMA(1,1) process using a noise sequence v(k) as the excitation;
2) generate the AR(1) process s(k) and add the white noise n(k) to obtain the equivalent ARMA(1,1) process x(k).

More details on this are given in Chapter 4.

When generating estimates of parameters from data, it is important to know the variances associated with those estimates. If a large number of estimates are available, sample statistics for the parameter estimates can be obtained. For low order ARMA processes, however, it is possible to obtain equations describing the variance of the parameter estimates. In Chapter 7 of [10], Box and Jenkins discuss model estimation procedures and develop the variance expressions for ARMA processes. Specifically, the variance of the parameter estimates is of interest in the following cases:

1) the estimate for a(1) in an AR(1) process;
2) the estimate for b(1) in an MA(1) process;
3) the estimates for a(1) and b(1) in an ARMA(1,1) process.

For the first order cases analyzed in Chapter 4, the sample variance of the parameters estimates are compared to the

theoretical values.

If N is the number of points in each frame of data, the variance of a = a(1) for the AR(1) process is

$$\text{var}[\hat{a}] = \frac{1}{N} (1 - a^2) . \tag{3.51}$$

Likewise, the variance for b = b(1) for an MA(1) process is

$$\text{var}[\hat{b}] = \frac{1}{N} (1 - b^2) . \tag{3.52}$$

For the parameter estimates of an ARMA(1,1) process, we have

$$\text{var}[\hat{a}] = \frac{1}{N} \frac{(1 - ab)^2}{(a - b)^2} (1 - a^2) , \tag{3.53a}$$

$$\text{var}[\hat{b}] = \frac{1}{N} \frac{(1 - ab)^2}{(a - b)^2} (1 - b^2) , \tag{3.53b}$$

$$\text{cov}[\hat{a},\hat{b}] = - \frac{1}{N} \frac{(1 - ab)}{(a - b)^2} (1 - a^2) (1 - b^2) . \tag{3.53c}$$

The expressions found in (3.51), (3.52), and (3.53) are for maximum likelihood estimates. Details for the derivations of these expressions are found in [10], Chapter 7.

Steiglitz Mode 1 Estimation Method

The estimation of ARMA parameters comprises the major effort of this dissertation. The following sections present five procedures that have been used in this work to estimate parameters. Details for the nonlinear regression algorithm suggested by Pagano for improving the AR estimates are also presented.

One possible procedure for estimating the parameters

of an autoregressive moving-average process is the mode 1 iterative method by Steiglitz and McBride [36]. The approach is: given input and output sequences for an unknown system, determine the filter which approximates the unknown system. In the Z-domain the model for the filter is the ratio of two rational polynomials A(z) and B(z). Graphically, the problem is illustrated in Figure 3-1. The polynomials A(z) and B(z) are given by

$$A(z) = \sum_{i=0}^{q} a(i)\, z^{-i} ,$$

a(0) = 1.0, and

$$B(z) = \sum_{i=0}^{p} b(i)\, z^{-i} .$$

Note that in this method, b(0) does not necessarily equal one. The resulting parameter set is $\{a(i)\}_1^q$ and $\{b(i)\}_0^p$. The variance of the excitation sequence is not estimated explicitly. It is, however, related to b(0). The coefficients a(i) and b(i) in A(z) and B(z), respectively, are selected to minimize E(z) in some sense. The model's response, U(z), is

$$U(z) = \frac{B(z)}{A(z)}\, V(z) \qquad (3.54)$$

or

$$A(z)\, U(z) = B(z)\, V(z) . \qquad (3.55)$$

Also, from Figure 3.1, the error is given by

$$E(z) = U(z) - X(z) . \qquad (3.56)$$

Steiglitz and McBride then perform a "quasi-linearization"

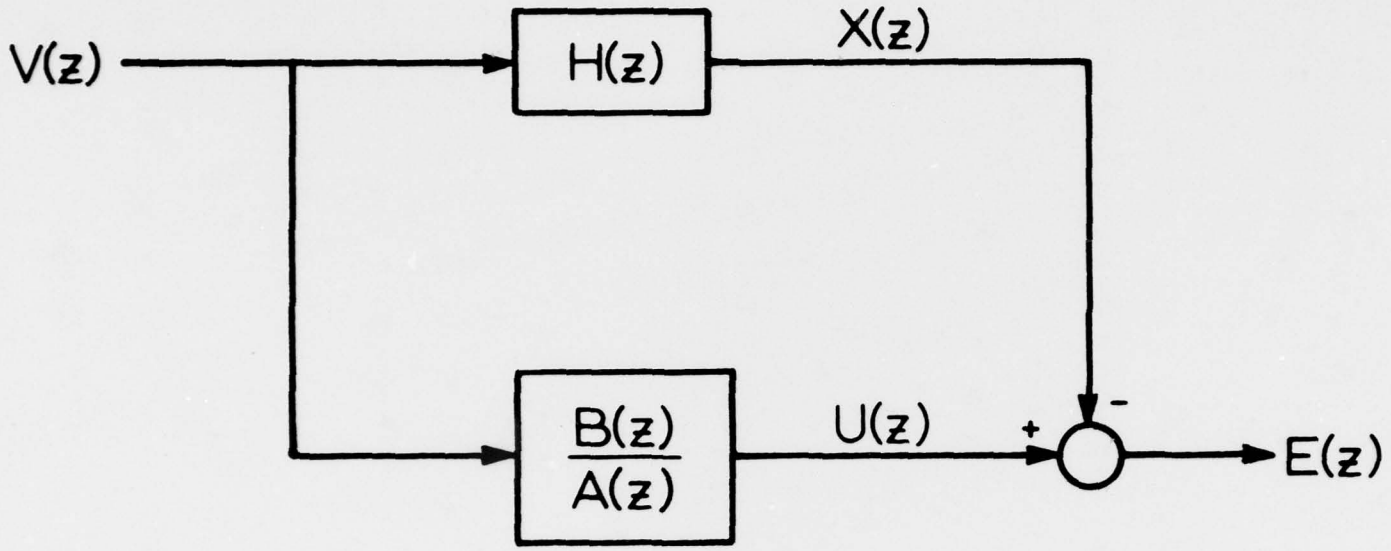


Figure 3-1: Block Diagram of System Identification Model used in Mode 1 Iterative Procedure for Estimating Parameters.

on (3.55), using previous iterations to form approximations to the derivatives,

$$A_i(z)\, U_i(z) + [A_{i+1}(z) - A_i(z)]\, U_i(z) + A_i(z)\, [U_{i+1}(z) - U_i(z)] = B_{i+1}(z)\, V(z) \; . \qquad (3.57)$$

The subscript indicates the iteration number. Replacing $U_i(z)$ with $X(z)$ in (3.57) and simplifying gives

$$A_i(z)\, U_{i+1}(z) = [A_i(z) - A_{i+1}(z)]\, X(z) + B_{i+1}(z)\, V(z) \; . \qquad (3.58)$$

Solving for $U_{i+1}(z)$ and using that expression for $U(z)$ in (3.56) gives

$$E_{i+1}(z) = U_{i+1}(z) - X(z) = \frac{B_{i+1}(z)}{A_i(z)}\, V(z) - \frac{A_{i+1}(z)}{A_i(z)}\, X(z) \; . \qquad (3.59)$$

It is the form of (3.59) that suggests the mode 1 technique presented in [36]. Noting that both $V(z)$ and $X(z)$ are recursively filtered through the i^{th} iteration of $A(z)$, define $\hat{V}(z) = V(z)/A_i(z)$ and $\hat{X}(z) = X(z)/A_i(z)$. With these definitions, the time domain representation for (3.59) is

$$e(k) = \sum_{i=0}^{p} b(i)\, \hat{v}(k-i) - \sum_{j=0}^{q} a(j)\, \hat{x}(k-j) \; . \qquad (3.60)$$

The iteration notation has been dropped for clarity. The coefficients $\{a(i)\}_1^q$ and $\{b(i)\}_0^p$ are selected to minimize

e(k) in the least squares sense. The least squares procedure requires the solution of the matrix equation

$$\underline{R}_{vx}\,\underline{\theta} = \underline{r}_{vx}\,, \tag{3.61}$$

where $\underline{R}_{vx}$ is a matrix composed of the auto- and cross-correlations of v(k) and x(k) and $\underline{r}_{vx}$ is a vector composed of those correlations. $\underline{\theta}$ is the solution vector containing the desired a(i) and b(i) coefficients. Use of this method thus requires the solution of a set of p+q+1 linear simultaneous equations.

For application to the estimation of the coefficients of an ARMA process as proposed by Steiglitz in [35], this technique must be modified slightly. When only the output of the system is known, v(k) is assumed to be the Kronecker delta function. Also, the system output x(k) may be modified so that it more closely resembles an impulse response, as the assumption for v(k) implies. Steiglitz proposes several operations that might improve the quality of the parameter estimates. These procedures, applied to the observed data x(k), include:

1) pre-emphasis;
2) windowing;
3) generation of a minimum phase signal $x_{mp}(k)$, which has the same log magnitude spectrum as x(k);
4) removal of periodicity.

Steps 3) and 4) involve cepstral domain operations. Results from tests on the mode 1 iterative method are

presented in Chapter 4 for an ARMA(10,2) process excited by an impulse, an impulse train, and a white noise sequence.

Anderson's Time Domain Maximum Likelihood Methods

Anderson presents the details for ARMA parameter estimation procedures based on the optimization of the Gaussian likelihood equation [2]. As reviewed in Chapter 2, the methods are characterized along the following divisions:

1) time domain versus frequency domain;
2) Newton-Raphson method versus the method of scoring (Gauss-Newton method);
3) parameter set 1 (AR coefficients, MA coefficients, and excitation sequence variance) versus parameter set 2 (AR coefficients and MA covariances).

The development of these methods by Anderson is based on a matrix formulation, useful for compact presentation of the equations. This compactness, however, tends to obscure the meaning of the operations. This section presents the time domain Newton-Raphson and Gauss-Newton methods for estimating the AR and MA coefficients and the variance of the excitation sequence. Included in this review of Anderson's methods is an elaboration on the matrix notation. The equivalent scalar notation is also discussed.

The description of the ARMA(q,p) process x(k) is

$$\sum_{i=0}^{q} a(i)\, x(k-i) = \sum_{j=0}^{p} b(j)\, v(k-j) ,$$

with $a(0) = b(0) = 1$. Using the matrix notation of [2], this becomes

$$\underline{A}\,\underline{x} = \underline{B}\,\underline{v} . \tag{3.62}$$

The N x N lower triangular matrices $\underline{A}$ and $\underline{B}$ are given by

$$\underline{A} = \sum_{i=0}^{q} a(i)\, \underline{L}^{i} , \tag{3.63a}$$

$$\underline{B} = \sum_{j=0}^{p} b(j)\, \underline{L}^{j} , \tag{3.63b}$$

and $\underline{x} = [x(0) \ldots x(N-1)]^T$ and $\underline{v} = [v(0) \ldots v(N-1)]^T$ are N x 1 vectors. As before, v(k) is an i.i.d. $N(0, \sigma_v^2)$ noise sequence. In (3.63) the matrix $\underline{L}$ is the N x N matrix lag operator defined by Anderson. If $\underline{I}_{N-k}$ is the (N-k) x (N-k) identity matrix, then

$$\underline{L}^{k} = \begin{bmatrix} 0 & 0 \\ \underline{I}_{N-k} & 0 \end{bmatrix} . \tag{3.64}$$

and $\underline{L}^{k}\underline{x} = [0 \ldots 0\ x(0) \ldots x(N-1-k)]^T$. Thus the effect of pre-multiplying a vector by the matrix $\underline{L}$ to the k^{th} power is to introduce zeros in the first k positions of the vector, shifting the elements of the vector down by k places, imposing zero initial conditions on the problem. Details for the development of the matrix model formulation in (3.62) and the use of the matrix lag operator $\underline{L}$ are given in Appendix D.

With the model now defined, the Gaussian likelihood

function to be maximized is

$$f(\underline{x}|\underline{B},\underline{A},\sigma_v^2) = \frac{1}{[2\pi]^{N/2}\ |\overline{\underline{x}\ \underline{x}^T}|^{1/2}}$$

$$\cdot\ \exp\left(-\frac{1}{2\sigma_v^2}\ \underline{x}^T\ \underline{A}^T\ \underline{B}^{T^{-1}}\ \underline{B}^{-1}\ \underline{A}\ \underline{x}\right) .$$

Taking the logarithm of this function gives

$$\log[f] = -\frac{N}{2}\log(2\pi) - \frac{1}{2}\log|\overline{\underline{x}\ \underline{x}^T}| -$$

$$\frac{1}{2\sigma_v^2}\ \underline{x}^T\ \underline{A}^T\ \underline{B}^{T^{-1}}\ \underline{B}^{-1}\ \underline{A}\ \underline{x} . \qquad (3.65)$$

In (3.65) the term $-(1/2)\log|\overline{\underline{x}\ \underline{x}^T}|$ can be simplified by using the relationship

$$\underline{x} = \underline{A}^{-1}\ \underline{B}\ \underline{v} , \qquad (3.66)$$

derived from (3.62). Using (3.66) we have

$$|\overline{\underline{x}\ \underline{x}^T}| = |\underline{A}^{-1}\ \underline{B}\ \overline{\underline{v}\ \underline{v}^T}\ \underline{B}^T\ \underline{A}^{-1^T}|$$

$$= |\underline{A}^{-1}\ \underline{B}\ \sigma_v^2\ \underline{B}^T\ \underline{A}^{-1^T}|$$

$$= (\sigma_v^2)^N\ |\underline{A}|^{-2}\ |\underline{B}|^2 .$$

With this, (3.65) becomes

$$\log[f] = -\frac{N}{2}\log(2\pi) - \frac{N}{2}\log(\sigma_v^2) + \log|\underline{A}| - \log|\underline{B}|$$

$$- \frac{1}{2\sigma_v^2}\ \underline{x}^T\ \underline{A}^T\ \underline{B}^{T^{-1}}\ \underline{B}^{-1}\ \underline{A}\ \underline{x} . \qquad (3.67)$$

Equation (3.67) is the modified likelihood function. It is

modified in the sense that the probability density function for $\underline{x}$ is conditioned on the initial values assumed for $\underline{x}$. In the case where the matrix lag operator $\underline{L}$ is used, the assumed values for the initial data points $x(1-q), \ldots, x(-1)$ are all zero. This also applies to $v(k)$: $v(1-p), \ldots, v(-1)$ are zero.

Since the function in (3.67) is to be maximized, we begin by taking the partial derivative of log[f] with respect to the parameters $a(1), \ldots, a(q), b(1), \ldots, b(p)$, and σ_v^2:

$$\frac{\partial}{\partial b_i} \log[f] = -\mathrm{tr}[\underline{B}^{-1} \underline{L}^i] + \frac{1}{\sigma_v^2} \underline{x}^T \underline{A}^T \underline{B}^{T^{-1}} \underline{B}^{-1} \underline{L}^i \underline{B}^{-1} \underline{A} \underline{x} , \qquad (3.68a)$$

$$\frac{\partial}{\partial a_j} \log[f] = -\mathrm{tr}[\underline{A}^{-1} \underline{L}^j] - \frac{1}{\sigma_v^2} \underline{x}^T \underline{A}^T \underline{B}^{T^{-1}} \underline{B}^{-1} \underline{L}^j \underline{x} , \qquad (3.68b)$$

$$\frac{\partial}{\partial \sigma_v^2} \log[f] = -\frac{N}{2\sigma_v^2} + \frac{1}{2\sigma_v^4} \underline{x}^T \underline{A}^T \underline{B}^{T^{-1}} \underline{B}^{-1} \underline{A} \underline{x} . \qquad (3.68c)$$

for $i = 1, \ldots, p$ and $j = 1, \ldots, q$. Setting the expressions in (3.68) equal to zero and solving for the $\{a(i)\}_1^q$, $\{b(i)\}_1^p$, and σ_v^2 that satisfy that condition will maximize the function log[f]. Unfortunately, the relationships in (3.68) are nonlinear. This requires an

iterative procedure to solve for the maximizing parameters, since an explicit solution is unlikely.

Two iterative parameter estimation procedures are considered here:

1) the Newton-Raphson method;

2) the Gauss-Newton method.

Refer to Appendix A for the details on what these two methods involve for the general optimization problem. Defining $\underline{a} = [a(1) \ldots a(q)]^T$ and $\underline{b} = [b(1) \ldots b(p)]^T$, the application of either the NR or GN method requires the solution of the matrix equation

$$\underline{R}_i \, [\underline{\theta}_{i+1} - \underline{\theta}_i] = \underline{g}_i \ , \tag{3.69}$$

where $\underline{\theta} = [\, \underline{a}^T \,|\, \underline{b}^T \,]^T$ is the parameter vector, $\underline{g} = [\, \underline{w}^T \,|\, \underline{u}^T \,]^T$ is the gradient vector, and

$$\underline{R} = \begin{bmatrix} \underline{\Phi} & \underline{\Omega} \\ \underline{\Omega}^T & \underline{\Psi} \end{bmatrix} .$$

is the coefficient matrix appropriate to the NR or GN methods. The subscripts i and i+1 indicate the iteration number. $\underline{R}_i$ and $\underline{g}_i$ indicate those quantities are evaluated at $\underline{\theta} = \underline{\theta}_i$, the present estimate for the parameter vector.

Using the partitioned forms for $\underline{\theta}$, $\underline{g}$, and $\underline{R}$, (3.69) can be written as

$$\underline{\Phi}_i \, [\underline{b}_{i+1} - \underline{b}_i] + \underline{\Omega}_i \, [\underline{a}_{i+1} - \underline{a}_i] = \underline{w}_i \ ,$$

$$\underline{\Omega}_i^T \, [\underline{b}_{i+1} - \underline{b}_i] + \underline{\Psi}_i \, [\underline{a}_{i+1} - \underline{a}_i] = \underline{u}_i \ .$$

The algorithm requires an initial guess $\underline{\theta}_0$ for the parameter vector. In order to compute the next estimate of $\underline{\theta}$, the following must be computed:

1) the matrices $\underline{\Phi}$, $\underline{\Omega}$, and $\underline{\Psi}$ of $\underline{R}$;

2) the vectors $\underline{w}$ and $\underline{u}$ of the gradient vector $\underline{g}$.

The $\underline{w}$ and $\underline{u}$ components of $\underline{g}$ are the same for the NR and GN methods. The form of the $\underline{\Phi}$, $\underline{\Omega}$, and $\underline{\Psi}$ components of $\underline{R}$ differ in the NR and GN methods. The expressions for the NR method are developed first.

The gradient vector $\underline{g}$ is formed from the p x 1 vector $\underline{w}$ and the q x 1 vector $\underline{u}$. The j^{th} element of $\underline{w}$ and the m^{th} element of $\underline{u}$ are given by

$$[w]_j = \frac{1}{\sigma_v^2} \underline{v}^T \underline{L}^j \underline{B}^{-1} \underline{v} \tag{3.70}$$

and

$$[u]_m = -\frac{1}{\sigma_v^2} \underline{v}^T \underline{L}^m \underline{A}^{-1} \underline{v} \tag{3.71a}$$

$$= -\frac{1}{\sigma_v^2} \underline{v}^T \underline{L}^m \underline{B}^{-1} \underline{x} \,, \tag{3.71b}$$

for j = 1, .., p and m = 1, ..., q. The matrix $\underline{R}$ is partitioned into the p x p matrix $\underline{\Phi}$, the p x q matrix $\underline{\Omega}$, and the q x q matrix $\underline{\Psi}$. The elements of these three matrices are given by

$$[\Phi]_{jk} = \frac{1}{\sigma_v^2} \underline{v}^T \underline{B}^{T^{-1}} \underline{L}^{T^j} \underline{L}^k \underline{B}^{-1} \underline{v} \,, \tag{3.72}$$

$$[\Omega]_{jm} = -\frac{1}{\sigma_v^2} \underline{v}^T \underline{B}^{T^{-1}} \underline{L}^{T^j} \underline{L}^m \underline{A}^{-1} \underline{v} \tag{3.73a}$$

$$= -\frac{1}{\sigma_v^2} \underline{v}^T \underline{B}^{T^{-1}} \underline{L}^{T^j} \underline{L}^m \underline{B}^{-1} \underline{x} , \tag{3.73b}$$

$$[\Psi]_{mn} = \frac{1}{\sigma_v^2} \underline{v}^T \underline{A}^{T^{-1}} \underline{L}^{T^m} \underline{L}^n \underline{A}^{-1} \underline{v} \tag{3.74a}$$

$$= \frac{1}{\sigma_v^2} \underline{x}^T \underline{B}^{T^{-1}} \underline{L}^{T^m} \underline{L}^n \underline{B}^{-1} \underline{x} , \tag{3.74b}$$

for $j, k = 1, \ldots, p$ and $m, n = 1, \ldots, q$.

Now define the sequences

$$\underline{\zeta} = \underline{B}^{-1} \underline{v} , \tag{3.75a}$$

$$\underline{\gamma} = \underline{A}^{-1} \underline{v} = \underline{B}^{-1} \underline{x} . \tag{3.75b}$$

Using these definitions, (3.70) to (3.74) become

$$[w]_j = \frac{1}{\sigma_v^2} (\underline{L}^0 \underline{\zeta})^T (\underline{L}^j \underline{\zeta}) , \tag{3.76a}$$

$$[u]_m = -\frac{1}{\sigma_v^2} (\underline{L}^0 \underline{v})^T (\underline{L}^m \underline{\gamma}) , \tag{3.76b}$$

$$[\Phi]_{jk} = \frac{1}{\sigma_v^2} (\underline{L}^j \underline{\zeta})^T (\underline{L}^k \underline{\zeta}) , \tag{3.76c}$$

$$[\Omega]_{jm} = -\frac{1}{\sigma_v^2} (\underline{L}^j \underline{\zeta})^T (\underline{L}^m \underline{\gamma}) , \tag{3.76d}$$

$$[\Psi]_{mn} = \frac{1}{\sigma_v^2} (\underline{L}^m \underline{\gamma})^T (\underline{L}^n \underline{\gamma}) . \tag{3.76e}$$

With $\underline{A}$ and $\underline{B}$ evaluated using the present parameter

estimates, the nature of the expressions in (3.76) suggests the following procedure:

1) compute $\underline{\gamma} = \underline{B}^{-1}\underline{x}$;

2) compute $\underline{v} = \underline{B}^{-1}\underline{Ax} = \underline{AB}^{-1}\underline{x} = \underline{A\gamma}$;

3) compute $\underline{\zeta} = \underline{B}^{-1}\underline{v}$;

4) evaluate the elements of $\underline{w}$, $\underline{u}$, $\underline{\Phi}$, $\underline{\Omega}$, and $\underline{\Psi}$.

We must now determine what matrix operations such as $\underline{B}^{-1}\underline{v}$ and $\underline{A\gamma}$ imply in scalar equations. If $\underline{y}$ is the vector of the MA sequence y(k), k = 0, ..., N-1, then $\underline{y}$ is given by

$$\underline{y} = \underline{B}\,\underline{v} \ . \tag{3.77}$$

The scalar expression for y(k), an MA(p) process, is given in (3.29), with q = p. Imposing the initial conditions of v(k) = 0, k = 1-p, ..., -1, y(k) can be written as

$$y(0) = v(0) \ , \tag{3.78a}$$

$$y(k) = v(k) + \sum_{i=1}^{k} b(i)\, v(k-i), \quad k = 1, \cdots, p-1, \tag{3.78b}$$

$$y(k) = v(k) + \sum_{i=1}^{p} b(i)\, v(k-i), \quad k = p, \cdots, N-1. \tag{3.78c}$$

In this formulation the zero initial conditions are implicitly applied by the equations of (3.78). Since the matrix model (3.77) imposes the same zero initial conditions for v(k), (3.77) is equivalent to the scalar representation of (3.78).

Solving for v(k) in (3.78), we have

$$v(0) = y(0) \ , \tag{3.79a}$$

$$v(k) = y(k) - \sum_{i=1}^{k} b(i)\, v(k-i), \quad k = 1, \cdots, p-1, \tag{3.79b}$$

$$v(k) = y(k) - \sum_{i=1}^{k} b(i)\, v(k-i), \quad k = p, \cdots, N-1. \tag{3.79c}$$

Since this is equivalent to

$$\underline{v} = \underline{B}^{-1}\, \underline{y} \ , \tag{3.80}$$

we see that operating on a vector by the inverse of a matrix of the form of $\underline{B}$ is equivalent to the scalar operations in (3.79). The expressions in (3.78) and (3.79) are recursive in nature and imply zero initial conditions on the vector multiplied in the equivalent matrix formulations. The results of (3.78) and (3.79) can be applied to such expressions as occur in (3.75), with appropriate changes in notation.

Once the sequences $v(k)$, $\gamma(k)$, and $\zeta(k)$, $k = 0, \ldots, N-1$, have been determined using the procedures illustrated in (3.78) and (3.79), it is possible to determine the contents of $\underline{R}$ and $\underline{g}$ using (3.76). Noting that the equations of (3.76) are all of the same form, we will develop the scalar equation implied from these expressions by examining (3.76d) in detail. From previous discussions of the matrix lag operator $\underline{L}$, it can be seen that

$$\underline{L}^{j}\, \underline{\zeta} = [0 \cdots 0 \quad \zeta(0) \cdots \zeta(N-1-j)]^{T}$$

and

$$\underline{L}^m \underline{\gamma} = [0 \cdots 0 \quad \gamma(0) \cdots \gamma(N-1-m)]^T .$$

where $\underline{L}^j \underline{\zeta}$ and $\underline{L}^m \underline{\gamma}$ are both N x 1 vectors. The product $(\underline{L}^j \underline{\zeta})^T (\underline{L}^m \underline{\gamma})$ is a scalar which will be indicated by $R_{\zeta\gamma}(j,m)$. Assuming that $j \geq m$, $R_{\zeta\gamma}(j,m)$ is given by

$$R_{\zeta\gamma}(j,m) = \sum_{i=0}^{N-1-j} \zeta(i) \; \gamma(i+|j-m|) . \tag{3.81a}$$

If $j \leq m$, the expression for $R_{\zeta\gamma}(j,m)$ is

$$R_{\zeta\gamma}(j,m) = \sum_{i=0}^{N-1-m} \gamma(i) \; \zeta(i+|j-m|) . \tag{3.81b}$$

Using the appropriate sequences in the example relationship given in (3.81), the elements of $\underline{R}$ and $\underline{q}$ can be calculated. The iterative step is now made using (3.69). For details on the NR method, refer to Appendix A.

The preceding discussion developed the expressions for the matrix $\underline{R}$ and the vector $\underline{q}$ based on the NR method. In considering the GN method, we observe that $\underline{q}$ is identical to that obtained for the NR method. The elements of the matrices $\underline{\Phi}$, $\underline{\Omega}$, and $\underline{\Psi}$ of $\underline{R}$, however, are given by

$$[\Phi]_{jk} = \mathrm{tr}[(\underline{L}^j \underline{B}^{-1})^T (\underline{L}^k \underline{B}^{-1})] , \tag{3.82a}$$

$$[\Omega]_{jm} = -\mathrm{tr}[(\underline{L}^j \underline{B}^{-1})^T (\underline{L}^m \underline{A}^{-1})] , \tag{3.82b}$$

$$[\Psi]_{mn} = \mathrm{tr}[(\underline{L}^m \underline{A}^{-1})^T (\underline{L}^n \underline{A}^{-1})] , \tag{3.82c}$$

for j, k = 1, ..., p and m, n = 1, ..., q. In (3.82) the matrix $\underline{L}$ operates on the N x N matrices $\underline{A}^{-1}$ and $\underline{B}^{-1}$. The result, prior to taking the trace, is an N x N matrix. Two

facts, however, allow considerable savings in computation:

1) because of the form of $\underline{A}$ and $\underline{B}$, $\underline{A}^{-1}$ and $\underline{B}^{-1}$ are lower triangular with equal elements along each diagonal (see Appendix D);

2) only the main diagonal elements of the final matrix, $(\underline{L}^j\underline{B}^{-1})^T(\underline{L}^m\underline{A}^{-1})$, for example, need be computed since the trace operator uses only those elements.

Fact 1) above establishes that $\underline{A}^{-1}$ and $\underline{B}^{-1}$ are characterized by the elements of their first column. If $\underline{\alpha} = [\alpha(0) \ \cdots \ \alpha(N-1)]^T$ is the first column of $\underline{A}^{-1}$, then the $\alpha(k)$ are given by

$$\alpha(0) = 1 , \tag{3.83a}$$

$$\alpha(k) = -\sum_{i=1}^{k} a(i)\ \alpha(k-i),\ k = 1, \cdots, q-1, \tag{3.83b}$$

$$\alpha(k) = -\sum_{i=1}^{q} a(i)\ \alpha(k-i),\ k = q, \cdots, N-1. \tag{3.83c}$$

Likewise, the elements of the first column of $\underline{B}^{-1}$ are given by

$$\beta(0) = 1 , \tag{3.84a}$$

$$\beta(k) = -\sum_{i=1}^{k} b(i)\ \beta(k-i),\ k = 1, \cdots, p-1, \tag{3.84b}$$

$$\beta(k) = -\sum_{i=1}^{p} b(i)\ \beta(k-i),\ k = p, \cdots, N-1. \tag{3.84c}$$

Using (3.82b) to illustrate the meaning of the matrix operations, we need consider only the cases where the n^{th} row of $(\underline{L}^j\underline{B}^{-1})^T$ multiplies the n^{th} column of $(\underline{L}^m\underline{A}^{-1})$. The

n^{th} row of $(\underline{L}^j\underline{B}^{-1})^T$ is the 1 x N row vector $[0 \ldots 0\ \beta(0) \ldots \beta(N-n-j)]$, and the n^{th} column of $(\underline{L}^m\underline{A}^{-1})$ is the N x 1 column vector $[0 \ldots 0\ \alpha(0) \ldots \alpha(N-n-m)]^T$. Their product is the scalar c(n,n), given by

$$c(n,n) = \sum_{i=0}^{N-n-j} \beta(i)\ \alpha(i+|j-m|) \ , \quad j \geq m \ , \tag{3.85a}$$

$$c(n,n) = \sum_{i=0}^{N-n-m} \alpha(i)\ \beta(i+|j-m|) \ , \quad m \geq j \ . \tag{3.85b}$$

From (3.82b) the jm^{th} element of $\underline{\Omega}$ is

$$[\Omega]_{jm} = \sum_{n=1}^{N-k} c(n,n) \ , \tag{3.86}$$

where k = max(j,m). Additional computational savings can be achieved by combining (3.85) and (3.86). If $j \geq m$, we have

$$[\Omega]_{jm} = \sum_{n=1}^{N-j} \sum_{i=0}^{N-n-j} \beta(i)\ \alpha(i+|j-m|)$$

$$= \sum_{i=0}^{N-1-j} (N-j-i)\ \beta(i)\ \alpha(i+|j-m|) \ . \tag{3.87a}$$

For $m \geq j$, the result is

$$[\Omega]_{jm} = \sum_{i=0}^{N-1-m} (N-m-i)\ \alpha(i)\ \beta(i+|j-m|) \ . \tag{3.87b}$$

The elements of $\underline{R}$ are thus weighted correlations of the appropriate sequences.

Summarizing the operations required for the GN method, the sequences $\gamma(k)$, $v(k)$, and $\zeta(k)$, k = 0, ..., N-1, must be computed. The elements of the gradient vector $\underline{g}$ are

determined from correlations of these sequences, with (3.81) illustrating the form of this correlation. The GN method then requires the generation of the $\alpha(k)$ and $\beta(k)$ sequences using (3.83) and (3.84), respectively. Note that these sequences are not required in the NR method. From these two sequences, the elements of $\underline{R}$ are determined using the weighted correlation illustrated by (3.87).

The final step in each iteration for both the NR and GN methods is the estimation of the variance of v(k). In both methods this estimate is obtained using

$$\hat{\sigma}_v^2 = \frac{1}{N} \sum_{k=0}^{N-1} v^2(k) \ ,$$

where v(k) is the sequence generated in the GN and NR methods. It is an estimate of the unknown excitation sequence.

Unconditional Sum of Squares

Following the development in Box and Jenkins [10], the unconditional sum of squares procedure is presented here. Combined with a direct search of the parameter space for the optimal solution, this technique is used in this work to check the operation of the NR and GN methods for the ARMA(1,1) process. The description of the unconditional sum of squares approach presented here will be based on the ARMA(1,1) process.

The ARMA(1,1) model, with a(1) = a the AR coefficient and b(1) = b the MA coefficient, is represented by (3.36).

For a given set of estimates a and b, the conditional sum of squares (CSSQ) is computed using

$$S_*(\hat{a},\hat{b}) = \frac{1}{N} \sum_{k=0}^{N-1} \hat{v}^2(k) \ , \tag{3.88}$$

where v(k) is the estimate of the white noise excitation sequence v(k). This estimate is generated according to

$$\hat{v}(k) = x(k) + \hat{a}\, x(k-1) - \hat{b}\, \hat{v}(k-1) \ ,$$

for $k = 0, \ldots, N-1$. In certain cases, however, the transient effect imposed by the assumption of zero initial conditions for v(k) and x(k), $k < 0$, can have a strong effect on the value obtained for $S_*(\hat{a},\hat{b})$. An example of this is when the AR singularity lies close to the unit circle. To avoid or lessen the effects of this transient, the unconditional sum of squares (USSQ) is recommended.

The basis of the USSQ is the estimation of v(k) over the range $k = 0, \ldots, N-1$ and the prediction of v(k) for a few points outside that range. For example, v(k) might be estimated over the range $k = -10, \ldots, N+10$. Using an iterative algorithm, v(k) is re-estimated until the USSQ

$$S(\hat{a},\hat{b}) = \frac{1}{N} \sum_{k=0}^{N-1} \hat{v}^2(k) \ , \tag{3.89}$$

computed for each estimate of v(k), is stable. Details for implementing the USSQ are found in [10].

As mentioned previously, the USSQ is used to check the validity of the solution found by the NR or GN algorithm discussed in the preceding section. For the ARMA(1,1)

model, the acceptable range for the AR parameter a is $-1.0 < a < 1.0$. That for the MA parameter b is also $-1.0 < b < 1.0$. To verify the operation of the other estimation procedures, the USSQ is computed for each pair of $(\hat{a},\hat{b})$ values as $\hat{a}$ and $\hat{b}$ go from -1.0 to 1.0 by some fixed increment. A useful result of scanning the ARMA(1,1) parameter space and generating the USSQ at each point is the generation of the sum of squares cost function surface for each process and set of data analyzed. The shape of the cost function surface can provide information that can aid in predicting the behavior of the more efficient estimation routines. Note that this procedure is practical only in a parameter space of small dimensions.

"Shifted" Yule-Walker AR Estimates

If y(k) is the MA(p) portion of an ARMA(q,p) process, then $R_{yy}(k) = 0$ for $|k| > p$. This moving-average process is the weighted sum of the present and p previous random shocks. When the lag in the autocorrelation of y(k) exceeds p, there is no longer any overlap in the random shocks summed. The result is a zero autocorrelation value at that lag. This property is used in an ARMA process to estimate the AR parameters. If $k \geq p+1$, the autocorrelation of the ARMA(q,p) process x(k) satisfies the recursive relationship

$$-R_{xx}(k) = \sum_{i=1}^{q} a(i)\, R_{xx}(k-i) \quad . \tag{3.90}$$

By allowing k to run from p+1 to q+p, we obtain the set of equations:

$$\begin{aligned} -R_{xx}(p+1) &= a(1)\, R_{xx}(p) + \cdots + a(q)\, R_{xx}(p+1-q) \\ &\vdots \qquad\qquad \vdots \qquad\qquad\qquad \vdots \\ -R_{xx}(p+q) &= a(1)\, R_{xx}(p+q-1) + \cdots + a(q)\, R_{xx}(p) \end{aligned} \tag{3.91}$$

This system of q equations is linear in the q unknowns a(1), ..., a(q). With estimates for $R_{xx}(k)$, the autocorrelation function of the data x(k), (3.91) can be used to estimate the $\{a(i)\}_1^q$.

This approach is often proposed as a method for obtaining the initial estimates for the AR parameters in an ARMA estimation procedure. Hannan, for example, uses (3.91) as the first step in his estimation procedure [15]. In Chapter 4, the estimates obtained by this technique are compared to the estimates obtained from the NR method. This estimate is referred to as the "shifted" Yule-Walker estimate (SYW).

Noncausal Wiener Filter

In the next chapter, one of the estimation procedures used is the application of LPC to the data after it has been filtered to suppress the noise. The filter used is based on the noncausal formulation of the Wiener filter. If the additive noise to be suppressed is white, the transfer function of the filter is

$$H(\omega) = \frac{\Phi_{ss}(\omega)}{\Phi_{ss}(\omega) + \sigma_n^2}$$

$\Phi_{ss}(\omega)$ is the power spectrum of the AR process and σ_n^2 is the variance of the additive white noise. In computing $H(\omega)$, $\Phi_{ss}(\omega)$ is calculated using the parameters of the AR model. Hence, $\Phi_{ss}(\omega)$ is not an estimate. The impulse response of $H(\omega)$ is obtained and is then used to filter the data. The autocorrelation method of LPC is applied to the resulting sequence to estimate the AR parameters. This procedure is used in tests on the AR(1) process in Chapter 4.

Nonlinear Regression Algorithm to Improve the AR Estimates

It has been shown that the addition of white noise to an AR(q) process produces a data sequence x(k) that is described by the ARMA(q,q) model. Use of this model requires an ARMA parameter estimation procedure, producing estimates of the AR parameters $\{a(i)\}_1^q$, estimates of the MA parameters $\{b(i)\}_1^q$, and an estimate of σ_v^2, the variance of the ARMA model excitation sequence. As suggested by Pagano, these parameters are converted to the parameter set comprised of the AR coefficient estimates $\{a(i)\}_1^q$ and the MA autocorrelation estimates $\{R_{yy}(k)\}_0^q$. A nonlinear regression can then be used to improve the estimates of the $\{a(i)\}_1^q$, σ_ε^2, and σ_n^2, the parameters of the original AR model. In this section is a brief presentation of the

nonlinear regression technique. A more detailed development is found in Appendix B.

Letting $\underline{z} = [\hat{a}(1) \ldots \hat{a}(q)\ \hat{R}_{yy}(0) \ldots \hat{R}_{yy}(q)]^T$ and $\underline{\theta} = [\tilde{a}(1) \ldots \tilde{a}(q)\ \tilde{\sigma}_\varepsilon^2\ \tilde{\sigma}_n^2]^T$, then the 2q+1 equations relating the two parameter sets $\underline{z}$ and $\underline{\theta}$ can be written as

$$z_i = f_i(\underline{\theta}) + e_i \ , \tag{3.92}$$

for i = 1, ..., 2q+1. The metric [16] for evaluating the effectiveness of $\underline{\theta}$ in minimizing the sum of squares of the e_i is given by

$$Q(\underline{\theta}) = \sum_{i=1}^{2q+1} [z_i - f_i(\underline{\theta})]^2 \tag{3.93}$$

Using (3.32c) to define the $f_i(\underline{\theta})$, i = 1, ..., 2q+1, gives the following set of equations:

$$\hat{a}(i) = \tilde{a}(i) + e_i \ , \tag{3.94a}$$

$$\hat{R}_{yy}(0) = \tilde{\sigma}_\varepsilon^2 + \tilde{\sigma}_n^2 \sum_{i=0}^{q} \tilde{a}^2(i) + e_{q+1} \ , \tag{3.94b}$$

$$\hat{R}_{yy}(k) = \tilde{\sigma}_n^2 \sum_{i=0}^{q-|k|} \tilde{a}(i)\ \tilde{a}(i+k) + e_{q+k+1} \ , \tag{3.94c}$$

for i, k = 1, ..., q and $\tilde{a}(0) = 1$. The $\{\tilde{a}(i)\}_1^q$, $\tilde{\sigma}_\varepsilon^2$, and $\tilde{\sigma}_n^2$ are chosen to minimize $Q(\underline{\theta})$, given in (3.93). Because of the nonlinear nature of the functions $f_i(\underline{\theta})$ in (3.94), an iterative procedure based on the GN method or modified GN method is used. The Gauss-Newton method is based on the linearization of the nonlinear functions $f_i(\underline{\theta})$ about $\underline{\theta}$. This will yield a solution $\underline{\theta}^*$ to (3.92) having the property

of convergence for a finite number of functional relationships $f_i(\underline{\theta})$. The $\underline{\theta}^*$ will be asymptotically efficient [17].

Characterization of the Noise Sequences

Some of the most important assumptions made about the AR-to-ARMA transformation model concern the statistical properties of the white noise sequences $\varepsilon(k)$, $v(k)$, and $n(k)$. In defining any ARMA(q,p) process it is customary to use an i.i.d. $N(O,\sigma_v^2)$ excitation sequence. This implies the "whiteness" of the sequence. It has been pointed out that the data used in this work to test the validity of the model and the operation of the algorithms are generated from known AR(q) models. The excitation sequences for the AR(q) processes and the ARMA(q,q) simulations, $\varepsilon(k)$ and $v(k)$, respectively, must be reasonable approximations to ideal white noise sequences.

This is also true for the sequence $n(k)$, the additive white noise. Addition of white noise to an AR(q) process theoretically results in an ARMA(q,q) process with the AR parameters unchanged. If the additive noise $n(k)$ is non-white, however, the resulting data, while still an ARMA process, will no longer have the same AR parameters. If $n(k)$ is non-white, further processing must be performed on the AR parameter estimates to retrieve the original AR(q) parameters. This problem is beyond the scope of this work.

Two approaches for generating the required noise files

have been used. In the early stages of algorithm development, the noise obtained by digitizing the output of an analog noise generator is used for the three noise sequences. For small sequences and for use in algorithm development, this approach is adequate. Unfortunately, these noise files have a sample power spectrum that decays slightly near the folding frequency, defined as one half of the sampling frequency. This is due to the anti-aliasing filter used prior to digitization. Some of the data in the following chapter is based on ARMA processes generated using this digitized analog noise. The noise used in these tests has been modified to reduce the effects of the anti-aliasing pre-filter. In effect, the signal is "resampled" at a lower frequency, below the pre-filter cutoff frequency, by using every other sample in the sequence.

Most of the parameter estimation statistics reported in the next chapter are taken from data sequences synthesized using noise samples derived from the FORTRAN software random number generator RAN. This number generator provides samples from a uniform distribution on the range [0,1], i.e., U[0,1]. Since samples of a random variable (r.v.) with a normal distribution are desired, the samples taken from RAN must be manipulated to achieve the correct distribution. By summing several samples taken from RAN and scaling appropriately, the resulting sample

r.v. approximates one taken from the desired normal distribution.

If u(k) is a sample of the U[0,1] r.v. and w is the desired sample from a r.v. with normal distribution $N(\mu_w, \sigma_w^2)$, then w can be approximated by

$$w = c + d \sum_{k=1}^{n} u(k) , \tag{3.95}$$

where n is the number of samples summed to approximate a normal distribution. A value of 10 is used for n in this work. The constants c and d in (3.95) scale and shift the sum to achieve the desired mean μ_w and variance σ_w^2. These constants are given by

$$c = \mu_w - [3n\sigma_w^2]^{1/2} ,$$

$$d = \frac{2}{n} [3n\sigma_w^2]^{1/2} .$$

The desired noise sequences $\varepsilon(k)$, v(k), and n(k) are formed by appending large numbers of the w generated by (3.95).

As will be seen in data reported in the next chapter, the noise sequences generated in this manner do not exhibit the decay at the folding frequency in the sample power spectrum. There may, however, be problems associated with software generated random numbers. One of the most serious defects as it affects this work would be periodic behavior in the samples generated by RAN. A large number of samples are needed for the tests in Chapter 4, with each sample requiring several values of the function RAN. Periodic

behavior in the noise sequences would effectively reduce the size of the tests. Verification tests performed on the noise sequences generated from RAN are reported in the next chapter.

In using the noise files generated according to (3.95) it is necessary to scale the noise sequences to achieve the appropriate sample variance needed in the test. This is especially true for the sequences n(k) and v(k). The estimator for the variance of the sequence n(k), for example, is given by

$$\hat{\sigma}_n^2 = \frac{1}{M} \sum_{k=1}^{M} n^2(k) - \bar{n}^2 , \tag{3.96}$$

where $\bar{n}$ is the sample mean

$$\bar{n} = \frac{1}{M} \sum_{k=1}^{M} n(k) . \tag{3.97}$$

M in (3.96) and (3.97) is the number of samples in the entire noise sequence, not the number of samples in a frame of data.

After scaling, the sequences are used to synthesize ARMA processes. The sample variance of these processes is also calculated and compared to the theoretical value for that process. Because of the importance of the sample variance estimator in checking the validity of a process, measures of its reliability are needed. The measures used are the mean and variance of the estimator in (3.96). If estimating σ_n^2, for example, $E[\hat{\sigma}_n^2]$ and $var[\hat{\sigma}_n^2]$ are the

desired reliability statistics. These statistics are now developed for the estimator given in (3.96)

Taking the expected value of (3.96) gives

$$E[\hat{\sigma}_n^2] = \frac{1}{M} \sum_{k=1}^{M} E[n^2(k)]$$

$$= \sigma_n^2 ,$$

since n(k) is a zero mean process and $E[n(k)] = 0$. Thus, $\hat{\sigma}_n^2$ as defined in (3.96) is an unbiased estimator of σ_n^2. Using the property that the n(k) are i.i.d., we have

$$E[(\hat{\sigma}_n^2)^2] = \frac{1}{M^2} E[\sum_{k=1}^{M} \sum_{j=1}^{M} n^2(k)\ n^2(j)]$$

$$= \frac{1}{M^2} \sum_{k=1}^{M} E[n^4(k)] + \frac{M-1}{M} \sigma_n^4 . \qquad (3.98)$$

$E[n^4(k)]$, the fourth moment of the normal r.v. n(k), is given by

$$E[n^4(k)] = 3\ \sigma_n^4 ,$$

as developed in Appendix E. Equation (3.98) becomes

$$E[(\hat{\sigma}_n^2)^2] = \frac{3}{M} \sigma_n^4 + \frac{M-1}{M} \sigma_n^4 ,$$

and

$$\text{var}[\hat{\sigma}_n^2] = E[(\hat{\sigma}_n^2)^2] - (E[\hat{\sigma}_n^2])^2$$

$$= (\frac{3}{M} + \frac{M-1}{M} - 1)\sigma_n^4$$

$$= \frac{2}{M} \sigma_n^4 \qquad (3.99)$$

is the desired expression. Use is made of (3.96) and the statistics of (3.97) and (3.99) in Chapter 4.

CHAPTER 4

EXPERIMENTAL RESULTS

Introduction

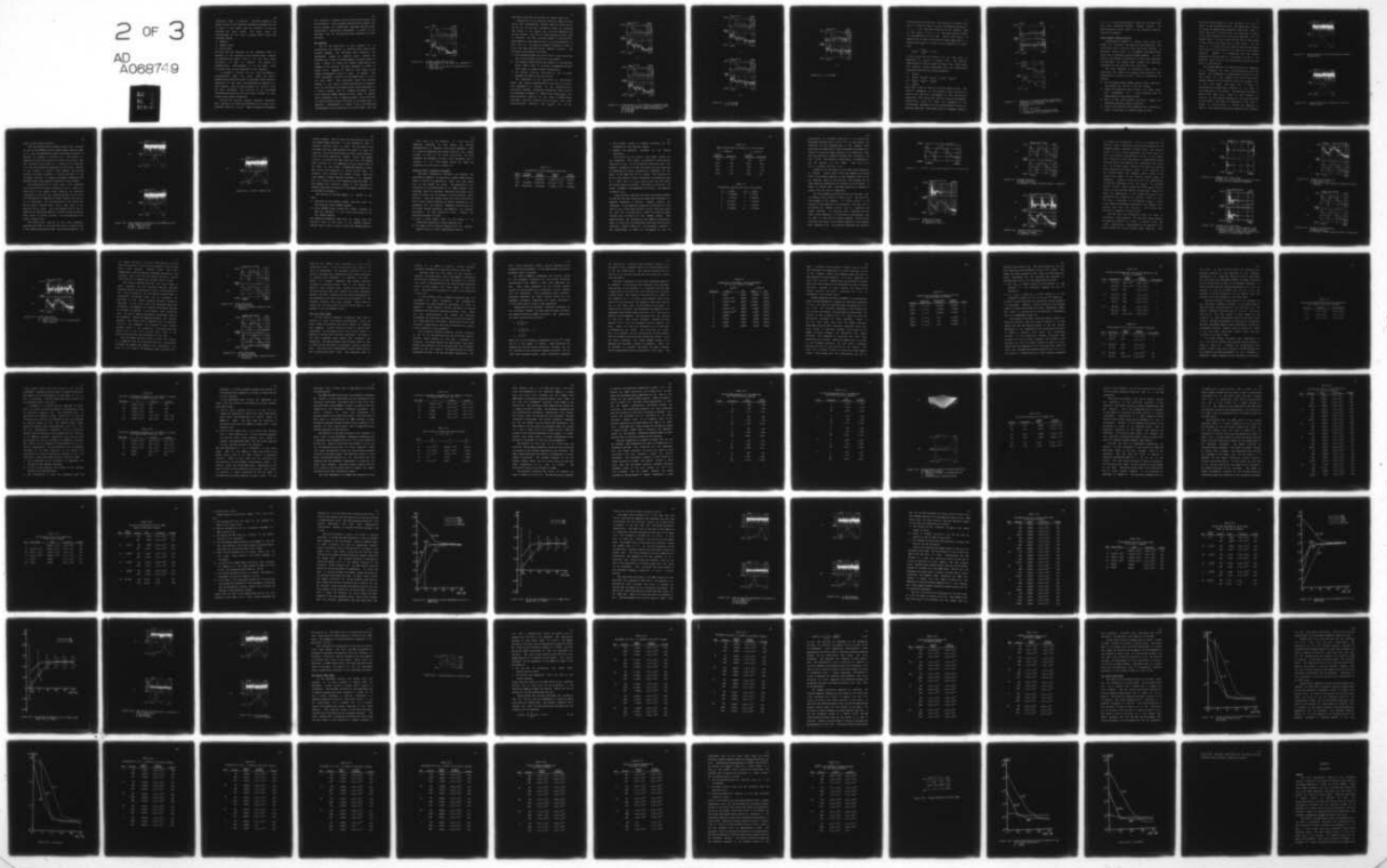
In this chapter the data from various experiments will be presented. The theoretical basis for these experiments is discussed in the preceding chapter. The first section presents results obtained from the autocorrelation method of LPC as applied to one frame of voiced speech. For a variety of signal-to-noise ratios, the sample spectrum determined from the LPC coefficient estimates illustrates the degrading effects of additive white noise. That section also shows the effects of using the simple autocorrelation correction method discussed in Chapter 2. The implications of the uncorrelated signal and noise assumption are discussed.

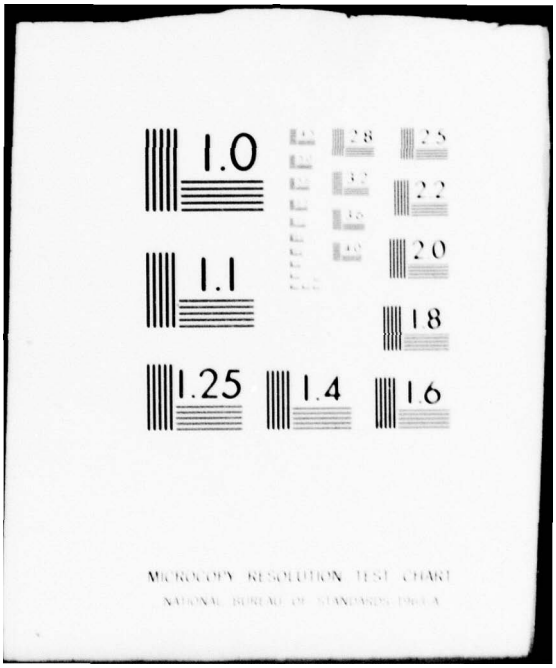
Because several noise sequences are required in testing the parameter estimation methods, a section on the characteristics of the noise files used is included. Sample power spectra and time domain amplitude histograms are shown. The sample variance required for a noise sequence in a given test is listed in the section describing that test.

The next section presents data obtained using

AD-A068 749 UTAH UNIV SALT LAKE CITY DEPT OF COMPUTER SCIENCE F/G 17/2
ESTIMATION OF THE PARAMETERS OF AN AUTOREGRESSIVE PROCESS IN TH--ETC(U)
DEC 78 W J DONE N00173-77-C-0041
UNCLASSIFIED UTEC-CSC-79-021 NL

2 OF 3
AD A068749




1.0
1.1
1.25
1.4
1.6
2.8
3.2
3.6
2.5
2.2
2.0
1.8
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Steiglitz's mode 1 algorithm. Desirable because of the simple nature of the algorithm, parameter estimates for the 10 pole, 2 zero model used by Steiglitz in [35] are obtained for three cases. The three cases are distinguished by the type of sequence used to excite the "unknown" system:

1) impulse;

2) impulse train;

3) white noise.

For each case, the spectrum of the estimated model is compared to the spectrum of the original model. Results are excellent for cases 1) and 2). For the white noise excitation of case 3), however, the results are disappointing. Unfortunately, it is the white noise excitation that is most important to this research.

In Chapter 3, details for the AR(1)-to-ARMA(1,1) transformation model are given. After the section demonstrating Steiglitz's mode 1 algorithm, there follows a comparison of the mode 1 and NR methods as applied to an AR(1) process. Data are then obtained for various AR(1) and MA(1) processes. The validity of the AR-to-ARMA transformation model for the first order process is then tested using several estimators.

The last two sections present parameter estimation data obtained by using the NR method on two higher order processes, where q = 2 and 4. These estimates are compared

with estimates computed using the shifted Yule-Walker and LPC procedures. Using distance measures which combine the errors between the coefficient estimates and the actual coefficients, significant improvement is shown in the estimates from the NR algorithm when compared to the LPC estimator.

LPC Analysis

One of the objectives of this research is to characterize the effects of additive white noise on LPC analysis systems. The following data illustrate the degradation caused by additive noise. Results are presented for a frame of voiced speech at varying levels of noise. Figure 4-1 shows the speech frame used as the example in this section. The time waveform is shown in Figure 4-1a). Sampled at 6667 Hz, this frame of 128 samples corresponds to about 19 msec. of speech. This frame represents a portion of the schwa vowel /ə /, as in the word "rust". This particular vowel was selected because of the nearly uniform distribution of formants. Also, on a dB scale the formants drop in peak magnitude at a nearly constant rate as frequency increases. Figure 4-1b) shows the sample spectrum of this frame of speech, after windowing with a Hamming window. On the dB scale, the nearly uniform formant structure of the schwa vowel is apparent. Superimposed on Figure 4-1b) is the spectrum corresponding to a 10 pole LPC fit of this frame. The LPC

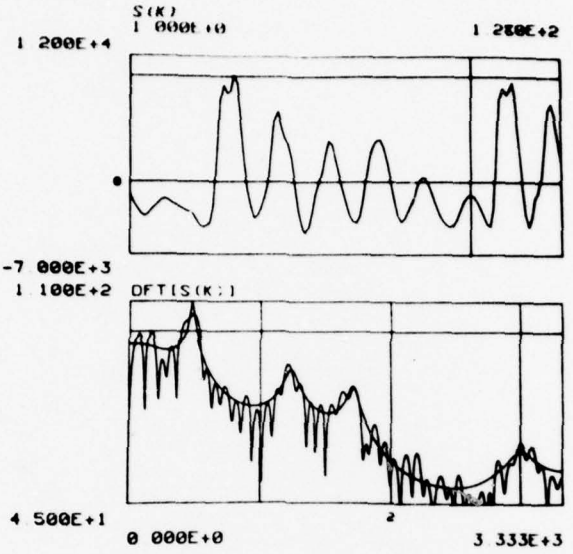


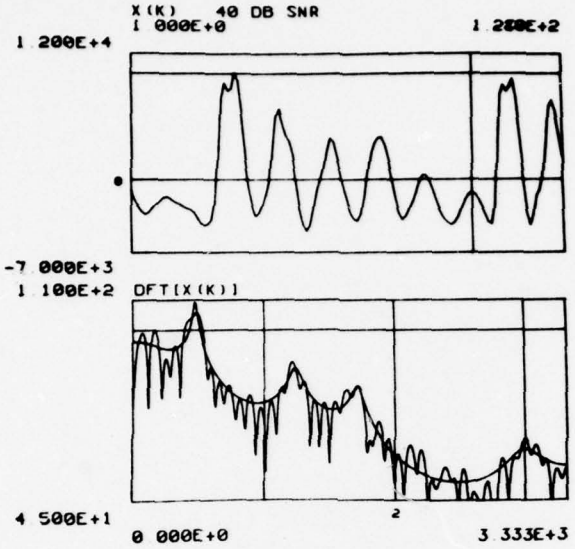
Figure 4-1: Example frame used as s(k)
a) 128 samples of the vowel /ə/, sampled at 6667. Hz
b) Spectrum of /ə/ and a 10 pole LPC fit to that spectrum

spectrum is smoother and matches the formant peaks well.

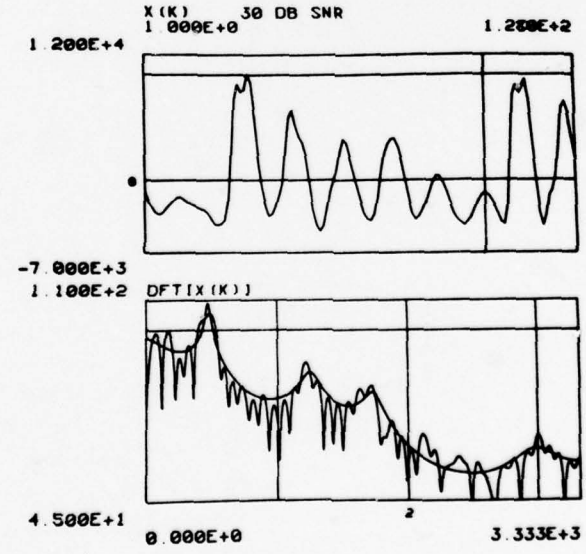
Figures 4-2, a)-e), show the effects of additive white noise with progressively smaller signal-to-noise ratios: 40, 30, 20, 10, and 0 dB. The SNR is found by averaging the energy in the speech and the noise sequences over several seconds. The ratio of these energies is then used to determine the SNR, defined as SNR = $\sum s^2(k) / \sum n^2(k)$. Superimposed on each spectral plot is the corresponding 10 pole LPC fit. All spectral graphs in Figures 4-1 and 4-2 are on the same scale and can be compared directly. The following noise effects are noted:

1) with decreasing SNR, the noise "floor" rises, obscuring more of the formant structure of the speech;
2) the formants identified by LPC analysis in increasingly poorer SNR's tend to be wider in bandwidth and have their peaks at slightly higher frequencies;
3) the formant structure identified by LPC is badly degraded for SNR's below about 20 dB.

The importance of the assumption of uncorrelated signal and noise is demonstrated in the next set of data. This assumption is primary to the autocorrelation correction methods of parameter estimation, some of which are discussed in Chapter 2. Figure 4-3a) shows $R_{ss}(k)$, the autocorrelation function for the frame of speech being discussed. Plotted in Figure 4-3b) are $R_{nn}(k)$, the noise autocorrelation function, and $R_{sn}(k)$, one of the

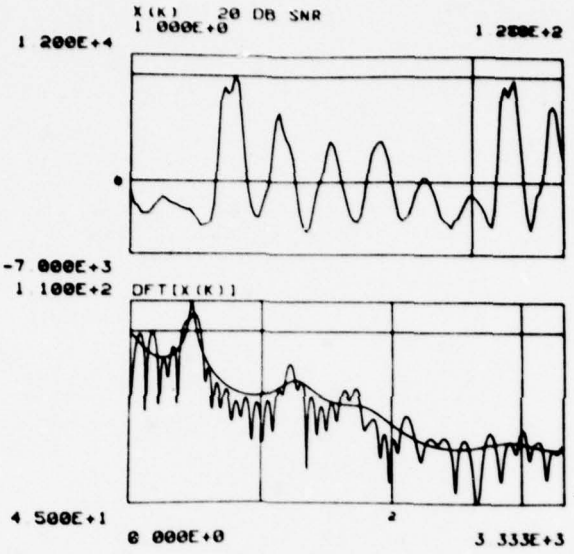


a)

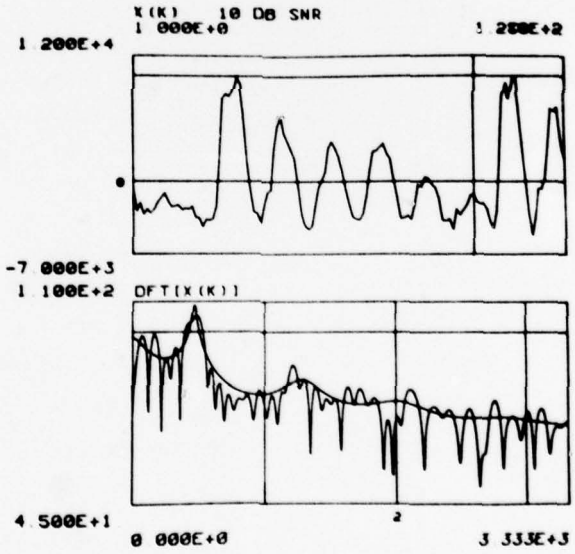


b)

Figure 4-2: Illustrations of the effects of additive white noise on the example speech frame and 10 pole LPC approximations to resulting spectrum
a) 40 dB SNR
b) 30 dB SNR

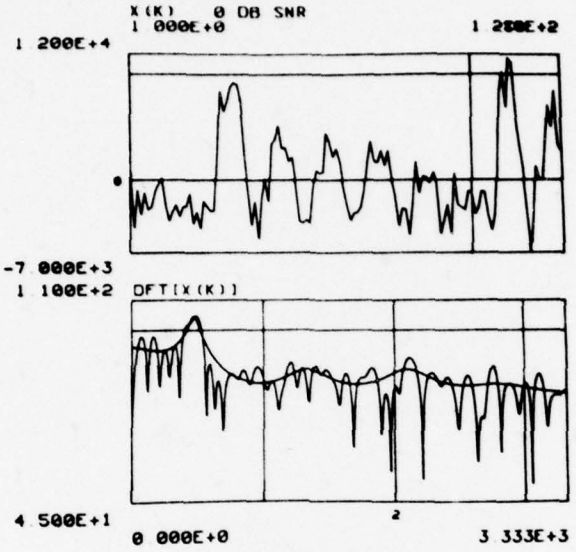


c)



d)

Figure 4-2: c) 20 dB SNR
d) 10 dB SNR



e)

Figure 4-2: e) 0 dB SNR

cross-correlation functions. The abscissa in Figures 4-3a) and b) starts at lag k = 0 and is followed by 50 lags for k = 1, ..., 50. The last 50 points are the negative lags in the order k = -50, ..., -1. The noise used for Figure 4-3 corresponds to a 10 dB SNR. Noting that $R_{sn}(k)$ in Figure 4-3b) is that curve with the larger magnitude, it is obvious that $R_{sn}(k) \neq 0$, based on the estimation of $R_{sn}(k)$ from

$$R_{sn}(k) = \sum_{i=0}^{N-1-k} s(i)\ n(i+k)$$

In fact, $R_{sn}(k)$ for this frame is of the same order of magnitude as $R_{ss}(k)$ in Figure 4-3a). The spectral implications of this are shown in Figure 4-3c), which shows four spectral curves determined from LPC coefficients calculated from the four autocorrelations:

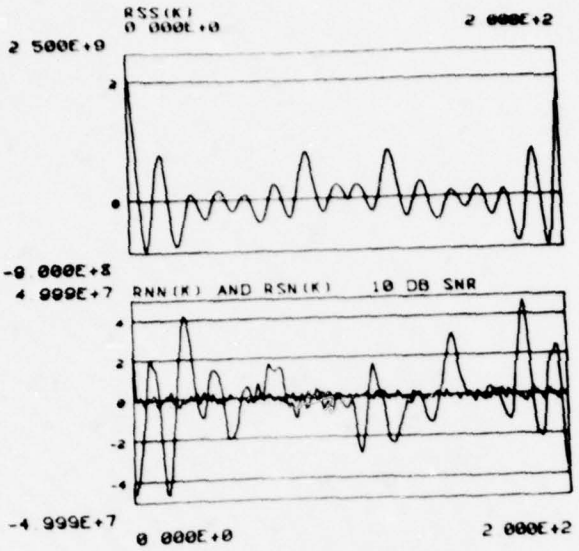
i) $R_{ss}(k)$,

ii) $R_{ss}(k) = R_{xx}(k) - R_{nn}(k) - R_{sn}(k) - R_{ns}(k)$,

iii) $\hat{R}_{ss}(k) = R_{xx}(k) - R_{nn}(k)$,

iv) $R_{xx}(k)$.

Note that i) and ii) result in the same spectral plot. The explicit assumption of uncorrelated signal and noise is used in iii), while iv) corresponds to LPC coefficients determined from noisy data, with no correction attempted. Figure 4-3c), curve iii, shows the inadequacy of the uncorrelated assumption for the autocorrelation correction modeling approach. Even though curve iii appears superior



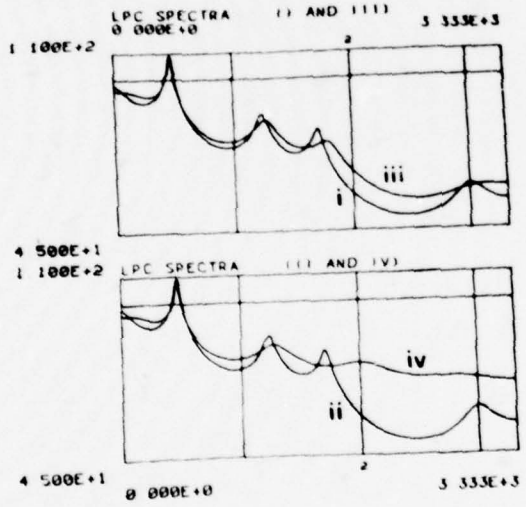


Figure 4-3: Comparison of autocorrelation and cross-correlation sequences for the sample frame of speech at a 10 dB SNR

a) $R_{ss}(k)$

b) $R_{nn}(k)$ and $R_{sn}(k)$

c) 10 pole LPC spectra based on auto-correlations i-iv discussed in text

to iv, in a large percentage of frames the LPC algorithms will fail, producing unstable inverse filters. An autocorrelation matrix which is not positive definite causes this problem.

Noise Sequence Characteristics

As mentioned in Chapter 3, several noise files are needed for excitation and additive noise sequences. The noise files used in this work come from three sources: 1) digitized analog noise, 2) "resampled" digitized analog noise, and 3) software generated random numbers. In this section the methods of generation of these noise sequences are discussed. The sample power spectra and amplitude histograms for the noise files are also given.

The first approach to generating noise files is digitizing the output of an analog noise generator. The procedure for creating the noise files is summarized as follows:

1) set the General Radio Company Random Noise Generator, type 1390-B, No. SGL-78, at the 20 KC range;
2) adjust the generator output controls and audio panel gain adjustments for a noise envelope that is approximately 8 volts peak-to-peak;
3) with a 3.2 KHz anti-aliasing pre-filter, sample the amplified generator output at 6667 Hz;
4) rescale the digitized noise sequences for a variance of about 1.0 and store in unpacked format on disk.

Two noise files obtained by this procedure are used in tests in the following sections. The upper plot in Figure 4-4 shows the sample power spectrum in dB of one frame of noise from one of these files. The lower plot in Figure 4-4 is the histogram of the time series amplitude for the example frame. For this figure and the rest of the figures in this section, the frame size is 1000 points, the DFT order is 11, and 40 cells are used to form the histogram. From Figure 4-4, note the near flat character of the sample spectrum. However, at frequencies near the folding frequency of 3333 Hz there is a noticeable roll-off in the spectrum. This is due to the anti-aliasing pre-filter used prior to digitization.

The second approach for generating the noise sequences is to "resample" the noise files obtained by the first method. This resampling is accomplished by taking as a new time series all noise samples with an even time index. Another sequence can be formed by taking the samples with an odd time index. The upper plot in Figure 4-5 illustrates the sample power spectrum for one frame of noise generated in this manner. The amplitude histogram is shown in the lower plot. There is less tendency for roll-off at the folding frequency for the noise generated by this method. The effect of resampling in this case is similar to pre-filtering at 3333 Hz and sampling at 3333 Hz. This causes aliasing and eliminates the spectral

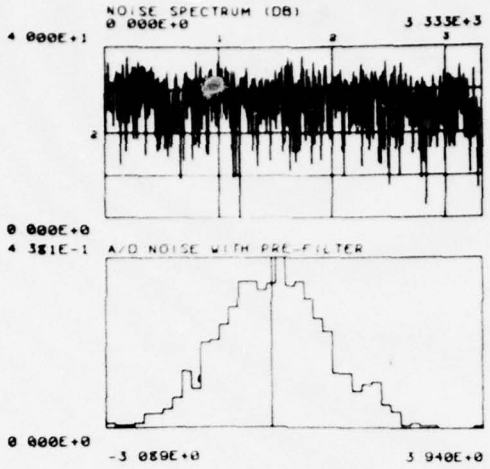


Figure 4-4: Noise generated by digitizing pre-filtered analog noise

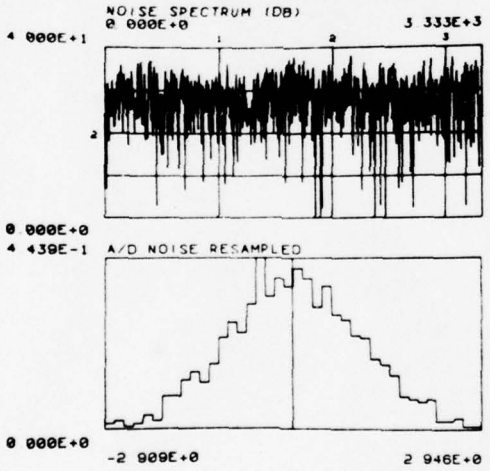


Figure 4-5: Noise generated by resampling digitized analog noise.

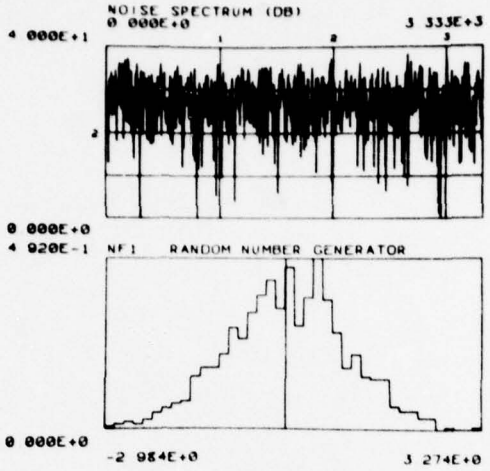
decay at the folding frequency.

The last method used to generate noise files involves the use of the FORTRAN uniform random number generator RAN. A r.v. with the appropriate normal distribution $N(0,\sigma^2)$ is obtained by summing n uniform r.v.'s. The uniform r.v. samples are scaled and shifted to achieve the correct mean and variance in the normal r.v. A value of 10 is used for n for the noise sequences generated for this work. Details on the creation of normal noise samples from the RAN function are found in Chapter 3. Most of the data in the following sections is based on processes synthesized and degraded using noise sequences obtained in this fashion.

The characteristics of three noise files generated with this method are given in Figure 4-6 a)-c). Part a) of Figure 4-6 is the sample power spectrum and amplitude histogram of noise file one. Abbreviated NF1, this file is used exclusively to generate the AR(q) process s(k) that is to be identified. The characteristics of the second file NF2 are given in Figure 4-6b). NF2 is used only as the additive white noise. The third sequence NF3 is used to generate the equivalent ARMA(1,1) process created by adding white noise to an AR(1) process. Its characteristics are shown in Figure 4-6c).

One further test required for the noise sequences generated from RAN is the verification that no cycles occur in the numbers generated by RAN. The function RAN has one

a)



b)

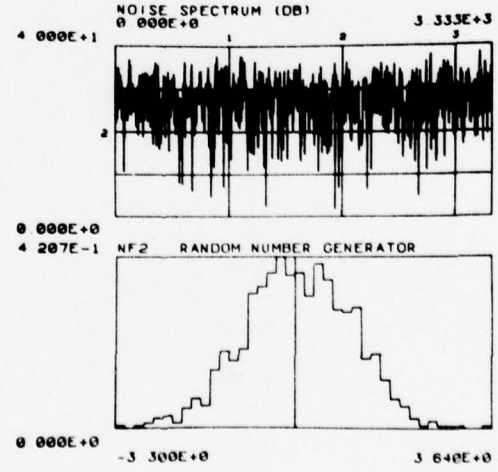


Figure 4-6: Noise sequences generated from FORTRAN uniform random number generator
a) NF1 - used as $\epsilon(k)$
b) NF2 - used as $n(k)$

c)

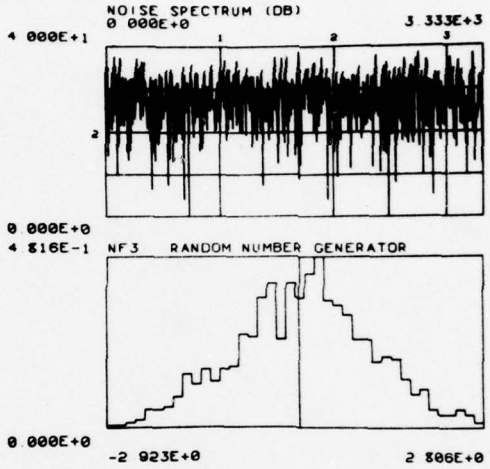


Figure 4-6: c) NF3 - used as v(k)

integer argument. This integer sets the starting value for the random number generator. If that argument is zero, a standard starting point is used. For the tests in the following sections, approximately 500 frames of data are required. With 256 points/frame and three noise sequences, 3.84×10^5 noise samples are needed. Since each sample requires ten values from RAN, there are 3.84×10^6 samples from a uniform r.v. To compensate for data values found at the start of each frame, 4300800 RAN samples are used. This yields three sequences of 143360 points each. In analysis, this provides 518 frames of data. With this large number of values required from RAN, it is important that no cycles occur in the sequence produced by RAN. If this happens, the effective length of the sequence is reduced, degrading the quality of the sample statistics for the parameter estimators.

The validity of the RAN sequence is checked in two ways:

1) determine if the standard integer starting value for RAN reoccurs within 4300800 samples;
2) after 4000000 values, record the integer argument of RAN and determine if that value occurs again in the next 300800 samples.

The sequence produced by RAN does not repeat from the starting value at any point in the 4300800 samples. Neither does it enter a cycle of less than 300800 points at

a later point in the sequence. It is felt the noise sequences generated in this manner are suitable approximations to Gaussian white noise sequences. Table 4-1 lists the beginning and ending values for the integer argument of RAN for the three noise files. All three sequences are designed to be zero mean processes with a variance of 1.0. The sample values for these two parameters are also given in Table 4-1.

Steiglitz Mode 1 Iterative Procedure

The mode 1 technique by Steiglitz and McBride [36] described in Chapter 3 is basically a system identification method in which it is assumed the input v(k) and output x(k) of the system are known. For application to the estimation of the coefficients of an ARMA process, this technique must be modified. When only the output of the system is known, v(k) is assumed to be the Kronecker delta function. Also, the system output x(k) may be modified so that it more closely resembles an impulse response, as the assumption for v(k) implies. Suggested modifications for the signal include windowing, pre-emphasis, and cepstral processing. To test Steiglitz's mode 1 method, the following procedure is used:

1) From a known model, which is the system to be identified, generate an output sequence x(k).
2) The input to the "unknown" system is one of: impulse, impulse train, or noise (approximately white).

Table 4-1

Generation of Noise Sequences Using RAN

Noise File	Starting Integer	Ending Integer	Sample Mean	Sample Variance
NF1	0	50312698	-1.69150×10^{-3}	1.00341
NF2	50312698	1254307719	-8.71251×10^{-4}	1.00257
NF3	1254307719	357121965	9.29829×10^{-4}	0.998244

3) Use the mode 1 method to compute estimates for the parameters of the "unknown" system.
4) Compare the parameter estimates to the design parameters.

The results for one 10 pole, 2 zero model system are now presented. This model is identical to that used by Steiglitz in [35]. In [35] Steiglitz arbitrarily sets the sampling frequency at 15 KHz and enters the pole locations by specifying the center frequency and bandwidth for the poles in the upper half of the Z-plane. The location of a pole in the Z-plane in polar coordinates is determined from $R = 1 - BW/2$ and $\theta = 2 f_c/f_s$, where R and θ are the radius and radian angle of the pole. The terms f_c and BW are the center frequency and bandwidth of the pole. The sampling frequency is f_s.

Since the work reported here began with speech sampled at 6667 Hz, the pole locations are specified according to a sampling frequency of 6667 Hz. However, because the assignment of the sampling frequency is arbitrary for this type of test, either specification of pole locations yields the same set of coefficients. Table 4-2 gives the upper Z-plane pole locations for the example system. Both specifications are provided for the reader's convenience. Table 4-3 lists the 10 coefficients resulting from the pole locations listed in Table 4-2. As discussed in Chapter 3, the coefficients in Table 4-3 correspond to the AR

Table 4-2

Upper Z-Plane Pole Locations for 10 Pole Model

f_s = 15000 Hz		f_s = 6667 Hz	
Center Frequency	Bandwidth	Center Frequency	Bandwidth
270	60	120	26.7
2290	100	1018	44.4
3010	120	1338	53.3
3500	175	1556	77.8
4500	281	2000	125.

Table 4-3

Denominator Coefficients for 10 Pole Model

k	a(k)	k	a(k)
1	-3.300959	6	13.53270
2	7.222431	7	-9.888342
3	-11.62311	8	5.741208
4	14.69756	9	-2.461647
5	-15.58842	10	0.7301360

coefficients of the model, with a(0) = 1. By solving the polynomial equation $p(z) = \sum a(i)\ z^{-i} = 0$, one obtains the locations of the singularities in the Z-plane. These locations are identical to those determined from Table 4-2. Figure 4-7 shows the model spectrum to be identified. Note that the zeros are a complex conjugate pair located on the unit circle. The numerator coefficients, corresponding to the MA portion of the model, are: b(0) = 1.0, b(1) = -1.414214, and b(2) = 1.0.

Figure 4-8 is the output of this model when excited by an impulse. Figure 4-8a) is the time sequence and Figure 4-8b) is the sample spectrum of that sequence in dB, as are all spectral plots in this section. The estimate for the model spectrum produced by one iteration of this method is shown in Figure 4-9b). The model spectrum is repeated in Figure 4-9a) for ease of comparison.

The time sequence produced by exciting the model with an impulse train is shown in Figure 4-10a). The period for this example is 100 samples. In Figure 4-10b) is the estimate of the spectrum of this process. The data x(k) is multiplied by a Hamming window prior to computing the sample spectrum. In using the mode l technique for this type of time sequence, it is desirable to pre-process x(k) to make it more like an impulse response. Figure 4-11a) shows the real part of the complex cepstrum of the windowed data sequence, x(k). By properly windowing this cepstrum,

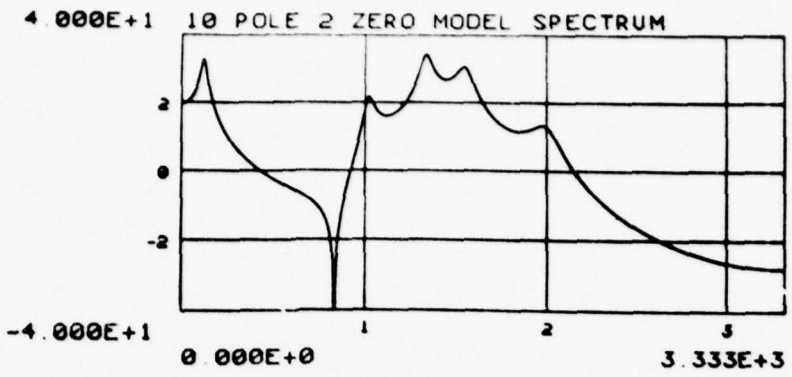


Figure 4-7: 10 Pole, 2 Zero Model Spectrum to be identified

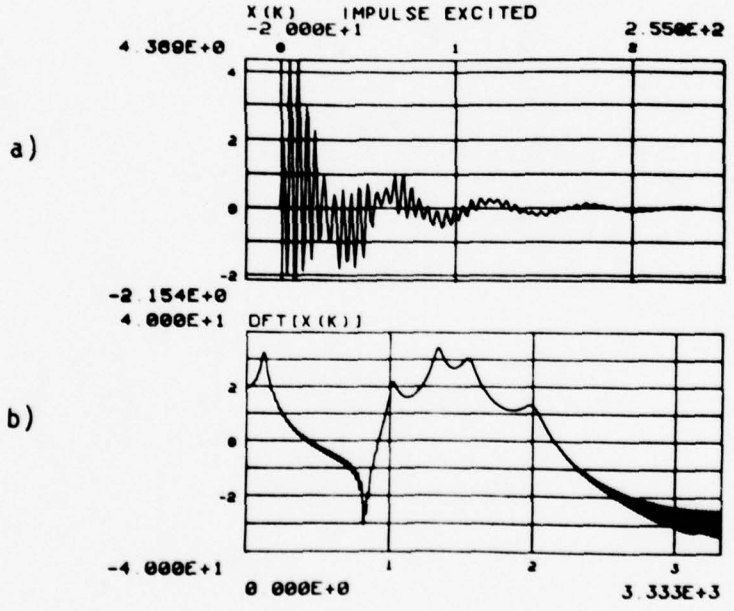
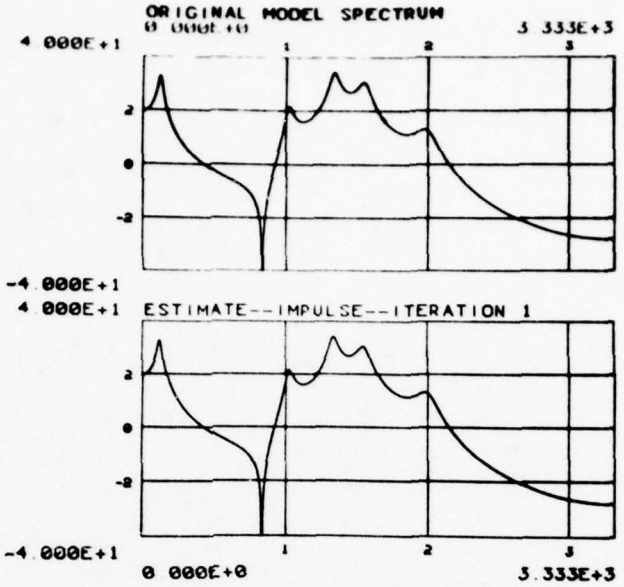


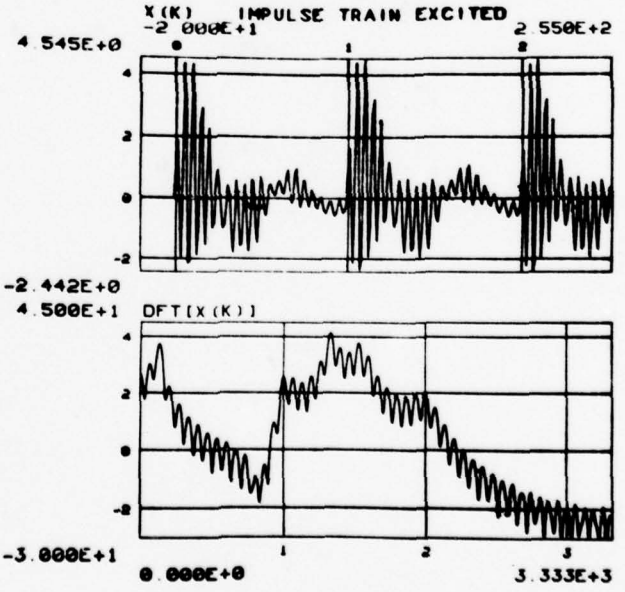
Figure 4-8: Impulse Excitation
a) Output of model
b) Spectrum of part a)



a)

b)

Figure 4-9: Impulse Excitation
a) Model Spectrum
b) Estimate of Model Spectrum after 1 iteration



a)

b)

Figure 4-10: Impulse Train Excitation
a) Output of model
b) Spectrum of part a)

two things are accomplished. First, by eliminating the cepstral spikes resulting from the pitch harmonics in the frequency domain, apparent in Figure 4-11a), the periodic nature of x(k) can be suppressed. The second step is to force this cepstral representation of x(k) to be causal. Upon returning to the time domain, if appropriate scaling has been done in the cepstrum, the resulting time series will be minimum phase. Figure 4-11b) shows the cepstrum after windowing and scaling. Figure 4-12b) contains the new minimum phase time sequence, while Figure 4-12a) contains the output of the impulse excited model for comparison. Figures 4-13a) and 4-13b) are, respectively, the spectral estimates of x(k) and $x_{mp}(k)$, the modified version of x(k). Note in Figure 4-12 that $x_{mp}(k)$ is quite similar to x(k) from the impulse excited case. Figure 4-13 shows the suppression of the harmonic structure on the spectrum of x(k) caused by the periodic nature of x(k). The mode 1 technique is now applied to $x_{mp}(k)$. Figure 4-14b) shows the estimated spectrum for the impulse train excited case after two iterations. The original model spectrum is repeated in Figure 4-14a).

The last case to be considered is when the model is excited by a noise sequence. The resulting output sequence and spectral estimate are shown in Figures 4-15a) and 4-15b), respectively. Superimposed on the spectrum of the noise excited x(k) is the original model spectrum. Note

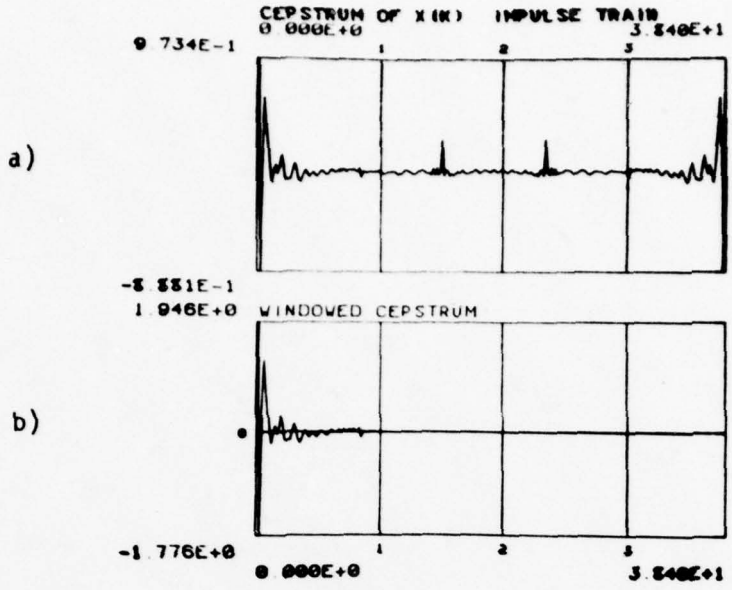


Figure 4-11: Impulse Train Excitation
a) Real part of the Complex Cepstrum of x(k)
b) Part a) after Cepstral Windowing

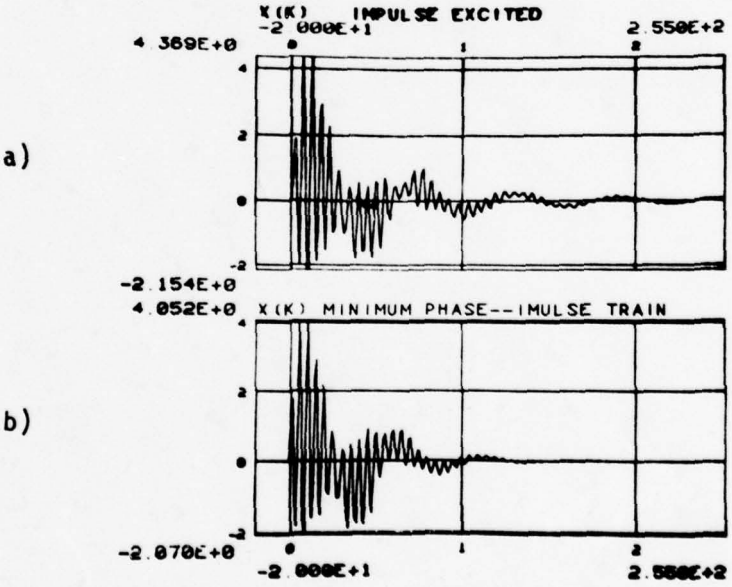


Figure 4-12: Impulse Train Excitation
a) Output of model with an impulse input
b) Modified system output after cepstral modifications to remove periodicity and simulate a minimum phase signal.

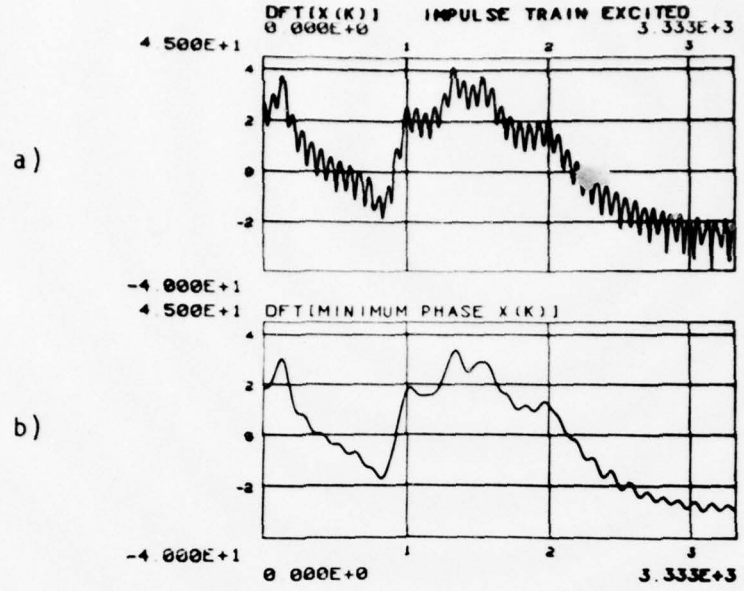


Figure 4-13: Impulse Train Excitation
a) Spectrum of x(k)
b) Spectrum of the cepstrally modified version of x(k)

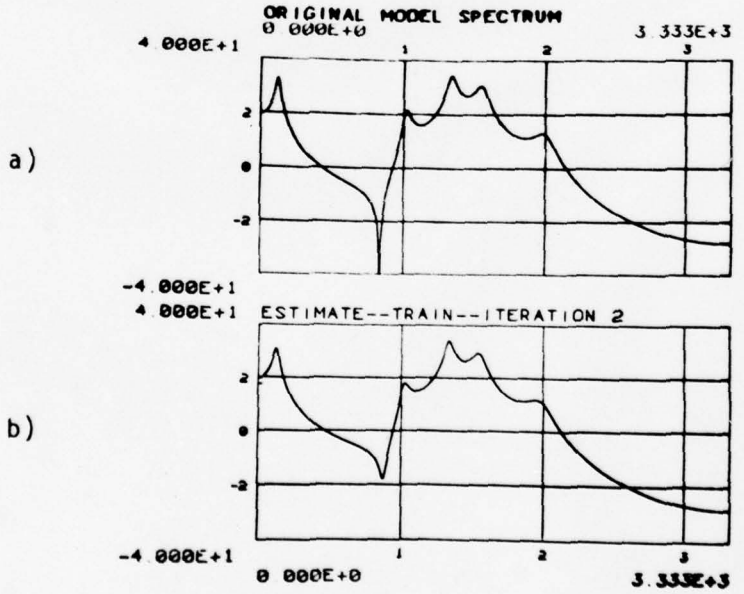


Figure 4-14: Impulse Train Excitation
a) Model spectrum
b) Estimate of model spectrum after 2 iterations

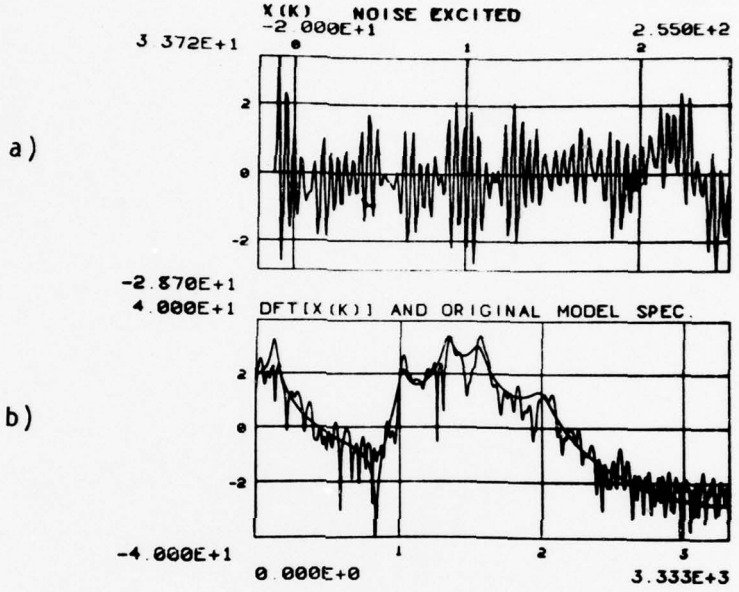


Figure 4-15: Noise Excitation
a) Output of model
b) Spectrum of part a) with superimposed model spectrum

the random variations from that ideal spectrum resulting from the deviation of the excitation sequence from an ideal white noise process. Figures 4-16b) and 4-17b), respectively, show the spectral estimates produced by this technique after the first and second iterations. Further iterations fail to improve the estimate, which is poor.

The results presented above for the three types of input represent the best possible spectral estimates obtained through the appropriate choice of modifications to x(k) prior to analysis. For the impulse excited model, no intermediate processing steps such as windowing are performed on x(k) before applying the mode l estimation algorithm. In fact, use of any of the suggested operations (windowing, pre-emphasis, or cepstral processing) degrades the parameter estimates. In the impulse train excitation case, the use of cepstral processing considerably improves the estimates. Adding pre-emphasis degrades the estimate somewhat. The initial step of windowing x(k) is necessary. If x(k) is not windowed, the estimated filter becomes unstable within a few iterations. No modifications are made to x(k) in the noise excited case. None of the options given provide any improvement in this case.

From the sample spectra in Figures 4-9b) and 4-14b), it is apparent that this technique does well in estimating the model for impulse and impulse train excitation. The error in the case of the impulsive input is nearly zero.

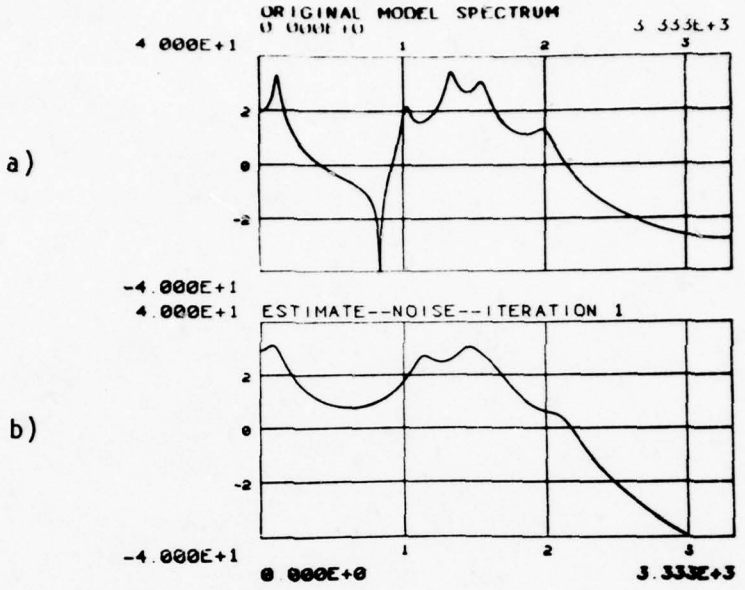


Figure 4-16: Noise Excitation
a) Model spectrum
b) Estimate of model spectrum after 1 iteration

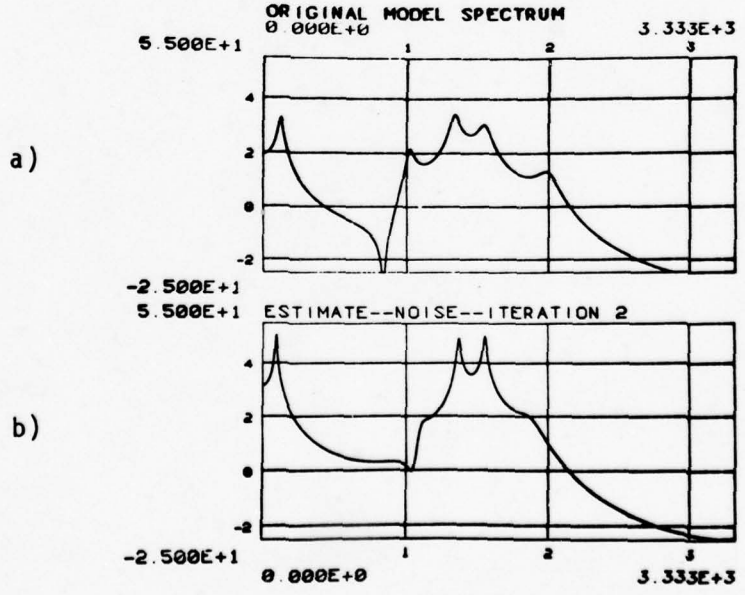


Figure 4-17: Noise Excitation
a) Model spectrum
b) Estimate of model spectrum after 2 iterations

That for the impulse train excitation is well within acceptable limits. In the noise excited case, however, the error is unacceptable. The estimates obtained for noise excited processes were consistently poor, often converging to unstable filter estimates. In addition, the mode 1 method is strongly dependent upon double precision arithmetic to achieve success, even in the impulse and impulse train excited cases. Because of the results for the noise excited case, this method has been discarded. However, the mode 1 estimate for the single AR parameter of the processes in the next section will be compared to the estimate obtained from other methods. This is done to ensure that the large value of 10 for the AR order in this example is not the dominant factor.

The First Order Model

In this section parameter estimation data from a variety of first order models are presented. First, the mode 1 scheme by Steiglitz and the NR method by Anderson are compared. An AR(1) process is used for testing these algorithms. In the second set of tests, the NR method is applied to several AR(1) and MA(1) processes. The tests on the AR(1) processes also produce LPC estimates for comparison. The data obtained from the MA(1) tests is used to check the NR algorithm's performance against previous work. The third set of experimental results deals with an AR(1) process plus white noise. The resulting data is

modeled as an ARMA(1,1) process. Several parameter estimation procedures are applied to data of this type.

The data base for the tests in this section is composed of various AR(1), MA(1), and ARMA(1,1) processes. The excitation sequences for these processes are derived from noise files generated by the three methods discussed earlier in this chapter. All three types of noise are used. The method of generation is noted in the discussion of each test.

In the following tests, parameter estimation data are presented for seven estimators. The mathematical development for these estimation procedures is given in Chapter 3. Three methods provide estimates for the AR parameters and excitation sequence variance, only. These are: the autocorrelation LPC method, the "shifted" Yule-Walker LPC technique, and Wiener filtering followed by the autocorrelation LPC method. Abbreviated as LPC, SYW, and W-LPC, respectively, these three methods do not require initial guesses for the parameters.

Four estimation methods considered provide estimates of the AR and MA parameters and the excitation sequence variance. These methods are the mode 1 procedure of Steiglitz, the Newton-Raphson and Gauss-Newton time domain methods from Anderson, and the unconditional sum of squares method by Box and Jenkins. The abbreviations for these techniques are mode 1, NR, GN, and USSQ, respectively. The

first three procedures require initial guesses for the parameters being estimated. In the USSQ method, the entire parameter space is scanned.

For tests on ARMA(1,1) sequences the initial values for the parameter estimates are the actual AR and MA coefficients of the ARMA(1,1) model that describes the data. Using these parameters as the initial guesses removes all uncertainty due to inaccurate initial guesses from the experiment. Tests performed on AR(1) and MA(1) processes sometimes use other initial values for the parameter estimates, especially all zeros. The type of initial guess used is noted in each experiment.

Comparisons of the parameter estimates generated by the preceding methods are made using the mean, variance, and standard deviation sample statistics. The expressions for these statistics are given by

$$\hat{\mu} = \frac{1}{M} \sum_{i=1}^{M} \hat{c}(i) \ ,$$

$$\hat{\sigma}^2 = \frac{1}{M} \sum_{i=1}^{M} \hat{c}^2(i) - \hat{\mu}^2 \ ,$$

$$\hat{\sigma} = [\hat{\sigma}^2]^{1/2} \ ,$$

where $\hat{c}(i)$ is the estimate of parameter c at the i^{th} frame and M is the number of frames. These statistics are computed for each AR and MA coefficient estimate and for the estimate of the excitation sequence variance. For the first order processes tested, these statistical measures

are sufficient. In higher order processes, however, it is desirable to use a combined measure for the AR coefficients or the MA coefficients. Two distance measures will be defined in the sections dealing with the second and fourth order AR models.

The mode 1 procedure is the first estimation method to be examined. In the section that discusses this algorithm, found earlier in this chapter, it is apparent that the mode 1 method does not do well when the excitation for the model is white noise. The results in that section are based on the analysis of a 10 pole, 2 zero system. Part of the reason for the failure of this method in that case could be due to the high order of the AR part of the model. To examine this, the mode 1 and NR methods are applied to data generated from an AR(1) model with a(1) = 0.5. No noise is added to the AR sequence. The excitation for the process is from NF1. The initial parameter estimate for both methods is a(1) = 0.5, the parameter used to generate the data. Table 4-4 lists the estimates for a(1) from these two methods for two frames. Ten iterations are given. Note the estimate for a(1) generated by the NR procedure does not change in the five most significant figures after the first iteration. For these example frames, it is apparent that the mode 1 method is inadequate. While the NR estimate is accurate and stable, the mode 1 estimate varies considerably from one iteration to the next. The

Table 4-4

Comparison of the Mode 1 and NR Estimates for a(1) of an AR(1) Process

	Frame 1		Frame 2	
Iteration	Mode 1	NR	Mode 1	NR
1	.20713	.52667	-.05109	.49538
2	$.27195 \times 10^{1}$	.52667	1.4135	.49538
3	$-.12183 \times 10^{1}$	.52667	-1.2294	.49538
4	.12604	.52667	-5.5827	.49538
5	$-.15914 \times 10^{1}$	.52667	-.24169	.49538
6	$-.10244 \times 10^{-6}$	.52667	.89261	.49538
7	-.90959	.52667	.97296	.49538
8	-.77367	.52667	.99496	.49538
9	-.49257	.52667	.99732	.49538
10	.83789	.52667	.99728	.49538

mode 1 estimate does converge in frame 2, but to a value that indicates the singularity is close to the unit circle in the Z-domain. Hence, the estimated model will tend to have marginally stable behavior. It is also important to note that all mode 1 computations are performed in double precision, while those for the NR method are done in single precision. Because of these results and those of the earlier section, the mode 1 technique will not be considered in any further tests.

Before examining the AR(1)-to-ARMA(1,1) transformation model, the NR procedure is applied to three AR(1) processes (a = 0.1, 0.5, and 0.9) and three MA(1) processes (b = 0.1, 0.5, 0.9). For the AR(1) processes, an LPC estimate for a(1) is also calculated. The processes in these tests are excited by NF1 scaled for $\hat{\sigma}_\varepsilon^2 = 1.0$. Table 4-5 lists the theoretical variances for these processes, based on $\sigma_\varepsilon^2 = 1.0$ This table also lists the sample variances determined from the data sequences generated using NF1. Using (3.99) to calculate $\text{var}[\hat{\sigma}_s^2]$, the variance of the sample variance estimate, the theoretical and sample variances are compared by observing how close these quantities are in value. Taking the square root of $\text{var}[\hat{\sigma}_s^2]$ as the standard deviation, $\hat{\sigma}_s^2$ lies within d standard deviations of σ_s^2, the true variance of the process. The value for d is given in Table 4-5 under the column labeled "Limit". This assumes that the distribution for $\hat{\sigma}_s^2$ is

Table 4-5

Theoretical and Sample Variances for the AR(1) and MA(1) Processes

Process	Value of Coefficient	Theoretical Variance	Sample Variance	Limit
AR(1)	a = 0.1	1.01010	1.00867	-1
AR(1)	a = 0.5	1.33333	1.32308	-3
AR(1)	a = 0.9	5.26316	5.16576	-5
MA(1)	b = 0.1	1.01	1.01140	1
MA(1)	b = 0.5	1.25	1.25700	2
MA(1)	b = 0.9	1.81	1.82259	2

Gaussian with a mean of σ_s^2. The initial guess for a(1) and b(1) required by the NR method is zero in all cases. The number of frames analyzed is 518. There are 256 points per frame. Convergence for the iterative NR procedure is achieved when both of the following conditions hold:

1) $|\hat{\sigma}_i^2 - \hat{\sigma}_{i-1}^2| < 0.0001$, where $\hat{\sigma}_i^2$ is the estimate of the variance of the excitation sequence at the i^{th} iteration;

2) $|\hat{c}_i - \hat{c}_{i-1}| < 0.0001\ (\hat{c}_{i-1} + 0.001)$, where $\hat{c}_i$ is the estimate of either a(1) or b(1) at the i^{th} iteration.

If both of these conditions are satisfied, the NR procedure is terminated. However, a limit is placed on the maximum number of iterations allowed per frame. Designated ITMAX, this limit is usually set at 30 iterations. Convergence test 2) is suggested by Bard [6].

The results of the AR(1) tests are found in Table 4-6. Those for the MA(1) tests are listed in Table 4-7. In both tables note that the sample variance of the NR estimate decreases as the magnitude of the coefficient increases. In Chapter 3, equation (3.51) gives the theoretical variance of the conditional maximum likelihood (CML) estimate for a(1) of an AR(1) process. The variance of the estimate for b(1) of an MA(1) process is given in (3.52). Noting that the expressions are of the same form, Table 4-8 lists the variance and standard deviation for a, b = 0.1, 0.5, and 0.9. A frame size of N = 256 is used in computing

Table 4-6

NR and LPC Estimates of a(1) and NR Estimate of σ_v^2 for AR(1) Processes

a(1)	Estimator	Sample Mean	Sample Variance	Convergence
0.1	NR a(1)	$.967 \times 10^{-1}$	4.09×10^{-3}	3
	LPC a(1)	$.991 \times 10^{-1}$	6.65×10^{-3}	
	NR σ_v^2	.993	7.54×10^{-3}	3
0.5	NR a(1)	.492	3.26×10^{-3}	3
	LPC a(1)	.492	5.35×10^{-3}	
	NR σ_v^2	.995	7.54×10^{-3}	3
0.9	NR a(1)	.891	9.33×10^{-4}	3
	LPC a(1)	.886	1.78×10^{-3}	
	NR σ_v^2	$.101 \times 10^{1}$	8.44×10^{-3}	3

Table 4-7

NR Estimate of b(1) and σ_v^2 for MA(1) Processes

b(1)	Estimator	Sample Mean	Sample Variance	Convergence
0.1	NR b(1)	.103	4.42×10^{-3}	9
	NR σ_v^2	.993	7.54×10^{-3}	9
0.5	NR b(1)	.505	3.46×10^{-3}	7
	NR σ_v^2	.994	7.53×10^{-3}	7
0.9	NR b(1)	.891	1.38×10^{-3}	16
	NR σ_v^2	$.101 \times 10^{1}$	8.67×10^{-3}	16

this data. As seen from this table, the variance of the estimate improves as the magnitude of the parameter approaches 1.0. Comparing Tables 4-6 and 4-7 with Table 4-8, one finds close agreement between the sample variances and the theoretical variance of the parameter estimate. In Table 4-6 note that the average LPC estimate for a(1) is superior to the NR estimate only for a(1) = 0.1. The variance of the NR estimate for a(1) is smaller than the variance of the LPC estimate in all three cases.

The entry "NR σ_v^2" in both tables is the estimate of the excitation sequence variance. In this case, with no additive noise, the excitation is $\varepsilon(k)$, with $\hat{\sigma}_\varepsilon^2 = 1.0$. The poorest estimate (occurring when $a = 0.9$ or $b = 0.9$) is in error by 1%. The last column in both tables is the iteration at which convergence occurs for the first frame of data. The MA(1) processes all require more iterations. The AR(1) processes require three iterations to satisfy the convergence criteria. However, there is usually no change in the five most significant figures after the first iteration in the AR(1) cases.

In a study of MA(1) processes with coefficients of 0.2, 0.5, and 0.9, Nelson [28] presents results similar to those in Table 4-7. Although his investigation uses smaller frame sizes, the variance of his CML estimate for b(1) exhibits the same improvement as b(1) increases in magnitude. Nelson compares several estimators in his work.

Table 4-8

Variance of CML Estimate for the Coefficient of an AR(1) or an MA(1) Process

a or b	var[$\hat{a}$]	st. dev.[$\hat{a}$]
0.1	3.87×10^{-3}	6.22×10^{-2}
0.5	2.93×10^{-3}	5.41×10^{-2}
0.9	7.42×10^{-4}	2.72×10^{-2}

In his studies, Nelson finds that the CML and ML (maximum likelihood) estimators perform best on sequences generated from models with the MA parameter in the range of 0.5 to 0.9 in magnitude. The CML method mentioned is most like the NR method used in this work.

In Chapter 3 it is shown that the addition of white noise to an AR(1) process introduces an MA parameter b and an excitation sequence v(k). The resulting data is described by the ARMA(1,1) model. Tables 3-2, 3-3, and 3-4 illustrate how b and σ_v^2, the variance of v(k), are affected by varying levels of n(k), the additive noise. These tables are based on AR(1) processes with a(1) = 0.1, 0.5, and 0.9, respectively. Equations (3.53) a)-c) give the variance and co-variance of the estimates for a(1) and b(1) of an ARMA(1,1) process. Tables 4-9, 4-10, and 4-11 list the variance for the a(1) estimate of an ARMA(1,1) process. The values of a(1) assumed for these tables are 0.1, 0.5, and 0.9, respectively. The MA coefficient b(1) is computed according to the SNR's specified in the tables. The coefficient b(1) and the theoretical variance of b(1) are also given. From these tables several trends are noted:

1) For a fixed SNR, as a(1) increases in magnitude, the variance of the estimate decreases.
2) If a(1) is held constant, the variance of the estimate worsens as the SNR decreases.
3) The introduction of even the relatively small MA

Table 4-9

Variance of Parameter Estimates of the ARMA(1,1) Process for Various SNR's and a(1) = 0.1

SNR (dB)	b	var[$\hat{a}$]	var[$\hat{b}$]
30	$.10091 \times 10^{-3}$	.387	.391
20	$.99990 \times 10^{-3}$	.394	.398
10	$.91667 \times 10^{-2}$	.468	.473
0	$.50126 \times 10^{-1}$	$.154 \times 10^{1}$	$.155 \times 10^{1}$
-10	$.90917 \times 10^{-1}$	$.460 \times 10^{2}$	$.461 \times 10^{2}$

Table 4-10

Variance of Parameter Estimates of the ARMA(1,1) Process for Various SNR's and a(1) = 0.5

SNR (dB)	b	var[$\hat{a}$]	var[$\hat{b}$]
30	$.66556 \times 10^{-3}$	$.117 \times 10^{-1}$	$.157 \times 10^{-1}$
20	$.65577 \times 10^{-2}$	$.120 \times 10^{-1}$	$.159 \times 10^{-1}$
10	$.57331 \times 10^{-1}$	$.141 \times 10^{-1}$	$.187 \times 10^{-1}$
0	.26795	$.408 \times 10^{-1}$	$.505 \times 10^{-1}$
-10	.45573	.891	.942

parameter at 30 dB noticeably degrades the variance of the estimate when compared to the data in Table 4-8 for an AR(1) process.

In the experiments that follow, two approaches are used to test the validity of the model for an AR(1) process plus white noise:

1) Given an AR(1) process with $a = 0.5$ and $\sigma_\varepsilon^2 = 1.0$, compute the parameters b and σ_v^2 that result from adding white noise. The parameters a, b, and σ_v^2 define an ARMA(1,1) model. The data used for analysis is obtained by exciting this ARMA(1,1) model with a noise sequence v(k).
2) Given an AR(1) model with $a = 0.5$, excite this process with a noise sequence $\varepsilon(k)$. To the resulting sequence s(k) add the white noise sequence n(k), scaled to achieve the appropriate SNR. The time series obtained is x(k), the data available for analysis.

The test described in 1) is a simulation of the noise model. That is, the ARMA(1,1) model that results from adding white noise to an AR(1) process is generated directly. Tests of this type will be referred to as simulations of the AR-to-ARMA transformation model. For brevity, these are called ARMA tests. Experiments of the type in item 2) above represent the situation that occurs in practice: a signal described by the AR model is corrupted by the actual addition of white noise. In the

discussion that follows, type 2) experiments are referred to as AR+N tests.

The ARMA and AR+N tests offer two methods for checking the validity of the noise model and the usefulness of the estimation algorithms. The former type of test, with data generated from a known ARMA(1,1) model, is used to test the capability of the parameter estimation procedure. The latter category of tests then determines the appropriateness of the ARMA noise model. Since the data in both tests is theoretically an ARMA process, a given estimation procedure should produce similar results in the analysis of each type of data. This is supported by the results presented in this section.

The behavior of the AR(1) process in the presence of white noise is now considered. Parameter estimation data is obtained from processes with a(1) = 0.5 as the single AR coefficient. Using the equations from Chapter 3 for b_- and σ^2_{v-}, but henceforth omitting the minus subscript, Table 4-12 gives the parameter values for this process corrupted by noise at the SNR's shown. The numerical values in this table are derived assuming $\sigma^2_\epsilon = 1.0$. From Table 4-12, one observes that the parameter b approaches a in value as the noise level worsens. The data listed in Tables 4-10 and 4-12 is used in the following tests to compare the sample statistics with their theoretical values.

The first experiment is an ARMA test simulation of the

Table 4-11

Variance of Parameter Estimates of the ARMA(1,1) Process for Various SNR's and a(1) = 0.9

SNR (dB)	b	var[$\hat{a}$]	var[$\hat{b}$]
30	$.46922 \times 10^{-2}$	$.918 \times 10^{-3}$	$.483 \times 10^{-2}$
20	$.43330 \times 10^{-1}$	$.934 \times 10^{-3}$	$.491 \times 10^{-2}$
10	.25884	$.106 \times 10^{-2}$	$.522 \times 10^{-2}$
0	.62679	$.189 \times 10^{-2}$	$.604 \times 10^{-2}$
-10	.83588	$.111 \times 10^{-1}$	$.176 \times 10^{-1}$

Table 4-12

AR(1) Process Corrupted by Additive Noise

$a = 0.5$, $\sigma_\varepsilon^2 = 1$

SNR	σ_n^2	b	σ_v^2
∞	0	0	1.0
30	$1.\dot{3} \times 10^{-3}$	$.66556 \times 10^{-3}$	1.0017
20	$1.\dot{3} \times 10^{-2}$	$.65577 \times 10^{-2}$	1.0166
10	$1.\dot{3} \times 10^{-1}$	$.57331 \times 10^{-1}$	1.1628
0	$1.\dot{3}$	.26795	2.4880
-10	$1.\dot{3} \times 10^{1}$	.45573	14.628

AR(1) process with a 0 dB SNR noise level. From Table 4-12, the parameters of the resulting ARMA(1,1) process are: a(1) = 0.5, b(1) = 0.26795, and σ_v^2 = 2.4880. This ARMA model is excited by digitized analog noise (DAN) in one case and "resampled" digitized noise (RDAN) in the second case. The excitation noise sequences are scaled for a sample variance of 2.4880. The parameter estimates from two procedures are considered: USSQ and NR. Five frames, 256 points each, are analyzed by each procedure. No mean correction is performed. The initial parameter estimates for the NR method are the actual model parameters. The USSQ method initially scans the parameter space for values of a and b from -1.0 to 1.0 in steps of 0.02. When the minimum of the surface generated in this manner is found, another scan takes place in a small neighborhood of the minimum in increments of 0.001 for both parameters.

The purpose of this test is to verify that the iterative NR procedure produces a solution which minimizes the energy in the residual generated by that solution. The USSQ method generates the surface corresponding to a large number of solutions. The minimum obtained by the two methods should agree and be in the vicinity of the true model parameters: a = 0.5 and b = 0.26795. The theoretical value of the minimum is 2.4880.

The parameter estimates from these two methods are given in Tables 4-13 and 4-14. The DAN excitation sequence

is used for the experiment summarized in Table 4-13. The results for RDAN excitation are in Table 4-14. In most cases the estimates agree, especially for σ_v^2. Only in two frames do the results differ greatly. These are frames 3 and 4 of Table 4-14. The most likely explanation is that the NR method occasionally oscillates between two solutions. The termination procedure for this algorithm does not account for this kind of problem at present. As a result, the procedure may be terminated at the wrong solution. Figure 4-18a) is an isometric plot of the surface generated by the USSQ method for frame 5 of Table 4-14. Cross-sections taken through the minimum are shown in Figure 4-18b). These plots illustrate the quadratic nature of the surface. Estimation methods based on least squares seek to find the minimum of this surface.

The next comparison to be made is between the NR and GN methods. Both procedures are used in an ARMA test at a simulated 0 dB SNR. The conditions for the experiment are: 518 frames analyzed, 256 points per frame, no mean correction, and true parameter values for initial estimates. The excitation sequence is NF3 scaled for a sample variance of 2.4880. The results, listed in Table 4-15, show the GN method produces parameter estimates somewhat closer to the true values of a(1) = 0.5, b(1) = 0.26795, and σ_v^2 = 2.4880. However, the sample variance of the GN method is larger, indicating a wider

Table 4-13

NR and USSQ Analysis of 0 dB ARMA Test
with DAN Excitation Sequence

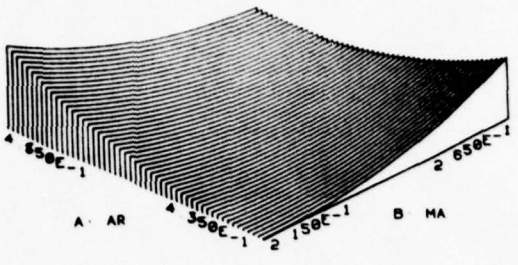
Frame	Parameter	NR Estimate	USSQ Estimate
1	a	.384	.381
	b	.104	.100
	σ_v^2	2.321	2.321
2	a	.253	.252
	b	-.130	-.132
	σ_v^2	2.791	2.791
3	a	.141	.140
	b	-.095	-.096
	σ_v^2	2.615	2.615
4	a	.239	.254
	b	-.040	-.025
	σ_v^2	2.534	2.533
5	a	.367	.381
	b	-.034	-.019
	σ_v^2	2.210	2.208

Table 4-14

NR and USSQ Analysis of 0 dB ARMA Test
with RDAN Excitation Sequence

Frame	Parameter	NR Estimate	USSQ Estimate
1	a	-.074	-.075
	b	-.233	-.234
	σ_v^2	1.979	1.979
2	a	.555	.555
	b	.383	.383
	σ_v^2	2.301	2.301
3	a	.143	.921
	b	.111	.936
	σ_v^2	2.763	2.762
4	a	-.096	-.961
	b	-.212	-1.000
	σ_v^2	2.634	2.622
5	a	.452	.453
	b	.236	.237
	σ_v^2	2.216	2.215

a)



b)

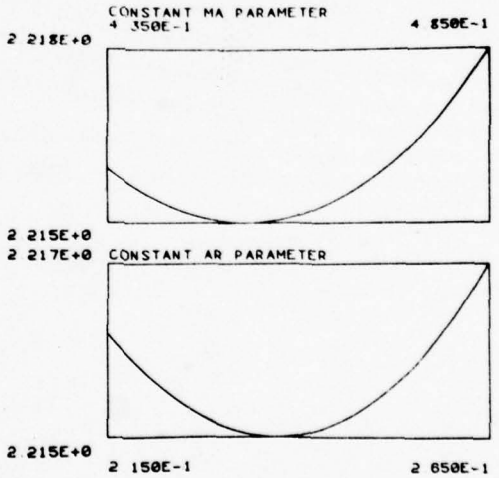


Figure 4-18: USSQ parameter estimation method applied to an ARMA(1,1) process
a) Quadratic surface in neighborhood of solution
b) Cross-sections through minimum

Table 4-15

NR and GN Analysis of 0 dB ARMA Test

Method	Parameter	Sample Mean	Sample Variance
NR	a(1)	.46881	6.2832×10^{-2}
GN	a(1)	.47317	6.9737×10^{-2}
NR	b(1)	.23245	7.3184×10^{-2}
GN	b(1)	.23619	7.8825×10^{-2}
NR	σ_v^2	2.467	4.011×10^{-2}
GN	σ_v^2	2.475	4.328×10^{-2}

spread in the estimates. But the differences do not appear to be striking, especially when the test is a 0 dB SNR simulation.

Recalling from Chapter 3 that the GN method requires the generation of two additional sequences in each iteration, that method requires a higher computational load per iteration. This is not compensated for by a decrease in the number of iterations per frame required to achieve convergence. Observations on a small number of frames indicate that neither method has an advantage in rate of convergence. Because of the computational savings of the NR method, it is used in all upcoming tests. This is especially important in the higher order models to be discussed. In the fourth order model, for example, the time required to process 518 frames of data is about three hours on a general purpose computer (DEC PDP-10).

The last two sets of experiments demonstrate the behavior of four estimators in the analysis of ARMA and AR+N tests based on the AR(1) process. Each of the estimators (NR, LPC, SYW, and W-LPC) is applied to sequences representing SNR's of ∞, 30, 20, 10, 0, and -10 dB. For each test 518 frames of data are available, with 256 points per frame. No mean correction is performed on the data. The Wiener filter used in the W-LPC method has a 21 point impulse response. It is generated as described in Chapter 3. The excitation sequence v(k) in

the ARMA tests is taken from NF3. NF3 is scaled for the proper sample variance $\hat{\sigma}_v^2$, the value of which is determined from Table 4-12. In the AR+N tests the AR(1) process (a(1) = 0.5) is synthesized using NF1 as the excitation $\varepsilon(k)$. NF1 is first scaled for a sample variance of $\hat{\sigma}_\varepsilon^2 = 1.0$. The additive noise required for the AR+N tests is generated by scaling NF2 appropriately. The results for the ARMA tests are presented first, followed by those for the AR+N tests.

Data obtained from the ARMA tests at the various simulated SNR's is listed in Table 4-16, 4-17, and 4-18. The estimates for a(1) are found in Table 4-16. Those for b(1) (NR method, only) are given in Table 4-17. The estimates for σ_v^2 from the NR and LPC methods are listed in Table 4-18. In these tables and those describing the AR+N tests, the data listed under "Mean" are the sample means of the parameter estimates. The sample variances are tabulated under "Variance". The number of frames analyzed is listed under "Frames". The synthesized data provides for a maximum of 518 frames. With the NR method it is possible for the Gauss elimination routine to detect a singular coefficient matrix. If that occurs, an error flag is set and the results for that frame are not included in the sample statistics for the estimates. The number of frames successfully analyzed by the NR method provides information about the stability of the estimation procedure

Table 4-16

Estimates of a(1) = 0.5 for ARMA Tests of the AR(1) Process

SNR	Method	Mean	Variance	Frames
∞	NR	.48312	1.35×10^{-2}	518
	SYW	.48810	3.28×10^{-3}	518
	LPC	.48958	5.48×10^{-3}	518
	W-LPC	.48958	5.48×10^{-3}	518
30	NR	.49519	1.36×10^{-2}	518
	SYW	.49250	1.36×10^{-2}	518
	LPC	.49474	5.20×10^{-3}	518
	W-LPC	.49597	5.23×10^{-3}	518
20	NR	.49505	1.39×10^{-2}	518
	SYW	.49252	1.38×10^{-2}	518
	LPC	.49034	5.26×10^{-3}	518
	W-LPC	.50032	5.17×10^{-3}	518
10	NR	.49355	1.67×10^{-2}	518
	SYW	.49279	1.67×10^{-2}	518
	LPC	.45039	5.75×10^{-3}	518
	W-LPC	.53679	4.68×10^{-3}	518
0	NR	.46881	6.28×10^{-2}	513
	SYW	.50986	8.74×10^{-2}	518
	LPC	.24955	7.29×10^{-3}	518
	W-LPC	.67112	2.96×10^{-3}	518
-10	NR	.24537	2.85×10^{-1}	221
	SYW	-3.3153	3.54×10^{3}	518
	LPC	.04897	7.05×10^{-3}	518
	W-LPC	.77210	1.80×10^{-3}	518

Table 4-17

NR Estimates of b(1) for ARMA Tests
of the AR(1) Process

SNR	True Value	Mean	Variance	Frames
∞	0	$-.88433 \times 10^{-2}$	1.68×10^{-2}	518
30	$.66556 \times 10^{-3}$	$-.59483 \times 10^{-2}$	1.96×10^{-2}	518
20	$.65577 \times 10^{-2}$	$-.21100 \times 10^{-3}$	2.00×10^{-2}	518
10	$.57331 \times 10^{-1}$	$.48995 \times 10^{-1}$	2.37×10^{-2}	518
0	.26795	.23245	7.32×10^{-2}	513
-10	.45573	.16904	2.95×10^{-1}	221

Table 4-18

NR and LPC Estimates of σ_v^2 for ARMA Tests of the AR(1) Process

SNR	True Value	Method	Mean	Variance	Frames
∞	1.0	NR	.9948	7.62×10^{-3}	518
		LPC	.9984	1.41×10^{-2}	518
30	1.0017	NR	.9942	6.58×10^{-3}	518
		LPC	1.001	1.29×10^{-2}	518
20	1.0166	NR	1.009	6.77×10^{-3}	518
		LPC	1.015	1.33×10^{-2}	518
10	1.1628	NR	1.154	8.86×10^{-3}	518
		LPC	1.162	1.74×10^{-2}	518
0	2.4880	NR	2.467	4.01×10^{-2}	513
		LPC	2.493	8.11×10^{-2}	518
-10	14.628	NR	14.48	1.48	221
		LPC	14.61	2.78	518

at varying noise levels.

These trends are noted from Tables 4-16, 4-17, and 4-18:

1) The estimates for a(1) are good for all methods at SNR's at and above 20 dB.
2) The LPC estimate for a(1) is noticeably degraded for SNR's below 20 dB.
3) Decreasing the SNR causes an increase in the sample variance of all estimates.
4) The increasing sample variance is strongest in the NR and SYW methods, which have sample variances greater than that for the LPC estimate in all cases.
5) While the LPC estimate for a(1) tends toward zero as the noise level worsens, the W-LPC estimate is increasing.
6) In terms of the sample mean, the NR and SYW estimates are superior to the other methods at SNR's of 0 and 10 dB. All methods perform badly at -10 dB.
7) At 0 dB the NR method fails in 5 frames. The method is successful in only 221 frames at -10 dB.
8) At all SNR's, the average NR estimate for σ_v^2 lies below the average LPC estimate for that parameter, indicating that the NR method is doing a better job of finding the minimum of the quadratic surface.

Figure 4-19 is a plot of the sample mean data for the a(1) estimates from Table 4-16. This clearly indicates the

degradation in the LPC and W-LPC estimates below 20 dB. At 0 dB the NR estimate has the same error as the LPC estimate at approximately 14 dB. The SYW estimate provides an even greater improvement with much lower computational requirements. However, it degrades radically between 0 and -10 dB.

The curves plotted in Figure 4-19 show a definite advantage in the NR and SYW estimates at 0 and 10 dB SNR's when compared to the LPC estimate. There is another aspect to these estimates that must be emphasized, however. Figure 4-20 shows a plot of the NR and LPC estimates from Figure 4-19. Also shown in the plot are vertical lines indicating one sample standard deviation interval away from the sample mean. The standard deviation is obtained by taking the square root of the sample variance of the estimator listed in Table 4-16. For SNR's above 10 dB the spread of the two estimates is comparable. At 10 dB the bias in the LPC estimate is evident, though the spread of the estimate stays about the same. At SNR's below 20 dB, the sample deviation for the NR estimate grows quickly. Even though the interval covered by the $\pm\sigma$ limits for the NR estimate always includes the desired value of 0.5, the large spread in the estimate at the poorer SNR's indicates that a single NR estimate can have a large error when compared to the true parameter value. Only in the average does the estimate approximate the true value well. The

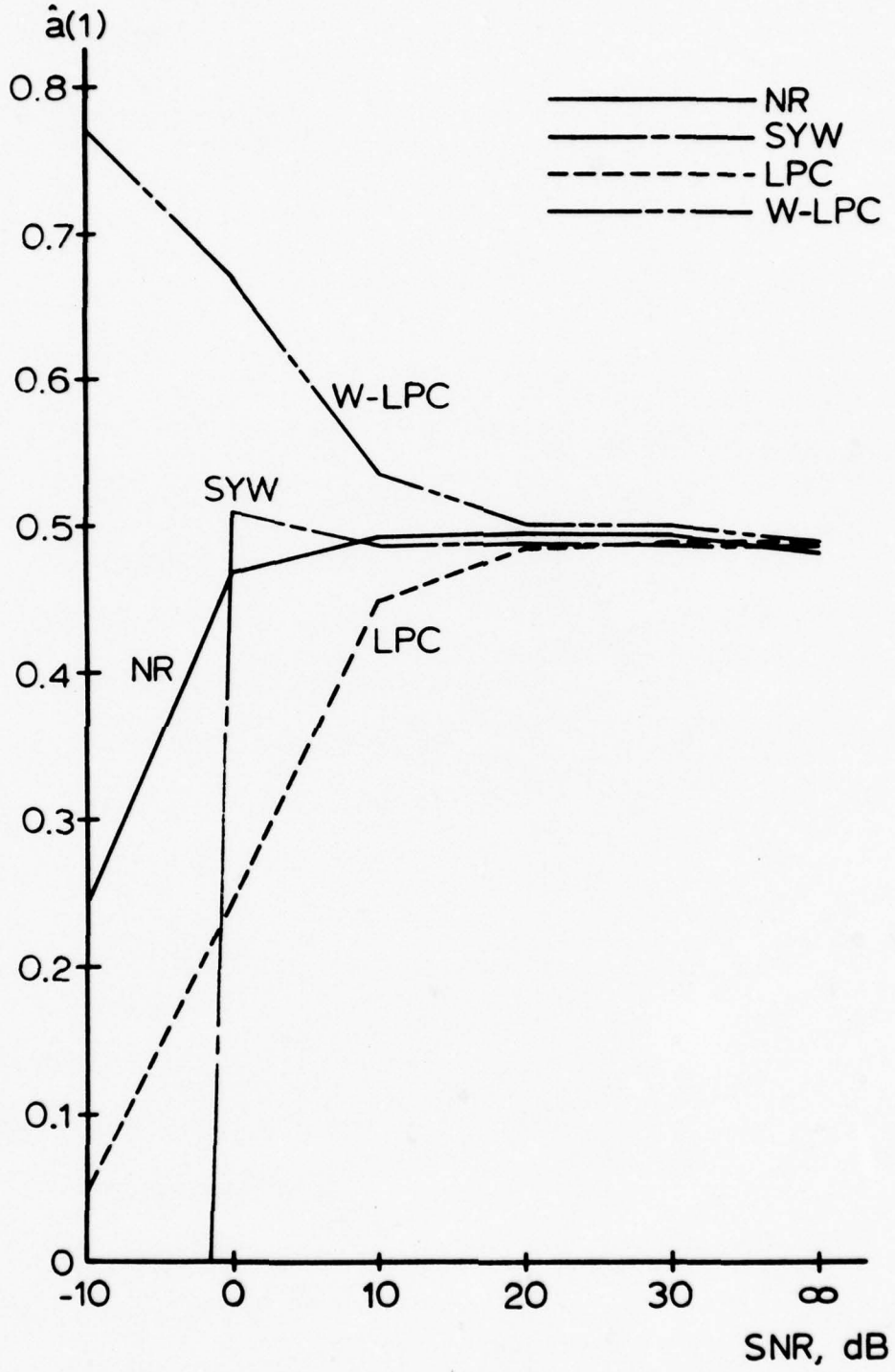


Figure 4-19: Comparison of four estimators of a(1) in ARMA tests

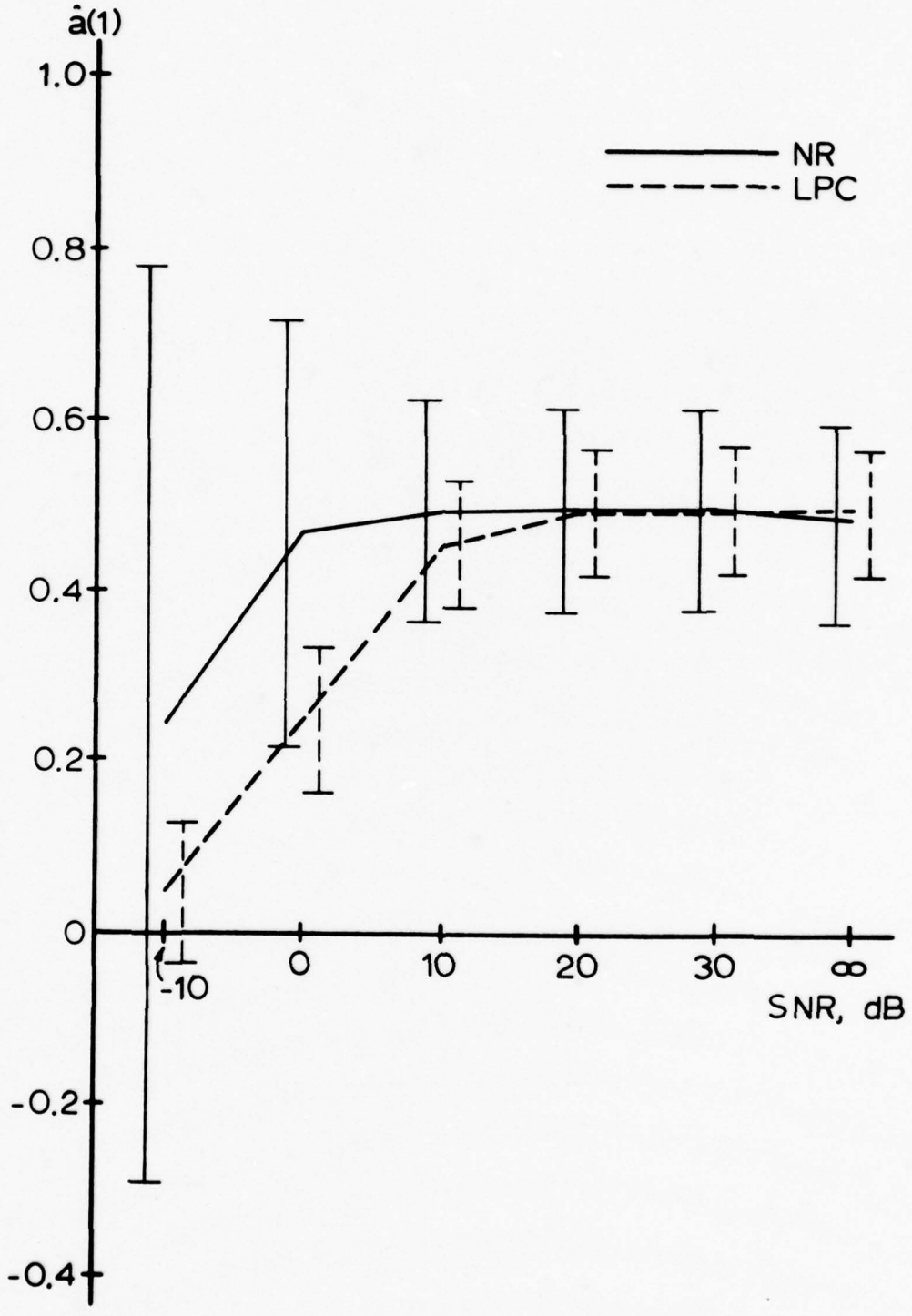


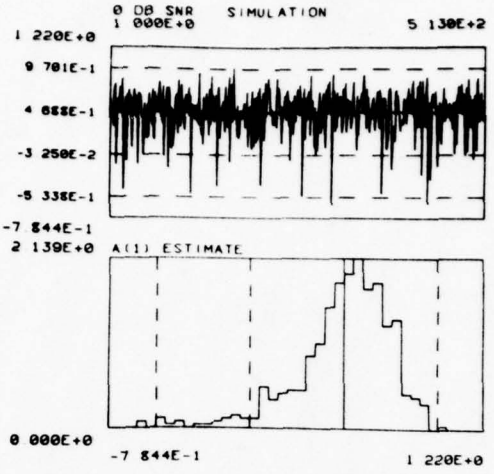
Figure 4-20: NR and LPC estimates of a(1) in ARMA tests, shown with $\pm$ σ limits

results for the SYW estimate are quite similar.

The upper plots in Figure 4-21 a)-d) show the time series generated by appending the estimates from each type of estimator for the 0 dB case. Parts a)-d) of this figure correspond to the NR, SYW, LPC, and W-LPC estimators, respectively. The lower plot in each part of the figure is an amplitude histogram obtained from the data in the upper plot. The histogram is divided into 40 cells. A solid line in each plot indicates the location of the sample mean. The dashed lines in the plots mark intervals around the sample mean of 2σ. Of the four estimates, the NR estimate has a somewhat asymmetrical distribution about the sample mean. The other estimators are more symmetrically distributed. The asymmetry of the NR estimate is in a direction that tends to favor the estimate. That is, the bulk of the distribution is shifted toward the true value of the parameter. This alleviates the larger spread of this estimator somewhat, though that is still a serious problem.

The experiments performed on the ARMA simulations just described are repeated in AR+N tests. The approach is to generate the AR(1) process, add noise to achieve the desired SNR, and apply the four estimation algorithms. The details for these experiments are described with those for the ARMA test. Table 4-19 lists the data for estimates of a(1). The NR estimates for b(1) are given in Table 4-20.

a)



b)

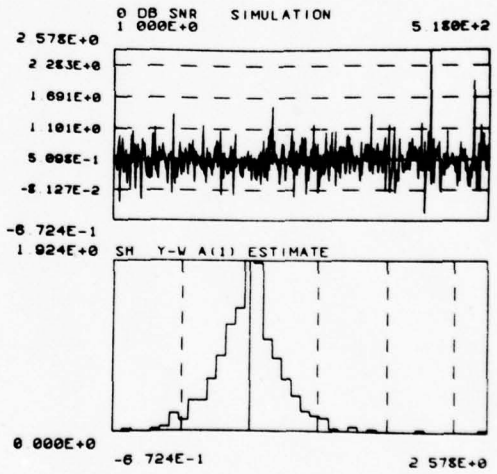
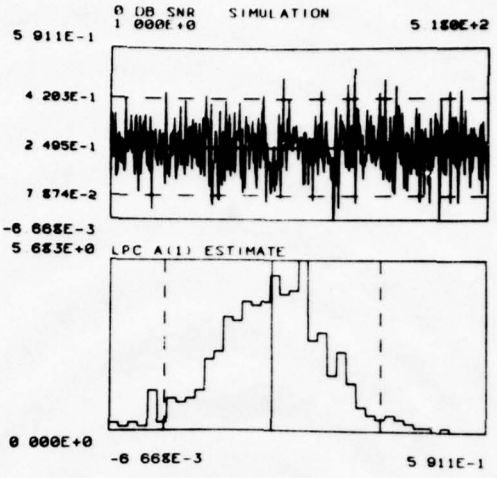


Figure 4-21: Time series and histograms of estimates of a(1) in ARMA tests
a) NR estimate
b) SYW estimate

c)



d)

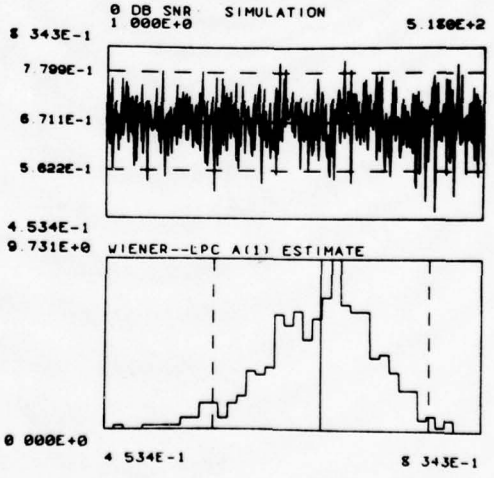


Figure 4-21: c) LPC estimate
d) W-LPC estimate

The NR and LPC estimates for σ_v^2 are listed in Table 4-21. The comments made for the corresponding tables in the ARMA tests hold for these results. The most important aspects of this data are summarized as follows:

1) the variance of the estimates increases with higher levels of noise;
2) based on average statistics, the NR and SYW are superior to LPC at SNR'S below 20 dB;
3) the variance of the NR and SYW estimates is larger than that for the LPC estimates.

Figure 4-22 presents the sample means of the four a(1) estimates versus SNR. The NR estimate at 0 dB is about the same as the LPC error at 14 dB. Interpolating the SYW estimate at -1 dB, the same error occurs as with LPC at 14 dB, an extension of 15 dB. However, the SYW estimate again degrades more rapidly below 0 dB than the NR method, though not as badly as in the ARMA tests. The NR and LPC estimates are shown again in Figure 4-23 with the $\pm\sigma$ limits indicated. The comments made concerning the ARMA test results in Figure 4-20 also apply to Figure 4-23. The advantage of the NR estimate applies only in the average. The large sample standard deviation weighs against the use of individual estimates.

The a(1) time series and histograms for the AR+N tests at 0 dB are shown in Figure 4-24 a)-d). The dashed lines mark deviations in the estimate from the sample mean by

Table 4-19

Estimates of a(1) = 0.5 for AR+N Tests of the AR(1) Process

SNR	Method	Mean	Variance	Frames
∞	NR	.48312	1.35×10^{-2}	518
	SYW	.48810	3.28×10^{-3}	518
	LPC	.48958	5.48×10^{-3}	518
	W-LPC	.48958	5.48×10^{-3}	518
30	NR	.48334	1.35×10^{-2}	518
	SYW	.48233	1.39×10^{-2}	518
	LPC	.48907	5.51×10^{-3}	518
	W-LPC	.49062	5.47×10^{-3}	518
20	NR	.48351	1.37×10^{-2}	518
	SYW	.48250	1.40×10^{-2}	518
	LPC	.48464	5.63×10^{-3}	518
	W-LPC	.49490	5.46×10^{-3}	518
10	NR	.48310	1.66×10^{-2}	518
	SYW	.48301	1.61×10^{-2}	518
	LPC	.44477	6.32×10^{-3}	518
	W-LPC	.53142	5.09×10^{-3}	518
0	NR	.46320	5.36×10^{-2}	515
	SYW	.49954	9.64×10^{-2}	518
	LPC	.24404	8.34×10^{-3}	518
	W-LPC	.66690	3.23×10^{-3}	518
-10	NR	.29938	2.56×10^{-1}	214
	SYW	.16791	5.98×10^{1}	518
	LPC	.04276	7.74×10^{-3}	518
	W-LPC	.76962	1.85×10^{-3}	518

Table 4-20

NR Estimates of b(1) for AR+N Tests of the AR(1) Process

SNR	True Value	Mean	Variance	Frames
∞	0	$-.88433 \times 10^{-2}$	1.68×10^{-2}	518
30	$.66556 \times 10^{-3}$	$-.79156 \times 10^{-2}$	1.68×10^{-2}	518
20	$.65577 \times 10^{-2}$	$-.19046 \times 10^{-2}$	1.70×10^{-2}	518
10	$.57331 \times 10^{-1}$	$.47712 \times 10^{-1}$	2.10×10^{-2}	518
0	.26795	.23168	6.22×10^{-2}	515
-10	.45573	.22541	2.65×10^{-1}	214

Table 4-21

NR and LPC Estimates of σ_v^2 for AR+N
Tests of the AR(1) Process

SNR	True Value	Method	Mean	Variance	Frames
∞	1.0	NR	.9948	7.62×10^{-3}	518
		LPC	.9984	1.41×10^{-2}	518
30	1.0017	NR	.9966	7.64×10^{-3}	518
		LPC	1.000	1.41×10^{-2}	518
20	1.0166	NR	1.012	7.89×10^{-3}	518
		LPC	1.015	1.45×10^{-2}	518
10	1.1628	NR	1.156	1.05×10^{-2}	518
		LPC	1.160	1.92×10^{-2}	518
0	2.4880	NR	2.464	5.15×10^{-2}	515
		LPC	2.473	9.14×10^{-2}	518
-10	14.628	NR	14.35	2.11	214
		LPC	14.41	2.99	518

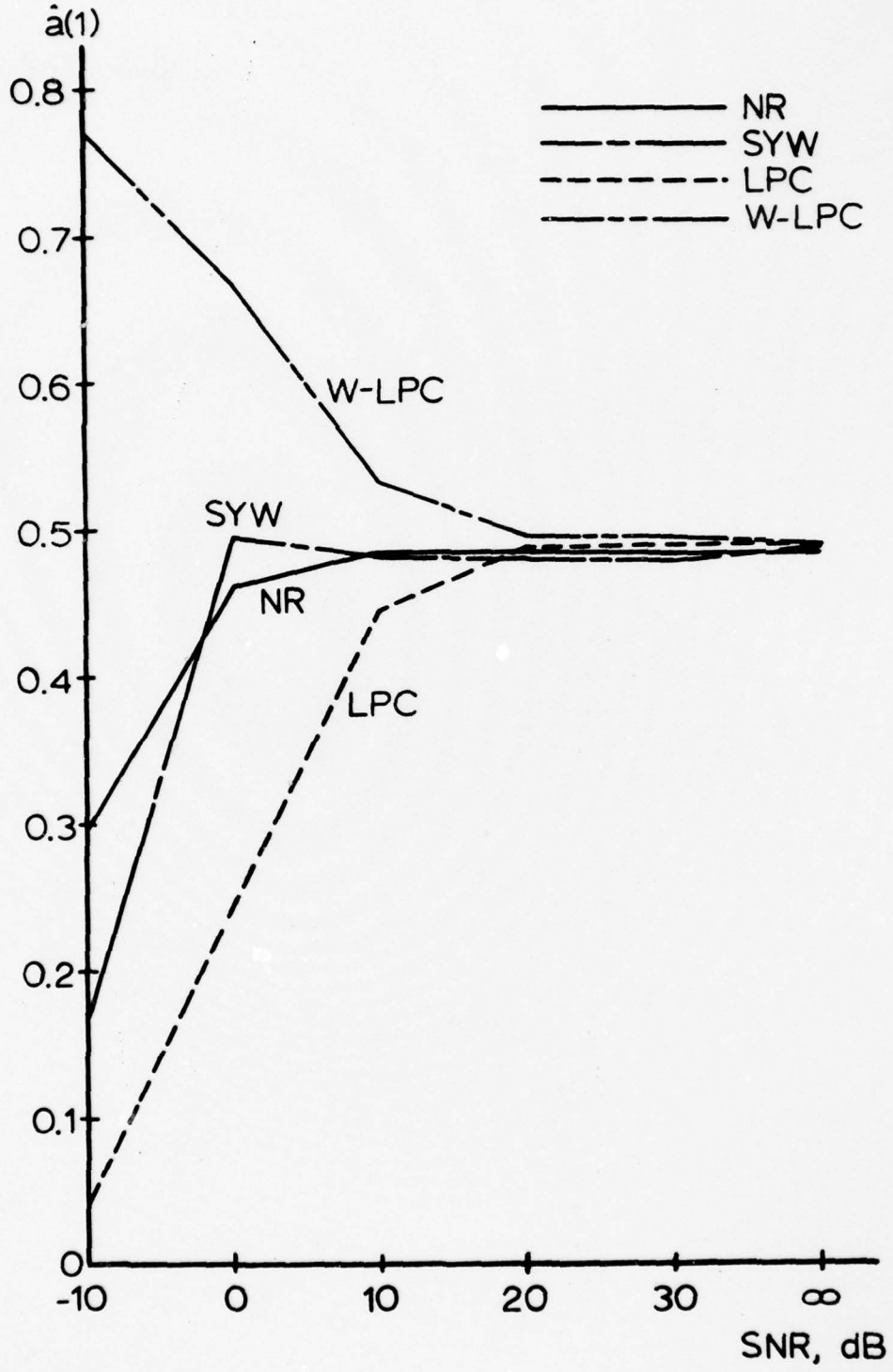


Figure 4-22: Comparison of four estimators of a(1) in AR+N tests

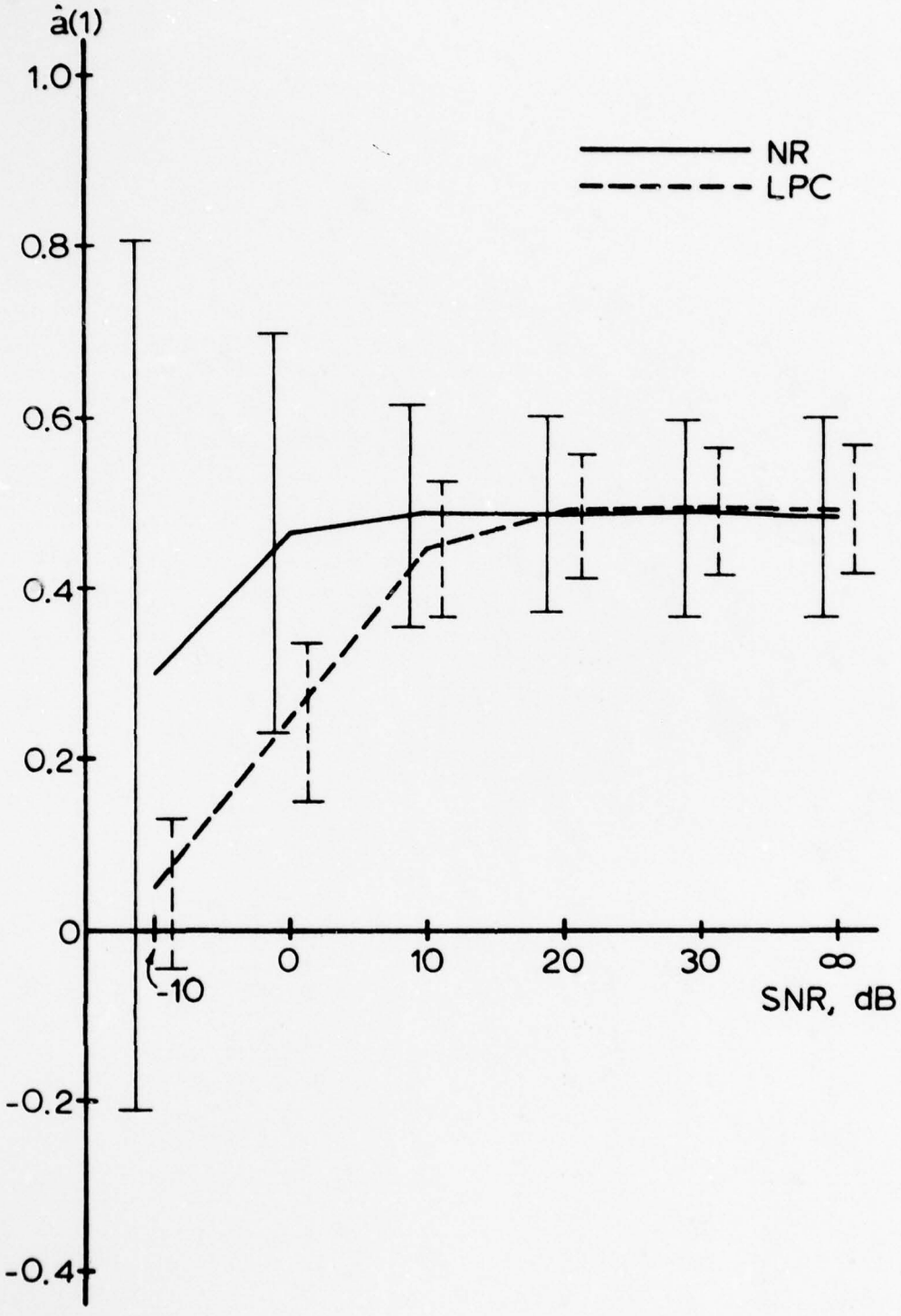
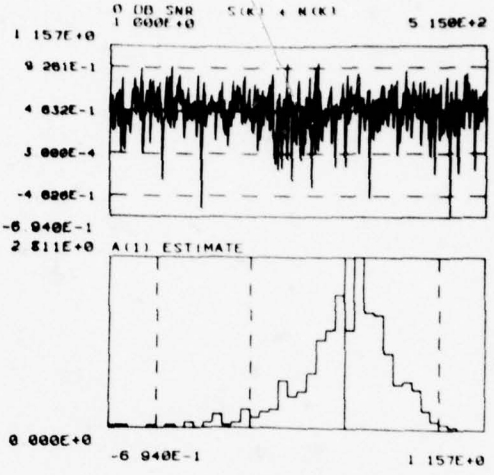


Figure 4-23: NR and LPC estimates of a(1) in AR+N tests, shown with $\pm$ σ limits

a)



b)

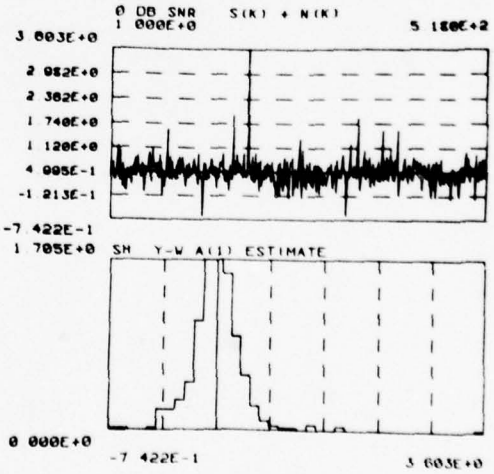
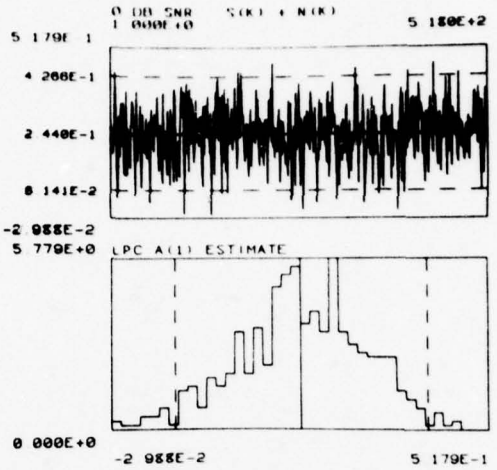


Figure 4-24: Time series and histograms of estimates of a(1) in AR+N tests
a) NR estimate
b) SYW estimate

c)



d)

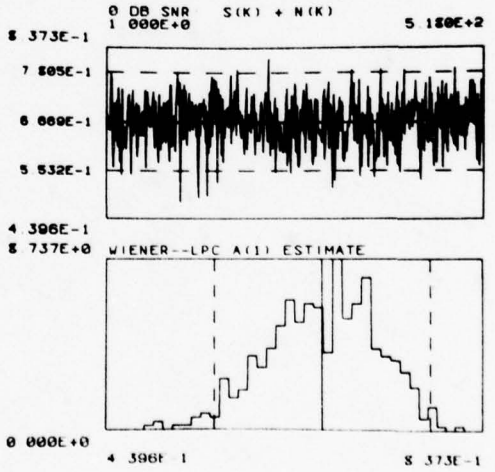


Figure 4-24: c) LPC estimate
d) W-LPC estimate

multiples of 2σ. The sample mean is indicated by the solid line. These plots are quite similar to those for the ARMA tests. The asymmetry of the NR estimate is apparent in the AR+N tests, as well.

This concludes the presentation of the data on various first order models. The AR(1) and MA(1) processes are analyzed to determine the behavior of the NR estimator in noiseless situations. Several estimators are then applied to the AR(1) plus white noise model. These tests are performed as ARMA simulations of the model and AR+N actual tests of the model. The results for the two approaches agree, establishing confidence in the algorithms and model.

The Second Order Model

In the preceding section, the results from the analysis of an AR(1) process at several SNR's are presented. In this section a single AR(2) process is considered. The AR model selected for this experiment has a complex-conjugate pole-pair located at a radius of 0.9 and a center frequency of $\pm$1000 Hz, referenced to a sampling frequency of 6667 Hz. This model results in two AR coefficients: a(1) = -1.05808 and a(2) = 0.81000. Figure 4-25 presents the inverse spectrum of this AR(2) operator. This spectrum shows on a dB scale the single resonant peak resulting from the conjugate pole-pair. The AR(2) process s(k) is obtained by exciting the AR(2) model with NF1, which is first scaled for a sample variance of

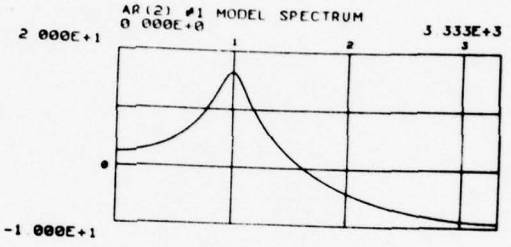


Figure 4-25: Inverse Spectrum of AR(2) model

1.0. NF2 is appropriately scaled and added as n(k) to produce x(k), the data to be analyzed. The theoretical variance of this AR(2) model is 4.41735. The sample variance of the AR(2) process generated by using the scaled NF1 noise as the excitation sequence is 4.43889. With 256 points per frame, 518 frames of data are available for analysis. The data in each frame is corrected for a zero sample mean prior to analysis by any method. The initial estimates for the parameters of the ARMA(2,2) model in the NR method are:

1) for the two AR parameters, the actual model coefficients are used;
2) for the two MA parameters, zeros are used as the initial guesses.

These tests are similar to the AR+N tests of the preceding section, but in this case the MA parameters in the equivalent ARMA(2,2) model are unknown. Hence, the initial guesses for the MA coefficients are zero.

Using the NR, LPC, and SYW algorithms, the estimation data for the a(1) and a(2) parameters is listed in Tables 4-22 and 4-23, respectively. Two distance measures which combine the error for each coefficient estimate into one parameter are now defined:

$$L[\hat{a}(k)] = \frac{1}{q} \sum_{i=1}^{q} [a(i) - \hat{a}(i)]^2 , \qquad (4.1a)$$

Table 4-22

Estimates for a(1) = -1.05808 of the AR(2) Process

SNR	Method	Sample Mean	Sample Variance	Frames
∞	NR	-1.0547	2.76×10^{-3}	518
	SYW	-1.0441	1.60×10^{-3}	518
	LPC	-1.0491	2.58×10^{-3}	518
30	NR	-1.0547	2.75×10^{-3}	518
	SYW	-1.0500	2.79×10^{-3}	518
	LPC	-1.0468	2.61×10^{-3}	518
20	NR	-1.0547	2.77×10^{-3}	518
	SYW	-1.0499	2.79×10^{-3}	518
	LPC	-1.0258	2.92×10^{-3}	518
10	NR	-1.0545	3.16×10^{-3}	518
	SYW	-1.0500	3.17×10^{-3}	518
	LPC	-.85837	6.27×10^{-3}	518
0	NR	-1.0505	8.78×10^{-3}	499
	SYW	-1.0505	2.62×10^{-2}	518
	LPC	-.34856	9.76×10^{-3}	518
-10	NR	-.69685	4.23×10^{-1}	183
	SYW	-4.0376	2.66×10^{3}	518
	LPC	-.05335	7.00×10^{-3}	518

Table 4-23

Estimates for a(2) = 0.81000 of the AR(2) Process

SNR	Method	Sample Mean	Sample Variance	Frames
∞	NR	.80474	1.66×10^{-3}	518
	SYW	.79584	1.55×10^{-3}	518
	LPC	.80199	2.32×10^{-3}	518
30	NR	.80471	1.66×10^{-3}	518
	SYW	.79822	1.74×10^{-3}	518
	LPC	.79988	2.36×10^{-3}	518
20	NR	.80478	1.70×10^{-3}	518
	SYW	.79820	1.79×10^{-3}	518
	LPC	.78061	2.82×10^{-3}	518
10	NR	.80538	2.10×10^{-3}	518
	SYW	.79894	2.73×10^{-3}	518
	LPC	.62791	6.67×10^{-3}	518
0	NR	.78657	1.81×10^{-2}	499
	SYW	.81823	6.25×10^{-2}	518
	LPC	.20474	8.00×10^{-3}	518
-10	NR	.45245	2.56×10^{-1}	183
	SYW	.42323	1.08×10^{2}	518
	LPC	.03104	6.13×10^{-3}	518

$$\%L[\hat{a}(k)] = \frac{1}{q} \sum_{i=1}^{q} [1 - \frac{\hat{a}(i)}{a(i)}]^2 \quad . \tag{4.1b}$$

In (4.1) the $\{\hat{a}(i)\}_1^q$ are estimates of the parameters $\{a(i)\}_1^q$. $L[\cdot]$ and $\%L[\cdot]$ are shown in (4.1) in terms of the AR parameters. With appropriate substitutions, these expressions can also be used to calculate distance measures between the MA coefficients, when known, and their estimates. The measures are computed at each frame of data. The expression in (4.1a) is the sum of squares of the error between the parameters and their estimates. This type of error criterion tends to give more weight to errors in parameters with a larger magnitude. The measure in (4.1b) is designed to counteract that tendency. The error criterion of (4.1b) computes the difference between the parameters and their estimates relative to the true value of the parameter.

The sample statistics obtained by averaging the distance measures computed at each frame for the three sets of AR estimates are listed in Tables 4-24 and 4-25. $L[\hat{a}(k)]$ is given in the former, $\%L[\hat{a}(k)]$ in the latter. As seen from the distance measure data, the NR and SYW methods perform better than the LPC method at all SNR's. The variance of these estimates is always smaller than that of the LPC estimates except at a SNR of -10 dB. The SYW method performs better than the NR method at a SNR of infinity. However, the NR method is forced to estimate two MA parameters in that case. Estimating these theoretically

Table 4-24

$L[\hat{a}(k)]$ Distance Measure for the AR(2) Process

SNR	Method	Sample Mean	Sample Variance
∞	NR	2.23×10^{-3}	1.01×10^{-5}
	SYW	1.77×10^{-3}	6.26×10^{-6}
	LPC	2.52×10^{-3}	1.05×10^{-5}
30	NR	2.23×10^{-3}	1.01×10^{-5}
	SYW	2.36×10^{-3}	1.34×10^{-5}
	LPC	2.60×10^{-3}	1.17×10^{-5}
20	NR	2.26×10^{-3}	1.04×10^{-5}
	SYW	2.39×10^{-3}	1.38×10^{-5}
	LPC	3.82×10^{-3}	3.02×10^{-5}
10	NR	2.65×10^{-3}	1.77×10^{-5}
	SYW	3.04×10^{-3}	2.07×10^{-5}
	LPC	4.30×10^{-2}	1.13×10^{-3}
0	NR	1.37×10^{-2}	2.28×10^{-3}
	SYW	4.44×10^{-2}	1.29×10^{-2}
	LPC	4.44×10^{-1}	1.36×10^{-2}
-10	NR	4.69×10^{-1}	5.08×10^{-1}
	SYW	1.39×10^{3}	4.82×10^{8}
	LPC	8.15×10^{-1}	1.20×10^{-2}

Table 4-25

%L[$\hat{a}(k)$] Distance Measure for the AR(2) Process

SNR	Method	Sample Mean	Sample Variance
∞	NR	2.53×10^{-3}	1.30×10^{-5}
	SYW	2.14×10^{-3}	9.18×10^{-6}
	LPC	3.00×10^{-3}	1.48×10^{-5}
30	NR	2.52×10^{-3}	1.30×10^{-5}
	SYW	2.70×10^{-3}	1.75×10^{-5}
	LPC	3.10×10^{-3}	1.67×10^{-5}
20	NR	2.56×10^{-3}	1.35×10^{-5}
	SYW	2.74×10^{-3}	1.83×10^{-5}
	LPC	4.58×10^{-3}	4.41×10^{-5}
10	NR	3.04×10^{-3}	2.25×10^{-5}
	SYW	3.62×10^{-3}	3.08×10^{-5}
	LPC	5.10×10^{-2}	1.64×10^{-3}
0	NR	1.82×10^{-2}	4.64×10^{-3}
	SYW	5.94×10^{-2}	2.78×10^{-2}
	LPC	5.14×10^{-1}	1.87×10^{-2}
-10	NR	5.40×10^{-1}	5.88×10^{-1}
	SYW	1.27×10^{3}	3.87×10^{8}
	LPC	9.21×10^{-1}	1.59×10^{-2}

zero parameters introduces more uncertainty into the AR estimates. The SYW again fails badly at -10 dB SNR.

Figure 4-26a) plots the sample mean of $L[\hat{a}(k)]$ versus the SNR. $\%L[\hat{a}(k)]$ is plotted in Figure 4-26b). From these plots it is evident that the NR and SYW methods do extend the range over which the AR parameters can be estimated in the presence of white noise. At 0 dB, the NR estimate has the same error as the LPC estimate at 14 dB. For the SYW estimate, the improvement is 10 dB, referenced to the error at 0 dB for the SYW estimate. From Table 4-22, it is again noted that the NR method fails to successfully estimate parameters in some frames at SNR's of 0 and -10 dB.

The Fourth Order Model

The tests of the preceding section on the AR(2) model are now performed on an AR(4) model. The AR coefficients are: a(1) = -0.49336, a(2) = 0.45804, a(3) = -0.28481, and a(4) = 0.58523. The AR operator with these coefficients has two Z-plane singularities with a radius of 0.9 at $\pm$800 Hz. The other two singularities have a radius of 0.85 at $\pm$2200 Hz. The center frequencies are referenced to a sampling frequency of 6667 Hz. The inverse spectrum of this AR model is shown in Figure 4-27 in dB. Excitation for the process is NF1 scaled for a sample variance of 1.0. Each frame of data is corrected for a zero sample mean before analysis with the NR, SYW, and LPC methods. The initial estimates in the NR method for the AR parameters

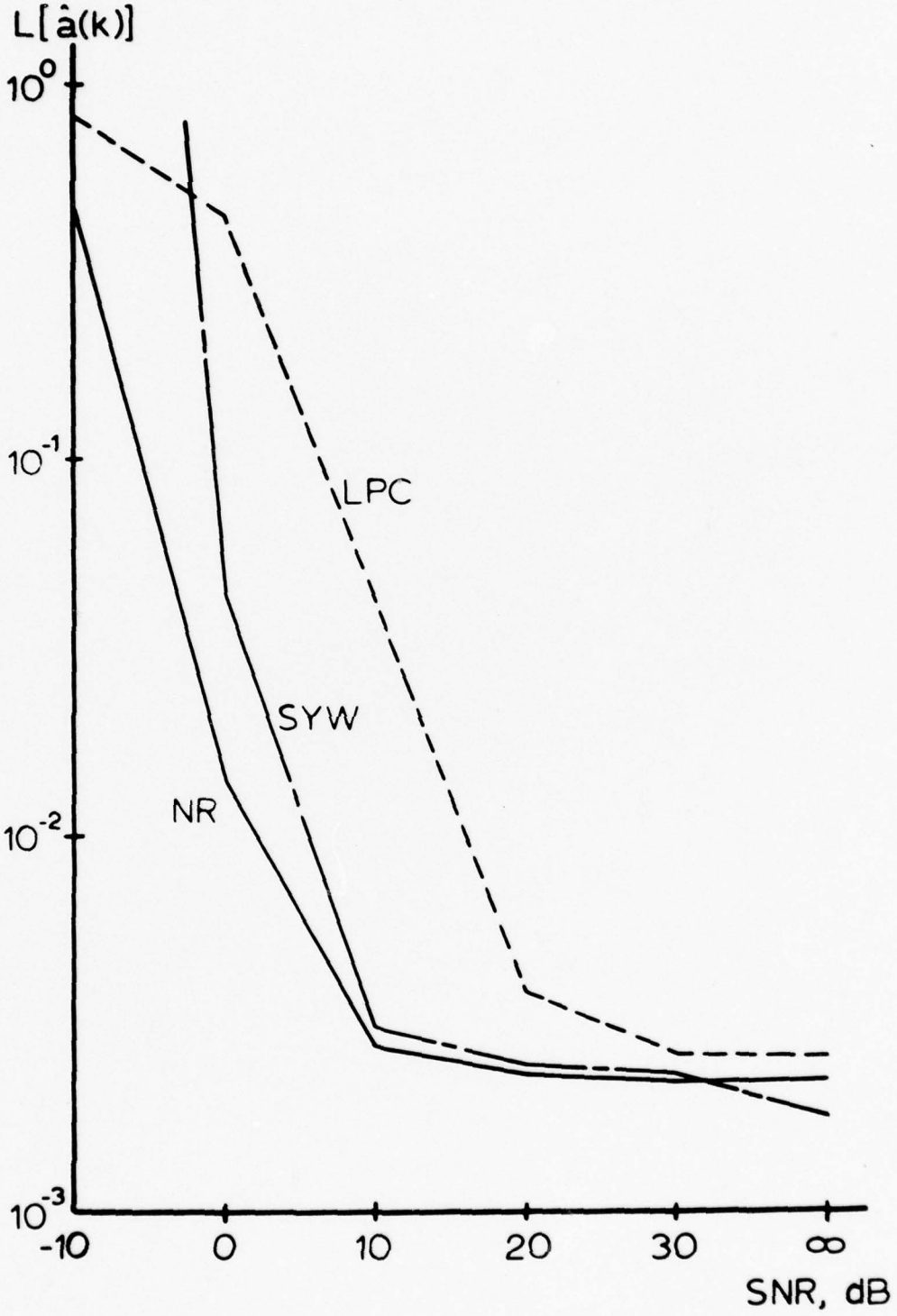


Figure 4-26: Distance measures for three estimates of the AR(2) model coefficients
a) $L[\hat{a}(k)]$

are the actual model coefficients. Those for the four MA coefficients of the equivalent ARMA(4,4) model are zero.

Tables 4-26, 4-27, 4-28, and 4-29 list the estimates for a(1), a(2), a(3), and a(4), respectively. The statistics for the two distance measure, obtained by averaging the distance measure for each frame analyzed, are given in Tables 4-30 and 4-31. Note from this data that the LPC estimates have the smallest error except at a SNR of infinity. At that SNR, the SYW estimates are slightly better. The improvement seen in the first and second order cases is not evident. It must be realized, however, that the two distance measures used combine the errors for the individual coefficients into one parameter. Smoothing of the coefficient errors occurs and is more significant for the AR(4) case.

Taking another approach, the two distance measures are computed for each type of estimator using the sample means for the AR coefficient estimates listed in Tables 4-26 through 4-29. As noted in the results for the first order model, the value of the AR estimates based on the transformation model and the NR estimation procedure lies mainly in the average of a large number of estimates. This is because of the larger variance of the NR estimates as compared to the LPC estimates. The distance measures defined in (4.1) are applied to the average values of the parameter estimates to determine whether or not the

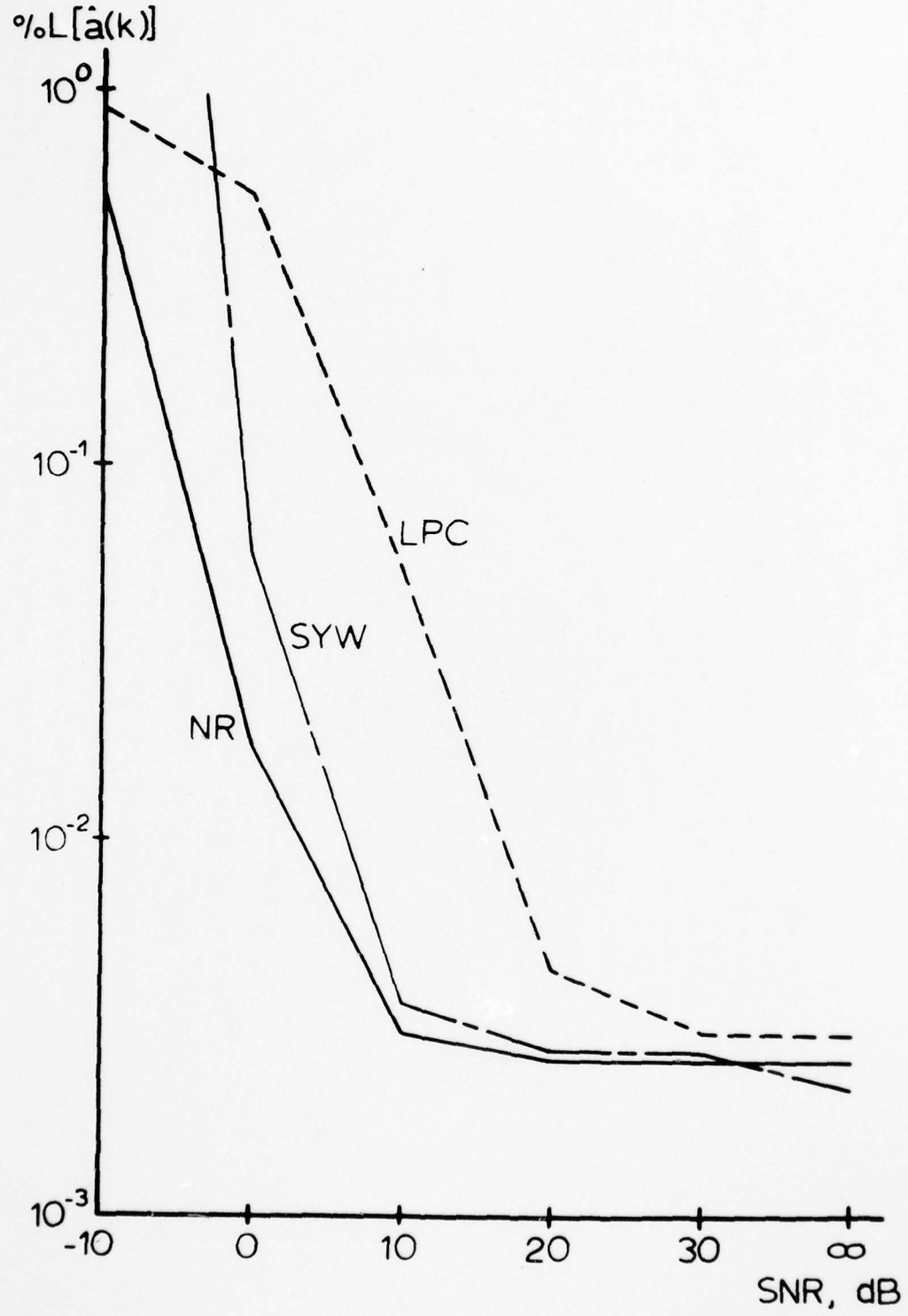


Figure 4-26: b) %L[â(k)]

Table 4-26

Estimates for a(1) = -0.49336 of the AR(4) Process

SNR	Method	Sample Mean	Sample Variance	Frames
∞	NR	-.48417	1.40×10^{-2}	518
	SYW	-.48727	2.66×10^{-3}	518
	LPC	-.48870	4.58×10^{-3}	518
30	NR	-.48265	1.40×10^{-2}	518
	SYW	-.48200	1.15×10^{-2}	518
	LPC	-.48793	4.62×10^{-3}	518
20	NR	-.48266	1.55×10^{-2}	518
	SYW	-.48028	1.23×10^{-2}	518
	LPC	-.47935	4.73×10^{-3}	518
10	NR	-.47231	2.37×10^{-2}	517
	SYW	-.47559	2.02×10^{-2}	518
	LPC	-.40877	5.34×10^{-3}	518
0	NR	-.48742	2.15×10^{-1}	469
	SYW	-.41230	5.96×10^{-1}	518
	LPC	-.18913	6.55×10^{-3}	518

Table 4-27

Estimates for a(2) = 0.45804 of the AR(4) Process

SNR	Method	Sample Mean	Sample Variance	Frames
∞	NR	.44983	1.82×10^{-2}	518
	SYW	.45059	3.43×10^{-3}	518
	LPC	.45964	5.71×10^{-3}	518
30	NR	.44888	1.84×10^{-2}	518
	SYW	.44778	1.54×10^{-2}	518
	LPC	.45865	5.70×10^{-3}	518
20	NR	.45029	1.95×10^{-2}	518
	SYW	.44724	1.60×10^{-2}	518
	LPC	.44662	5.82×10^{-3}	518
10	NR	.44611	3.09×10^{-2}	517
	SYW	.44789	2.21×10^{-2}	518
	LPC	.34974	6.87×10^{-3}	518
0	NR	.48444	3.07×10^{-1}	469
	SYW	.52888	1.25	518
	LPC	.10894	7.05×10^{-3}	518

Table 4-28

Estimates for a(3) = -0.28481 of the AR(4) Process

SNR	Method	Sample Mean	Sample Variance	Frames
∞	NR	-.27538	1.20×10^{-2}	518
	SYW	-.27375	3.37×10^{-3}	518
	LPC	-.27907	5.50×10^{-3}	518
30	NR	-.27500	1.21×10^{-2}	518
	SYW	-.27460	1.03×10^{-2}	518
	LPC	-.27798	5.55×10^{-3}	518
20	NR	-.27571	1.19×10^{-2}	518
	SYW	-.27480	1.03×10^{-2}	518
	LPC	-.26659	5.85×10^{-3}	518
10	NR	-.27877	1.85×10^{-2}	517
	SYW	-.27789	1.29×10^{-2}	518
	LPC	-.17874	7.60×10^{-3}	518
0	NR	-.29406	1.80×10^{-1}	469
	SYW	-.39988	1.33	518
	LPC	.44233	7.81×10^{-3}	518

Table 4-29

Estimates for a(4) = 0.58523 of the AR(4) Process

SNR	Method	Sample Mean	Sample Variance	Frames
∞	NR	.58194	6.76×10^{-3}	518
	SYW	.56907	2.69×10^{-3}	518
	LPC	.57145	4.32×10^{-3}	518
30	NR	.58288	6.69×10^{-3}	518
	SYW	.57339	6.10×10^{-3}	518
	LPC	.57051	4.36×10^{-3}	518
20	NR	.58389	7.45×10^{-3}	518
	SYW	.57476	6.45×10^{-3}	518
	LPC	.56129	4.56×10^{-3}	518
10	NR	.59180	1.20×10^{-2}	517
	SYW	.58034	1.17×10^{-2}	518
	LPC	.48643	5.68×10^{-3}	518
0	NR	.58380	1.25×10^{-1}	469
	SYW	.67750	5.34×10^{-1}	518
	LPC	.24356	7.25×10^{-3}	518

Table 4-30

$L[\hat{a}(k)]$ Distance Measure for the AR(4) Process

SNR	Method	Sample Mean	Sample Variance
∞	NR	1.28×10^{-2}	2.76×10^{-4}
	SYW	3.16×10^{-3}	9.98×10^{-6}
	LPC	5.09×10^{-3}	1.87×10^{-5}
30	NR	1.29×10^{-2}	2.83×10^{-4}
	SYW	1.09×10^{-2}	1.60×10^{-4}
	LPC	5.13×10^{-3}	1.94×10^{-5}
20	NR	1.37×10^{-2}	3.29×10^{-4}
	SYW	1.14×10^{-2}	1.78×10^{-4}
	LPC	5.55×10^{-3}	2.50×10^{-5}
10	NR	2.14×10^{-2}	1.61×10^{-3}
	SYW	1.69×10^{-2}	5.24×10^{-4}
	LPC	1.63×10^{-2}	1.85×10^{-4}
0	NR	2.07×10^{-1}	2.88×10^{-1}
	SYW	9.37×10^{-1}	9.47×10^{1}
	LPC	1.11×10^{-1}	1.26×10^{-3}

Table 4-31

%L[$\hat{a}(k)$] Distance Measure for the AR(4) Process

SNR	Method	Sample Mean	Sample Variance
∞	NR	7.83×10^{-2}	1.19×10^{-2}
	SYW	1.98×10^{-2}	5.61×10^{-4}
	LPC	3.19×10^{-2}	9.73×10^{-4}
30	NR	7.89×10^{-2}	1.23×10^{-2}
	SYW	6.70×10^{-2}	7.21×10^{-3}
	LPC	3.22×10^{-2}	1.01×10^{-3}
20	NR	8.19×10^{-2}	1.26×10^{-2}
	SYW	6.89×10^{-2}	7.58×10^{-3}
	LPC	3.50×10^{-2}	1.30×10^{-3}
10	NR	1.28×10^{-1}	6.51×10^{-2}
	SYW	9.60×10^{-2}	1.63×10^{-2}
	LPC	1.04×10^{-1}	9.27×10^{-3}
0	NR	1.23	1.10×10^{1}
	SYW	6.67	6.45×10^{3}
	LPC	6.28×10^{-1}	5.23×10^{-2}

improvement seen in the lower order cases for the NR estimates becomes apparent when the average statistics are used. Designating these measures as $L[\overline{\hat{a}(k)}]$ and $\%L[\overline{\hat{a}(k)}]$, the results are listed in Table 4-32. Figure 4-28a) is a plot of the $L[\overline{\hat{a}(k)}]$ data in Table 4-32 versus SNR. The $\%L[\overline{\hat{a}(k)}]$ data in Table 4-32 is plotted in Figure 4-28b). The following points are noted:

1) The LPC estimates have the smallest error at ∞ and 30 dB SNR's.
2) For SNR's below 30 dB, the NR estimates have the smallest error.
3) The NR estimates are superior to the SYW estimates except at 10 dB.

Data at -10 dB SNR is not presented because of the lengthy computation time for the NR method for the AR(4) process. As seen in the first and second order cases, all methods do poorly at -10 dB SNR. From Figure 4-28, it is evident that the NR and SYW methods again provide an extension of the successful operating range when estimating AR parameters in white noise. Using the distance measure curves in Figures 4-28 a) and b), the NR estimate error at 0 dB is equal to the LPC estimate error at approximately 22 dB. The estimates from the SYW method provide a 10 dB improvement. The SYW estimates at 0 dB are significantly poorer than the NR estimates, however. The results achieved by applying the distance measures to the average values of the

Table 4-32

$\overline{L[\hat{a}(k)]}$ and $\overline{\%L[\hat{a}(k)]}$ Distance Measures for the AR(4) Process

SNR	Method	$\overline{L[\hat{a}(k)]}$	$\overline{\%L[\hat{a}(k)]}$
∞	NR	6.29×10^{-5}	4.49×10^{-4}
	SYW	1.19×10^{-4}	6.72×10^{-4}
	LPC	6.18×10^{-5}	2.66×10^{-4}
30	NR	7.51×10^{-5}	5.18×10^{-4}
	SYW	1.20×10^{-4}	6.82×10^{-4}
	LPC	7.33×10^{-5}	3.33×10^{-4}
20	NR	6.48×10^{-5}	4.46×10^{-4}
	SYW	1.24×10^{-4}	7.04×10^{-4}
	LPC	3.08×10^{-4}	1.80×10^{-3}
10	NR	1.66×10^{-4}	7.69×10^{-4}
	SYW	1.23×10^{-4}	6.12×10^{-4}
	LPC	9.97×10^{-3}	6.31×10^{-2}
0	NR	2.05×10^{-4}	1.13×10^{-3}
	SYW	8.34×10^{-3}	5.98×10^{-2}
	LPC	2.15×10^{-1}	1.96

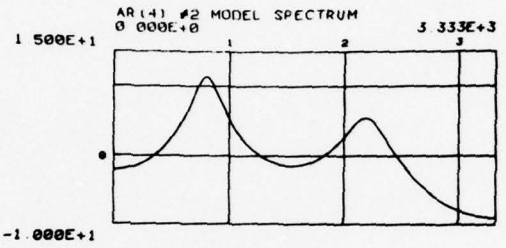


Figure 4-27: Inverse spectrum of AR(4) model

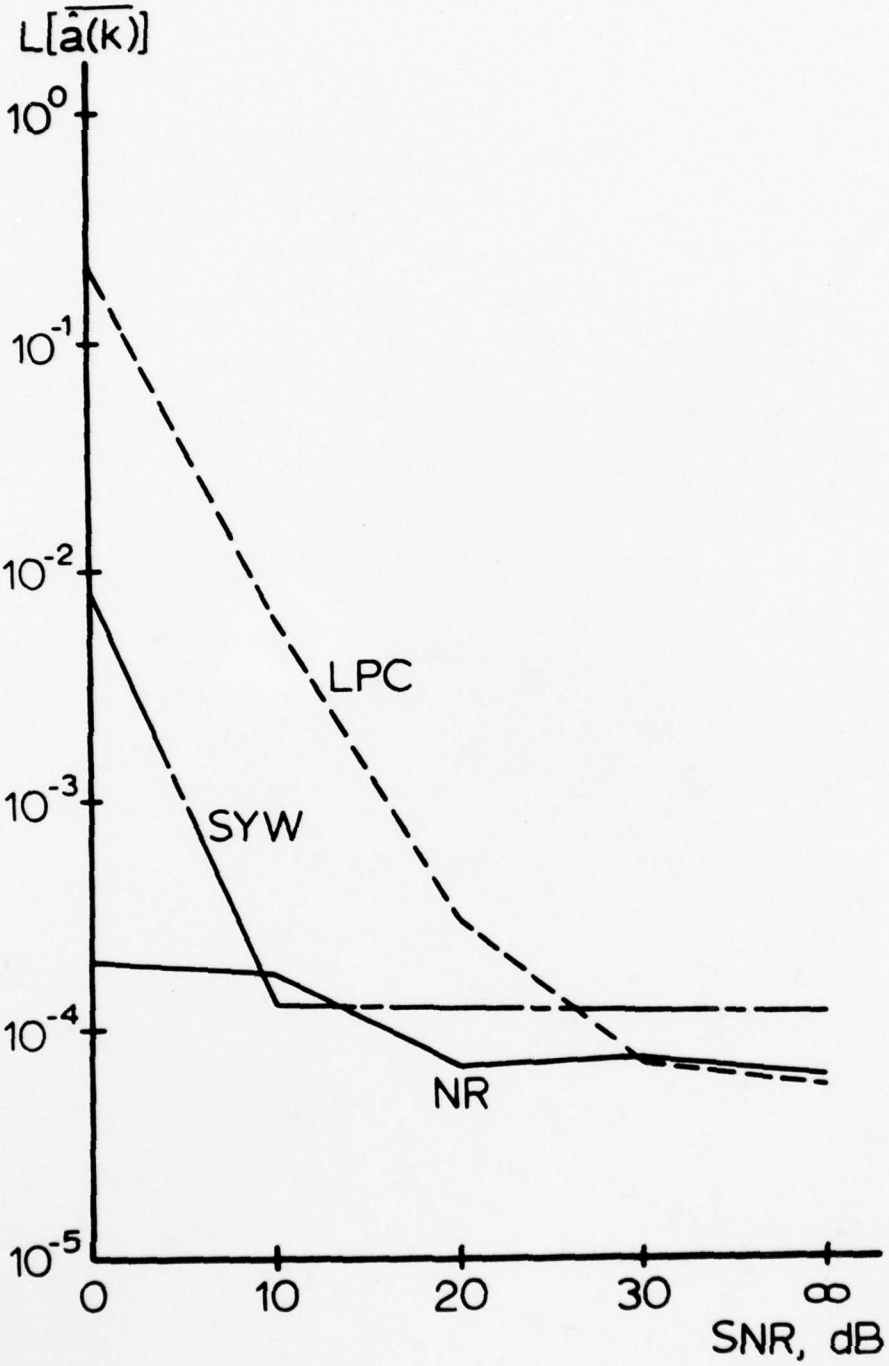


Figure 4-28: Distance measures for three estimates of the AR(4) model coefficients

a) $L[\overline{\hat{a}(k)}]$

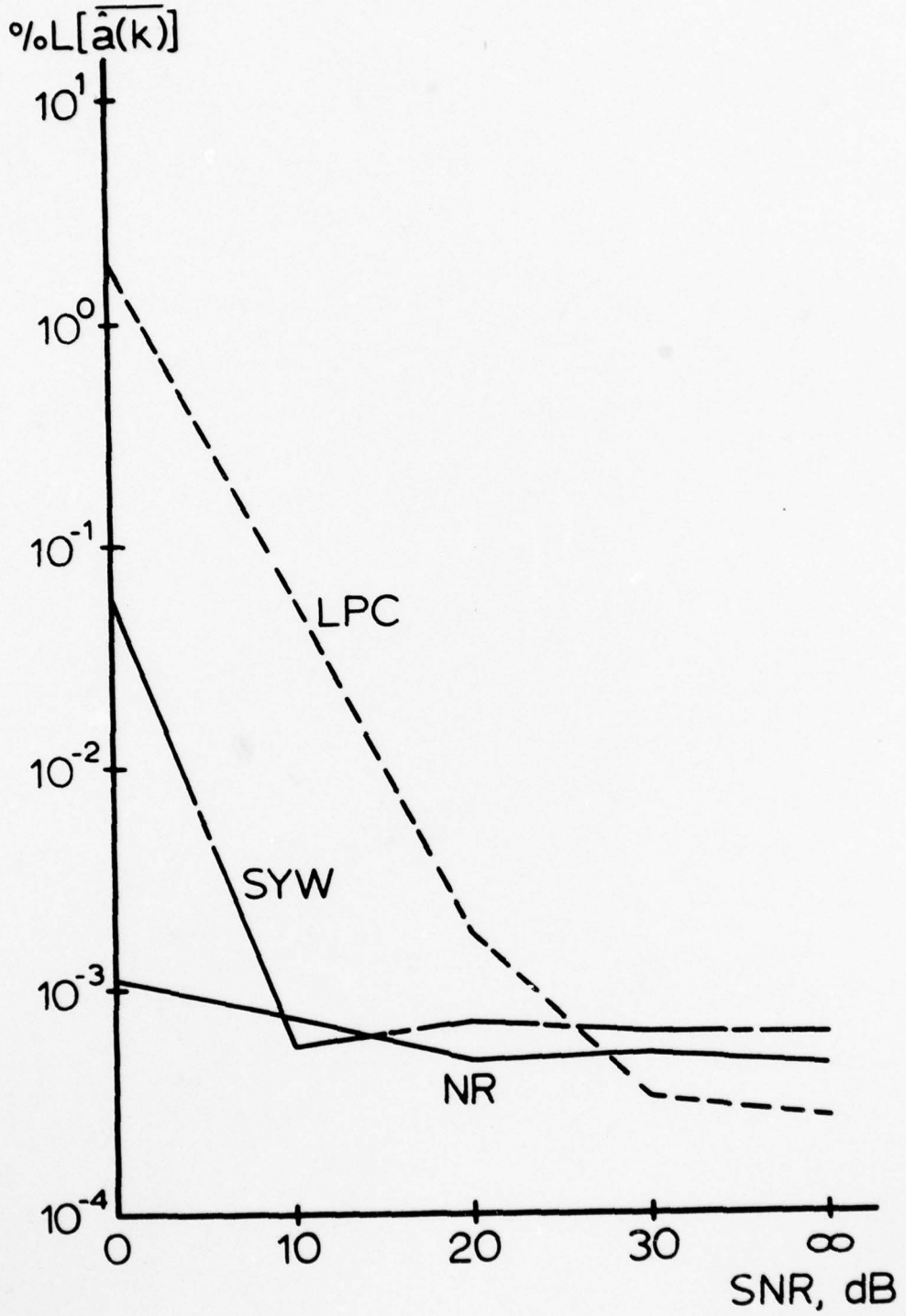


Figure 4-28: b) %L[$\overline{\hat{a}(k)}$]

coefficient estimates again point out the need to use the estimates from individual frames with caution.

CHAPTER 5

CONCLUSIONS

Summary

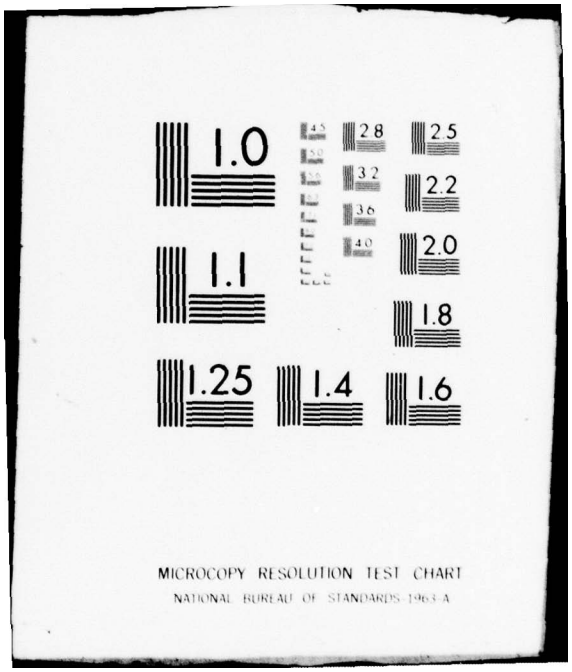
The first experimental results in the preceding chapter illustrate the effects of additive white noise on the sample spectrum of a frame of voiced speech. These data, shown in Figures 4-2 a)-e), also show the 10 pole LPC fit to the sample spectrum. The results for this frame of speech are given for several SNR's. Using the same frame of speech, Figure 4-3 presents the auto- and cross-correlations obtained from the data and four 10 pole LPC spectra, including the LPC spectrum arrived at by assuming the signal and noise are uncorrelated. The data plotted in Figure 4-3 demonstrate the risk associated with assuming independence between the signal and noise.

The next set of experiments tests the applicability of the mode 1 estimation procedure due to Steiglitz [35]. That algorithm is used to estimate the parameters of a 10 pole, 2 zero model from data generated using three different inputs. The input sequences used to drive the model are: 1) an impulse, 2) an impulse train, and 3) a white noise sequence. The results, plotted in Figures 4-7 through 4-17, show this method is useful for impulse and

AD-A068 749 UTAH UNIV SALT LAKE CITY DEPT OF COMPUTER SCIENCE F/G 17/2
ESTIMATION OF THE PARAMETERS OF AN AUTOREGRESSIVE PROCESS IN TH--ETC(U)
DEC 78 W J DONE N00173-77-C-0041
UNCLASSIFIED UTEC-CSC-79-021 NL

3 OF 3
AD
A068749

END
DATE
FILMED
6-79
DDC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

impulse train excitation. However, the performance is poor for the noise excited case. It is the noise excited case, unfortunately, which is most important in this research.

In the following set of experiments, several estimation algorithms are applied to data generated from an AR(1) process, with the single AR coefficient a(1) = 0.5, that is degraded by additive white noise at various SNR's. These experiments test the validity of the AR-to-ARMA transformation model for the first order case. Results from these tests clearly show the nature of the estimation problem to be the minimization of a two dimensional quadratic surface. The estimate for a(1) obtained from the autocorrelation method of LPC has the smallest variance of the estimators tested, but at SNR's below 20 dB a severe bias is introduced in the estimate. An estimate obtained from a Newton-Raphson implementation of a conditional maximum likelihood formulation provides a superior estimate at SNR's through 0 dB, based on the average of the estimates for a(1). The variance of the NR estimate is larger, however. An estimator referred to as the "shifted" Yule-Walker estimator yields results similar to the NR estimate. This estimator requires operations similar to LPC and only provides estimates for the AR coefficients. The SYW estimate does take into account the MA component of the transformation model, where the LPC method does not. All estimates perform poorly at a -10 dB SNR.

The last two sets of experiments apply the NR, LPC, and SYW estimators to an AR(2) process and an AR(4) process, each degraded by varying levels of additive white noise. Again, the NR and SYW estimates are superior to the LPC estimate in the 0 dB to 30 dB SNR range. In the AR(4) test, however, this improvement is apparent only when the distance measure used is applied to the average values of the estimates. This again emphasizes the importance of averaging the estimates in the NR method.

Contributions

This research illustrates the effect of additive white noise on speech. It also points out the risks of the assumption of uncorrelated signal and noise. This assumption is often made in what the author calls autocorrelation correction methods for noise suppression in LP algorithms.

The major contribution of this work is the experimental verification of the AR-to-ARMA transformation model. This model states that the addition of white noise to an AR process produces a data sequence which is an ARMA process. Test results for this model, presented in Chapter 4, show that estimates for the AR coefficients obtained from algorithms based on the transformation model are superior to those obtained using the autocorrelation method LP algorithm. This superiority, however, is achieved by averaging a large number of estimates. The variance of the

transformation model estimators tested is large at the poorer SNR's. This places the value of AR estimates produced by these methods from a single frame of data in question.

Directions for Future Research

Guidelines for extensions of this research are limited to the AR-to-ARMA transformation model. The experimental data show the value of the model for AR(q) models with q = 1, 2, and 4. Tests on higher order models should be undertaken. Also, different kinds of AR models could be studied. For example, the AR(2) model might have two real roots, or the complex roots can be shifted farther from the unit circle. If the analysis of higher order models, q = 10, for example, is successful and if the problem of the large variance of the estimators can be alleviated, this technique might then be applied to the analysis of speech signals.

Anderson's paper [2] proposes several estimation techniques based on the NR and GN methods. Those using the frequency domain approach are not used in this work. In their most useful form, these methods estimate the AR coefficients and the MA covariances. If there is no reason to explicitly estimate the MA coefficients, as was desired for this research, the frequency domain methods are probably of more value if the nonlinear regression on the AR coefficients is to be used. That operation is based on

the nonlinear relationship between the MA covariances and the AR coefficients that results from the noise model. There could be an advantage in estimating the MA covariances directly. The nonlinear regression suggested by Pagano [30] should be tested to see if it improves the AR parameter estimates.

Also, artificial restrictions were placed on the nature of the experiments in this research. It was assumed in all tests that the orders of the ARMA process are known. In addition, the parameters of the true model were often used as the initial parameter estimates. These assumptions were made to concentrate the experiment on the practicality of the transformation model and the estimation algorithm's ability to produce accurate parameter estimates based on that model. These restrictions must be removed in a practical analysis system. The problems of estimating the process order and initial parameter estimates have been dealt with extensively in the literature [1], [10], [15], [27], [29], [37], and [38]. The assumption that the additive noise is white is also restrictive. The extension of the model to allow n(k) to be non-white introduces the need for additional nonlinear analysis once estimates for the AR parameters for the data are found. The reader is referred to [10] for a discussion of this problem.

One of the experimental parameters noted in tests of the NR method is the number of frames successfully analyzed

by the algorithm. This success is a function of whether or not the Gauss elimination procedure fails in solving for the ARMA parameters. At SNR's of 0 and -10 dB, more frames result in failure of the Gauss elimination method. This indicates increasing instability in the NR algorithm. Some of the estimation procedures discussed in Chapter 2 could cancel this trend.

Finally, the NR algorithm requires much more computation than the LPC or SYW methods. This research does not consider the detailed computational requirements of the algorithms. Further work should take this into account. Efficient FORTRAN coding is used in the programs which implement the algorithms discussed, but programming in assembly language could produce considerable computational savings, as would use of an array processor.

APPENDIX A

In Chapters 2 and 3, mention is made of the Newton-Raphson and Gauss-Newton methods for nonlinear parameter estimation. This appendix discusses the generalized formulation of these methods. If $\underline{\theta}$ is the n x 1 parameter vector and $Q(\underline{\theta})$ is the scalar cost function, then one seeks the appropriate choice for $\underline{\theta}$ which will optimize $Q(\underline{\theta})$. Defining $\underline{g}(\underline{\theta})$ as the n x 1 gradient vector, the i^{th} element of $\underline{g}(\underline{\theta})$ is $g_i(\underline{\theta}) = \partial Q(\underline{\theta})/\partial \theta_i$. Except for the case where $Q(\underline{\theta})$ is linear in $\underline{\theta}$, $\underline{g}(\underline{\theta})$ will also be a function of $\underline{\theta}$. The optimum solution $\underline{\theta}^*$ is found by solving the n equations $g_i(\underline{\theta}^*) = 0$, $i = 1, \ldots, n$.

The NR method proceeds by linearizing about $\underline{\theta}$ the vector form of the following equation:

$$\underline{g}(\underline{\theta}^*) = \underline{g}(\underline{\theta}) + \underline{g}'(\underline{\theta}) \, (\underline{\theta}^* - \underline{\theta}) \, .$$

Setting this equation equal to zero yields

$$\underline{g}(\underline{\theta}) + \underline{g}'(\underline{\theta}) \, (\underline{\theta}^* - \underline{\theta}) = 0 \, .$$

Solving for $\underline{\theta}^*$ gives

$$\underline{\theta}^* = \underline{\theta} - [\underline{g}'(\underline{\theta})]^{-1} \, \underline{g}(\underline{\theta}) \, . \qquad \text{(A.1)}$$

The term $\underline{g}'(\underline{\theta})$ in (A.1) is the derivative of the vector $\underline{g}(\underline{\theta})$ with respect to $\underline{\theta}$ and is an n x n matrix. This term is also the second derivative of $Q(\underline{\theta})$ with respect to $\underline{\theta}$ and

is designated $\underline{H}(\underline{\theta})$, the Hessian of $Q(\underline{\theta})$. The ij^{th} element of $\underline{H}(\underline{\theta})$ is

$$h_{ij}(\underline{\theta}) = \frac{\partial^2 Q}{\partial \theta_j \, \partial \theta_i} .$$

Using the definition for $\underline{H}(\underline{\theta})$, (A.1) becomes

$$\underline{\theta}^* = \underline{\theta} - \underline{H}^{-1}(\underline{\theta}) \, \underline{g}(\underline{\theta}) . \qquad (A.2)$$

This is the usual formulation for the NR method. An initial guess $\underline{\theta}$ is required, and the gradient $\underline{g}(\underline{\theta})$ and the Hessian $\underline{H}(\underline{\theta})$ are evaluated at that initial point. Equation (A.2) is then used to generate a new estimate $\underline{\theta}^*$ for the parameter vector. These steps are usually repeated to form an iterative procedure.

Development of the GN method is based on the assumption that the cost function $Q(\underline{\theta})$ can be written in quadratic form:

$$Q(\underline{\theta}) = \underline{F}^T(\underline{\theta}) \, \underline{F}(\underline{\theta}) , \qquad (A.3)$$

where $\underline{\theta}$ is the n x 1 parameter vector and $\underline{F}(\underline{\theta})$ is an m x 1 vector of nonlinear functions, where $m \geq n$. In this case the n x 1 vector gradient of $Q(\underline{\theta})$ is

$$\underline{g}(\underline{\theta}) = \frac{\partial Q}{\partial \underline{\theta}} = 2 \, \underline{F}'(\underline{\theta}) \, \underline{F}(\underline{\theta}) , \qquad (A.4)$$

where $\underline{F}'(\underline{\theta})$ is the n x m matrix of partial derivatives of $\underline{F}(\underline{\theta})$ with respect to θ. The ij^{th} element of $\underline{F}'(\underline{\theta})$ is

$$f'_{ij}(\underline{\theta}) = \frac{\partial F_j(\underline{\theta})}{\partial \theta_i} ,$$

where $F_j(\underline{\theta})$ is the j^{th} element of the vector $\underline{F}(\underline{\theta})$. The

cost function $Q(\underline{\theta})$ is optimized by setting (A.4) equal to zero. This gives

$$\underline{F}'(\underline{\theta}^*)\ \underline{F}(\underline{\theta}^*) = 0 \qquad \text{(A.5)}$$

as the equation from which the optimal solution $\underline{\theta}^*$ is determined. Linearizing $\underline{F}(\underline{\theta}^*)$ about $\underline{\theta}$ yields

$$\underline{F}(\underline{\theta}^*) = \underline{F}(\underline{\theta}) + \underline{F}'^{T}(\underline{\theta})\ (\underline{\theta}^* - \underline{\theta})\ . \qquad \text{(A.6)}$$

Substituting (A.6) into (A.5) for $\underline{F}(\underline{\theta}^*)$ gives

$$\underline{F}'(\underline{\theta}^*)\ [\underline{F}(\underline{\theta}) + \underline{F}'^{T}(\underline{\theta})\ (\underline{\theta}^* - \underline{\theta})] = 0\ . \qquad \text{(A.7)}$$

If $\underline{\theta}$ is close enough to $\underline{\theta}^*$, then $\underline{F}'(\underline{\theta}^*)$ in (A.7) can be approximated by $\underline{F}'(\underline{\theta})$ and (A.7) becomes

$$\underline{F}'(\underline{\theta})\ [\underline{F}(\underline{\theta}) + \underline{F}'^{T}(\underline{\theta})\ (\underline{\theta}^* - \underline{\theta})] = 0\ .$$

Solving this for $\underline{\theta}^*$, one obtains

$$\underline{\theta}^* = \underline{\theta} - [\underline{F}'(\underline{\theta})\ \underline{F}'^{T}(\underline{\theta})]^{-1}\ \underline{F}'(\underline{\theta})\ \underline{F}(\underline{\theta}) \qquad \text{(A.8)}$$

as the computational procedure required in the GN method. Again, an initial guess for $\underline{\theta}$ is required before $\underline{\theta}^*$ can be computed. Equation (A.8) is usually implemented as an iterative algorithm.

APPENDIX B

In the literature review presented in Chapter 2, three sources on the Gauss-Newton and modified GN methods are given [16], [17], [19]. The GN formulation presented in these sources is recommended for performing the nonlinear regression on the ARMA estimates as suggested by Pagano [30]. The nonlinear regression (NLR) technique needed for the ARMA model approach to parameter estimation attempts to find the vector $\underline{\theta}$ which minimizes $\underline{e}$ in

$$\underline{z} = \underline{f}(\underline{\theta}) + \underline{e} \ . \tag{B.1}$$

If the minimization is accomplished in the least squares sense, the loss function measuring the performance of a particular $\underline{\theta}$ is given by

$$Q(\underline{\theta}) = \sum_{k=1}^{2q+1} [z_k - f_k(\underline{\theta})]^2 \ , \tag{B.2}$$

where the $f_k(\underline{\theta})$, $k = 1, \ldots, 2q+1$ are the nonlinear relationships which map from the $q+2$ parameters of $\underline{\theta}$ to the $2q+1$ parameters of $\underline{z}$. The linear terms of the Taylor expansion for $f_k(\underline{\theta}^*)$ about $\underline{\theta}$ are

$$f_k(\underline{\theta}^*) \simeq f_k(\underline{\theta}) + \sum_{j=1}^{q+2} (\theta_j^* - \theta_j) \, f_k^{(j)}(\underline{\theta}) \ , \tag{B.3}$$

$k = 1, \ldots, 2q+1$. In (B.3) $f_k^{(j)}(\underline{\theta})$ indicates the partial derivative of $f_k(\underline{\theta})$ with respect to the j^{th} component of $\underline{\theta}$.

Denoting the partial derivative of $Q(\underline{\theta})$ with respect to the i^{th} parameter of $\underline{\theta}$ as $Q^{(i)}(\underline{\theta})$,

$$Q^{(i)}(\underline{\theta}) = -2 \sum_{k=1}^{2q+1} [z_k - f_k(\underline{\theta})] \, f_k^{(i)}(\underline{\theta}) \, , \qquad (B.4)$$

$i = 1, \ldots, q+2$. The least squares equations are obtained by setting $Q^{(i)}(\underline{\theta}) = 0$, for $i = 1, \ldots, q+2$, and solving for the solution $\underline{\theta}^*$, as in

$$\sum_{k=1}^{2q+1} [z_k - f_k(\underline{\theta}^*)] \, f_k^{(i)}(\underline{\theta}^*) = 0 \, . \qquad (B.5)$$

Substituting (B.3) into (B.5) for $f_k(\underline{\theta}^*)$ gives, for $i = 1, \ldots, q+2$,

$$\sum_{k=1}^{2q+1} [z_k - f_k(\underline{\theta}) - \sum_{j=1}^{q+2} (\theta_j^* - \theta_j) \, f_k^{(j)}(\underline{\theta})] \, f_k^{(i)}(\underline{\theta}^*) = 0 \qquad (B.6)$$

or

$$\sum_{k=1}^{2q+1} [z_k - f_k(\underline{\theta})] \, f_k^{(i)}(\underline{\theta}^*) = \sum_{k=1}^{2q+1} \sum_{j=1}^{q+2} (\theta_j^* - \theta_j) \, f_k^{(j)}(\underline{\theta}) \, f_k^{(i)}(\underline{\theta}^*) \, . \qquad (B.7)$$

Define $D_j = \theta_j^* - \theta_j$ and (B.7), after changing the order of summation, becomes

$$\sum_{j=1}^{q+2} D_j \sum_{k=1}^{2q+1} f_k^{(j)}(\underline{\theta}) \, f_k^{(i)}(\underline{\theta}) = \sum_{k=1}^{2q+1} [z_k - f_k(\underline{\theta})] \, f_k^{(i)}(\underline{\theta}) \, , \qquad (B.8)$$

$i = 1, \ldots, q+2$. In deriving (B.8) from (B.7), all terms $f_k^{(i)}(\underline{\theta}^*)$ are evaluated at $\underline{\theta}$. By using (B.8) and the relationship $\underline{\theta}^* = \underline{\theta} + \underline{D}$, the procedure for obtaining the new estimate of $\underline{\theta}$ is defined. The process is made iterative by letting $\underline{\theta} = \underline{\theta}^*$ and repeating the process described by (B.8). The process described above is the Gauss-Newton method for performing a nonlinear regression.

The modified Gauss-Newton technique can also be used [19], [29]. If $\underline{J}(\underline{\theta})$ is defined as the Jacobian matrix, where the ij^{th} element of $\underline{J}(\underline{\theta})$ is given by $J_{ij} = \partial f_i(\underline{\theta})/\partial \theta_j$, then (B.8) can be written in matrix form as

$$\underline{J}^T(\underline{\theta})\ \underline{J}(\underline{\theta})\ \underline{D} = \underline{J}^T(\underline{\theta})\ [\underline{z} - \underline{f}(\underline{\theta})]\ . \qquad (B.9)$$

With this notation established, the modified Gauss-Newton method for NLR can be written as

$$\omega\ [\underline{J}^T(\underline{\theta})\ \underline{J}(\underline{\theta}) + \lambda\ \underline{I}]\ \underline{D} = \underline{J}^T(\underline{\theta})\ [\underline{z} - \underline{f}(\underline{\theta})]\ . \qquad (B.10)$$

The parameter ω in (B.10) is chosen to ensure that $Q(\underline{\theta}^*) \leq Q(\underline{\theta})$. The parameter λ is selected to guarantee the invertibility of $[\underline{J}^T(\underline{\theta})\ \underline{J}(\underline{\theta}) + \lambda\ \underline{I}]$.

For the general Gauss-Newton method described above, knowledge of the nonlinear functions $f_k(\underline{\theta})$ and the partial derivatives $f_k^{(i)}(\underline{\theta})$ is required. In the specific case of an AR process obscured by additive white noise, the functions $f_k(\underline{\theta})$ are derived from

$$R_{yy}(k) = \sigma_\varepsilon^2\ \delta(k) + \sigma_n^2 \sum_{i=0}^{q-k} a(i)\ a(i+k)\ , \qquad (B.11)$$

where $\delta(k)$ is the Kronecker delta function and $a(0) = 1.0$. The partial derivatives of $f_k(\underline{\theta})$ are taken with respect to the elements of $\underline{\theta}$: $a(1), \ldots, a(q), \sigma_\varepsilon^2$, and σ_n^2. The $f_k(\underline{\theta})$ are given by

$$f_j(\underline{\theta}) = a(j)\ ,$$

$$f_{q+1}(\underline{\theta}) = \sigma_\varepsilon^2 + \sigma_n^2 \sum_{i=0}^{q} a^2(i)\ ,$$

$$f_{q+1+j}(\underline{\theta}) = \sigma_n^2 \sum_{i=0}^{q-j} a(i)\ a(i+j)\ ,$$

for $j = 1, \ldots, q$. If $\partial f_k(\underline{\theta})/\partial\theta_i = f_k^{(i)}(\underline{\theta})$ and $\theta_i = a(i)$, $i = 1, \ldots, q$, $\theta_{q+1} = \sigma_\varepsilon^2$, and $\theta_{q+2} = \sigma_n^2$, then the $f_k^{(i)}(\underline{\theta})$ are

$$f_k^{(i)}(\underline{\theta}) = \delta(k-i)\ ;$$

$$f_{q+1}^{(j)}(\underline{\theta}) = 2\ \sigma_n^2\ a(j)\ ;$$

$$f_{q+1}^{(q+1)}(\underline{\theta}) = 1\ ;$$

$$f_{q+1}^{(q+2)}(\underline{\theta}) = \sum_{j=0}^{q} a^2(j)\ ;$$

$$f_{q+1+k}^{(j)}(\underline{\theta}) = \sigma_n^2\ \delta(k-j) + \sigma_n^2\ \left(\langle a(j+k)\rangle + \langle a(j-k)\rangle\right),$$

$$\langle a(m)\rangle = \begin{cases} a(m), & m = 1, \ldots, q \\ 0, & \text{otherwise} \end{cases}\ ;$$

$$f_{q+1+k}^{(q+1)}(\underline{\theta}) = 0\ ;$$

$$f^{(q+2)}_{q+1+k}(\underline{\theta}) = \sum_{i=0}^{q-k} a(i)\ a(i+k)\ ;$$

for j, k = 1, ..., q and i = 1, ..., q+2. Assembling the $f^{(i)}_j(\underline{\theta})$ into a matrix $\underline{J}(\underline{\theta})$, with the ji^{th} element of $\underline{J}(\underline{\theta})$ being $J_{ji} = f^{(i)}_j(\underline{\theta})$, gives the Jacobian matrix.

APPENDIX C

In Chapter 3, the AR(1)-to-ARMA(1,1) transformation model is discussed in detail, with numerical results for that model presented in Chapter 4. Equations (3.40) and (3.41) of Chapter 3 give the expressions for the parameters b and σ_v^2 of the ARMA(1,1) process obtained by adding white noise to an AR(1) process, with a(1) = a the single AR parameter. Those equations are repeated here as (C.1) and (C.2):

$$b = \frac{1}{2\ \sigma_n^2\ a}\ \left([\sigma_\varepsilon^2 + \sigma_n^2\ (1 + a^2)] \pm \right.$$

$$\left. [[\sigma_\varepsilon^2 + \sigma_n^2\ (1 + a^2)]^2 - 4\ \sigma_n^4\ a^2]^{1/2}\right), \quad \text{(C.1)}$$

$$\sigma_v^2 = \sigma_n^2\ \frac{a}{b}\ . \quad \text{(C.2)}$$

The parameters b and σ_v^2 possess certain properties. For example, it must be shown that $b = b_-$ or $b = b_+$ is real. The parameter σ_v^2 is the variance of the ARMA excitation sequence v(k) and must be a real, positive scalar.

First, the properties of the MA coefficient b are examined. If c is defined as the argument of the radical in (C.1), c becomes

$$c = [\sigma_\varepsilon^2 + \sigma_n^2\ (1 + a^2)]^2 - 4\ \sigma_n^4\ a^2$$

$$c = \sigma_\varepsilon^4 + 2\,\sigma_\varepsilon^2 + \sigma_n^2\,(1 + a^2) + \sigma_n^4\,(1 + a^2)^2 - 4\,\sigma_n^4\,a^2 .$$

Simplifying this expression gives

$$c = \sigma_\varepsilon^4 + 2\,\sigma_\varepsilon^2\,\sigma_n^2\,(1 + a^2) + \sigma_n^4\,(1 - a^2)^2 . \qquad (C.3)$$

By definition, σ_ε^2 is positive, σ_n^2 is non-negative, and a is a real number. In addition, to satisfy the stationarity requirement [10], $|a| < 1.0$. With these properties for a, σ_ε^2, and σ_n^2, one sees that c is real and $c > 0$. This establishes the first property given in Chapter 3: b is real (either b_- or b_+).

Defining

$$b_- = \frac{1}{2\,\sigma_n^2\,a}\left([\sigma_\varepsilon^2 + \sigma_n^2\,(1 + a^2)] - [c]^{1/2}\right)$$

and

$$b_+ = \frac{1}{2\,\sigma_n^2\,a}\left([\sigma_\varepsilon^2 + \sigma_n^2\,(1 + a^2)] + [c]^{1/2}\right) ,$$

note that

$$b_-\,b_+ = \frac{1}{4\,\sigma_n^4\,a^2}\left([\sigma_\varepsilon^2 + \sigma_n^2\,(1 + a^2)]^2 - c\right)$$

$$= \frac{1}{4\,\sigma_n^4\,a^2}\,[4\,\sigma_n^4\,a^2] = 1$$

and $b_+ = 1/b_-$. Because they are reciprocals and real, b_- and b_+ must have the same sign. One can thus deduce from (C.1) that the sign of the numerator in (C.1) is the same for b_- and b_+. This establishes two facts:

1) the sign of b equals the sign of a;

2) $|b_-| < |b_+|$ (for $c > 0$).

Using 2) and the fact that $b_+ = 1/b_-$, one deduces that $|b_-| < 1.0$. This completes the derivation of property 2) in Chapter 3.

Knowing that $\sigma_n^2 \geq 0$ in (C.2) and using fact 1) above, we can state that $\sigma_v^2 \geq 0$ and σ_v^2 is real (since a, b, and σ_n^2 are real), property 3) of Chapter 3. Defining $\sigma_{v+}^2 = \sigma_n^2\ a/b_+$ and $\sigma_{v-}^2 = \sigma_n^2\ a/b_-$, then

$$\sigma_{v+}^2 = \frac{\sigma_n^2\ a}{b_+}$$

$$= \sigma_n^2\ a\ b_-$$

$$= b_-^2\ \left(\frac{\sigma_n^2\ a}{b_-}\right)$$

$$= b_-^2\ \sigma_{v-}^2 \qquad \text{(C.4)}$$

establishes the last property required for the development in Chapter 3.

Having developed the properties for b_-, b_+, σ_{v-}^2, and σ_{v+}^2, we proceed to illustrate how these parameters behave for extremes in SNR. Recall from Chapter 3 that the variance of an AR(1) process is $\sigma_s^2 = \sigma_\varepsilon^2/(1 - a^2)$. If $SNR = \sigma_s^2/\sigma_n^2$, we are concerned about the behavior of the MA parameters as $SNR \to \infty$ $(\sigma_n^2 \to 0)$ and $SNR \to 0$ $(\sigma_n^2 \to \infty)$. The characteristics of these parameters will be developed by looking at b_- and σ_{v-}^2. The parameters b_+ and σ_{v+}^2 can then be characterized by using the properties developed above.

Defining N as the numerator of (C.1), using the minus

sign, and D as the denominator of (C.1), $b_- = N/D$.

Case I: $SNR \to \infty$, $\sigma_n^2 \to 0$

For b_-, consider

$$\lim_{\sigma_n^2 \to 0} b_- = \lim_{\sigma_n^2 \to 0} \frac{N}{D}$$

$$= \frac{\lim_{\sigma_n^2 \to 0} N'}{\lim_{\sigma_n^2 \to 0} D'} = \frac{0}{2\,a} = 0 \ ,$$

where $N' = dN/d\sigma_n^2$ and $D' = dD/d\sigma_n^2$ are required in the use of L'Hospital's rule. The behavior of σ_{v-}^2 is given by

$$\lim_{\sigma_n^2 \to 0} \sigma_{v-}^2 = a \lim_{\sigma_n^2 \to 0} \frac{\sigma_n^2 D}{N}$$

$$= \frac{a \lim_{\sigma_n^2 \to 0} F''}{\lim_{\sigma_n^2 \to 0} N''}$$

$$= a \frac{4\,a}{(4\,a^2/\sigma_\varepsilon^2)} = \sigma_\varepsilon^2 \ ,$$

where $N'' = d^2N/d(\sigma_n^2)^2$ and $F'' = d^2(\sigma_n^2 D)/d(\sigma_n^2)^2$. In summary, as the SNR approaches infinity, $b_- \to 0$ and $\sigma_{v-}^2 \to \sigma_\varepsilon^2$. From $b_+ = 1/b_-$ and $\sigma_{v+}^2 = b_-\,\sigma_{v-}^2$, we see that $b_+ \to \infty$ and $\sigma_{v+}^2 \to 0$.

Case II: $SNR \to 0$, $\sigma_n^2 \to \infty$

As the level of noise increases, b_- approaches a in value:

$$\lim_{\sigma_n^2 \to \infty} b_- = \lim_{\sigma_n^2 \to \infty} \frac{N}{D}$$

$$= \lim_{\sigma_n^2 \to \infty} \frac{N/\sigma_n^2}{2\,a}$$

$$= \frac{2\,a^2}{2\,a} = a \ .$$

The behavior of the variance σ_{v-}^2 is given by

$$\lim_{\sigma_n^2 \to \infty} \sigma_{v-}^2 = \lim_{\sigma_n^2 \to \infty} \sigma_n^2 \frac{a}{b_-}$$

$$= \infty \ .$$

For b_+ and σ_{v+}^2, we have $b_+ \to 1/a$ and $\sigma_{v+}^2 \to \infty$ as SNR $\to 0$. The results for the behavior of these parameters from Case I and Case II is summarized in Table 3-1.

APPENDIX D

Anderson, in his presentation of estimation procedures for ARMA models [2], uses a matrix notation to simplify the equations involved. For the time domain approach a matrix operator is required which will impose the assumption of zero initial conditions on the ARMA process $x(k)$ and the excitation sequence $v(k)$. If $x(0), \ldots, x(N-1)$ are the observed data, the operator is the $N \times N$ matrix $\underline{L}$, given by

$$\underline{L} = \begin{bmatrix} 0 & 0 \\ \underline{I}_{N-1} & 0 \end{bmatrix},$$

where $\underline{I}_{N-1}$ is the $(N-1) \times (N-1)$ identity matrix. If $N = 5$, for example, $\underline{L}$ is

$$\underline{L} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

For this $N = 5$ case, $\underline{L}^2$ is found to be

$$\underline{L}^2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ \underline{I}_3 & 0 \end{bmatrix} .$$

In general, $\underline{L}^i$ is

$$\underline{L}^i = \begin{bmatrix} 0 & 0 \\ \underline{I}_{N-i} & 0 \end{bmatrix} .$$

The effect of pre-multiplying a vector by $\underline{L}^i$ is now examined. Forming the data vector $\underline{x}$, $\underline{x} = [x(0) \ldots x(N-1)]^T$, we have $\underline{L}^i\underline{x} = [0 \ldots 0\ x(0) \ldots x(N-1-i)]^T$. The multiplication by $\underline{L}^i$ shifts the elements of the vector $\underline{x}$ down i places, introducing zeros in the first i positions.

In scalar form the ARMA(q,p) process x(k) is given by

$$\sum_{i=0}^{q} a(i)\ x(k-i) = \sum_{j=0}^{p} b(j)\ v(k-j) , \qquad (D.1)$$

with $a(0) = b(0) = 1$ and $x(k) = v(k) = 0$ for $k < 0$. Noting that $\underline{L}^0 = \underline{I}_N$, the matrix formulation for (D.1) is

$$\sum_{i=0}^{q} a(i)\ \underline{L}^i\ \underline{x} = \sum_{j=0}^{p} b(j)\ \underline{L}^j\ \underline{v} . \qquad (D.2)$$

Defining the matrices $\underline{A}$ and $\underline{B}$ as

$$\underline{A} = \sum_{i=0}^{q} a(i)\ \underline{L}^i$$

and

$$\underline{B} = \sum_{j=0}^{p} b(j)\ \underline{L}^j ,$$

(D.2) becomes

$$\underline{A}\,\underline{x} = \underline{B}\,\underline{v} \ . \qquad \text{(D.3)}$$

To see what form $\underline{A}$ and $\underline{B}$ have, consider the case where $p = 2$ and $N = 5$. For this example the matrix $\underline{B}$ is

$$\underline{B} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ b(1) & 1 & 0 & 0 & 0 \\ b(2) & b(1) & 1 & 0 & 0 \\ 0 & b(2) & b(1) & 1 & 0 \\ 0 & 0 & b(2) & b(1) & 1 \end{bmatrix} .$$

As seen in this example, the elements along each diagonal are equal and the matrix is lower triangular. The matrix $\underline{A}$ has the same form.

As described in Chapter 3, the parameter estimation procedure requires the generation of n x 1 vectors of the form $\underline{\gamma} = \underline{B}^{-1}\underline{x}$. Given the matrix $\underline{B}$ as described above, we are interested in the structure of $\underline{B}^{-1}$. Since $\underline{B}$ is lower triangular, $\underline{B}^{-1}$ will also be lower triangular. Also, the elements along the diagonals of $\underline{B}^{-1}$ are equal. Designating the first column of $\underline{B}^{-1}$ as the vector $\underline{\beta} = [\beta(0)\ \beta(1)\ \ldots\ \beta(N-1)]^T$, we have $\underline{B}\underline{\beta} = [1\ 0\ \ldots\ 0]^T$. Equating the elements of the left and right hand sides of this equation and solving for the $\beta(k)$, $k = 0, \ldots, N-1$, gives

$$\beta(0) = 1 \ ,$$

$$\beta(k) = -\sum_{i=1}^{k} b(i)\ \beta(k-i), \quad k = 1, \cdots, p-1 \ ,$$

$$\beta(k) = -\sum_{i=1}^{p} b(i)\ \beta(k-i), \quad k = p, \cdots, N-1 .$$

For the p = 2, N = 5 example used above, $\underline{B}^{-1}$ is of the form

$$\underline{B}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ \beta(1) & 1 & 0 & 0 & 0 \\ \beta(2) & \beta(1) & 1 & 0 & 0 \\ \beta(3) & \beta(2) & \beta(1) & 1 & 0 \\ \beta(4) & \beta(3) & \beta(2) & \beta(1) & 1 \end{bmatrix} .$$

The matrix $\underline{A}^{-1}$ is formed in the same manner as $\underline{B}^{-1}$ and has the same properties: lower triangular and equal elements along the diagonals.

APPENDIX E

In Chapter 3 an expression for the estimator of the variance of a noise sequence is developed. Two measures of the usefulness of this estimator are its sample mean and variance. The development of the sample variance of the estimator requires knowledge of the fourth moment of a normal random variable.

If n is a r.v. with distribution $N(\mu,\sigma^2)$, then the moment generating function of n is

$$M_n(t) = \exp\{\mu t + \frac{1}{2} t^2 \sigma^2\} . \qquad (E.1)$$

Differentiating (E.1) with respect to t gives

$$M_n^{(1)}(t) = \frac{d}{dt} M_n(t) = (\mu + t\sigma^2) M_n(t) .$$

The i^{th} derivative of $M_n(t)$ with respect to t is

$$M_n^{(i)}(t) = \frac{d^i}{dt^i} M_n(t)$$

$$= (\mu + t\sigma^2) M_n^{(i-1)}(t)$$

$$+ (i - 1) M_n^{(i-2)}(t) . \qquad (E.2)$$

Equation (E.2) is valid for $i \geq 2$.

The i^{th} moment of n is found by evaluating $M_n^{(i)}(t)$ at $t = 0$. Thus, the fourth moment of n is

$$M_n^{(4)}(0) = \mu^4 + 6\,\mu^2\,\sigma^2 + 3\,\sigma^4 .$$

If $E[n] = \mu = 0$ as in Chapter 3, the fourth moment of n is given by

$$E[n^4] = M_n^{(4)}(0) = 3\,\sigma^4 .$$

REFERENCES

[1] H. Akaike, "Maximum Likelihood Identification of Gaussian Autoregressive Moving Average Models," Biometrika, vol. 60, no. 2, 1973, pp. 255-265.

[2] T. W. Anderson, "Estimation for Autoregressive Moving Average Models in the Time and Frequency Domains," Annals of Statistics, vol. 5, no. 5, 1977, pp. 842-865.

[3] B. S. Atal and S. L. Hanauer, "Speech Analysis and Synthesis by Linear Prediction of the Speech Wave," J. Acoustical Soc. of America, vol. 50, no. 2, August 1971, pp. 637-655.

[4] M. A. Atashroo, "Pole-Zero Modeling and its Application to Speech Processing," Computer Science Dept., Univ. of Utah, Salt Lake City, UTEC-CSc-76-271, August 1976.

[5] Y. Bard, "Comparison of Gradient Methods for the Solution of Nonlinear Parameter Estimation Problems," SIAM Journal of Numerical Analysis, vol. 7, no. 1, March 1970, pp. 157-186.

[6] Y. Bard, Nonlinear Parameter Estimation, Academic Press, New York, 1974.

[7] J. V. Beck and K. J. Arnold, Parameter Estimation in Engineering and Science, Wiley, New York, 1977.

[8] S. F. Boll, "A Priori Digital Speech Analysis," Computer Science Dept., Univ. of Utah, Salt Lake City, UTEC-CSc-73-123, March 1973.

[9] S. F. Boll, "Improving Linear Prediction Analysis of Noisy Speech by Predictive Noise Cancellation," 1977 IEEE International Conf. of Acoustics, Speech, and Signal Processing Record, May 1977, pp. 10-12.

[10] G. E. Box and G. M. Jenkins, Time Series Analysis, Forecasting and Control, Holden-Day, San Francisco, 1976.

[11] R. W. Christiansen, "Word Recognition in Continuous

Speech Using Linear Prediction Analysis," Computer Science Dept., Univ. of Utah, Salt Lake City, UTEC-CSc-76-226, August 1976.

[12] J. Durbin, "Efficient Estimation of Parameters in Moving Average Models," Biometrica, vol. 46, 1959, pp. 306-316.

[13] D. Graupe, et. al., "Identification of Autoregressive Moving-Average Parameters of Time Series," IEEE Trans. on Automatic Control, vol. AC-20, no. 1, Feb. 1975, pp. 104-107.

[14] N. K. Gupta and R. K. Mehra, "Computational Aspects of Maximum Likelihood Estimation and Reduction in Sensitivity Function Calculations," IEEE Trans. on Automatic Control, vol. AC-19, no. 6, Dec. 1974, pp. 774-783.

[15] E. J. Hannan, "The Estimation of Mixed Moving Average Autoregressive Systems," Biometrika, vol. 56, no. 3, 1969, pp. 579-593.

[16] H. O. Hartley, "The Modified Gauss-Newton Method for the Fitting of Non-Linear Regression Functions by Least Squares," Technometrics, vol. 3, no. 2, May 1961, pp. 269-280.

[17] H. O. Hartley and A. Booker, "Nonlinear Least Squares Estimation," Annals of Mathematical Statistics, vol. 36, 1965, pp. 638-650.

[18] F. Itakura, "Minimum Prediction Residual Principle Applied to Speech Recognition," IEEE Trans. on Acoustics, Speech, and Signal Processing, vol. ASSP-23, no. 1, Feb. 1975, pp. 67-72.

[19] R. I. Jennrich, "Asymptotic Properties of Non-Linear Least Squares Estimators," Annals of Mathematical Statistics, vol. 40, no. 2, April 1969, pp. 633-643.

[20] T. Kailath, "A View of Three Decades of Linear Filtering Theory," IEEE Trans. on Information Theory, vol. IT-20, no. 2, March 1974, pp. 146-181.

[21] R. L. Kashyap and R. E. Nasburg, "Parameter Estimation in Multivariate Stochastic Difference Equations," IEEE Trans. on Automatic Control, vol. AC-19, no. 6, Dec. 1974, pp. 784-797.

[22] J. S. Lim and A. V. Oppenheim, "All-Pole Modeling of Degraded Speech," IEEE Trans. on Acoustics, Speech, and Signal Processing, vol. ASSP-26, no. 3, June 1978, pp. 197-210.

[23] J. Makhoul, "Linear Prediction: A Tutorial Review," Proc. of the IEEE, vol. 63, no. 4, April 1975, pp. 561-580.

[24] J. Makhoul, "Spectral Linear Prediction: Properties and Applications," IEEE Trans. on Acoustics, Speech, and Signal Processing, vol. ASSP-23, no. 3, June 1975, pp. 283-296.

[25] D. W. Marquardt, "An Algorithm for Least-Squares Estimation of Nonlinear Parameters," SIAM, vol. 11, no. 2, June 1963, pp. 431-441.

[26] R. J. McAulay, "Optimum Classification of Voiced Speech, Unvoiced Speech and Silence in the Presence of Noise and Interference," Mass. Inst. of Tech., Lincoln Laboratory, Technical Note 1976-7, June 1976.

[27] W. Murray, ed., Numerical Methods for Unconstrained Optimization, Academic Press, New York, 1972.

[28] C. R. Nelson, "The First-Order Moving Average Process," Journal of Econometrics, vol. 2, 1974, pp. 121-141.

[29] J. M. Ortega and W. C. Rheinboldt, Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, New York, 1970.

[30] M. Pagano, "Estimation of Models of Autoregressive Signal Plus White Noise," Annals of Statistics, vol. 2, no. 1, 1974, pp. 99-108.

[31] E. Parzen, "Some Recent Advances in Time Series Modeling," IEEE Trans. on Automatic Control, vol. AC-19, no. 6, Dec. 1974, pp. 723-730.

[32] M. J. D. Powell, "A Survey of Numerical Methods for Unconstrained Optimization," SIAM Review, vol. 12, no. 1, Jan. 1970, pp. 79-97.

[33] M. R. Sambur and N. S. Jayant, "LPC Synthesis Starting from White Noise Corrupted or Differentially Quantized Speech," 1976 IEEE International Conf. on Acoustics, Speech, and Signal Processing Record, April 1976, pp. 237-240.

[34] M. R. Sambur and N. S. Jayant, "LPC Analysis/Synthesis from Speech Inputs Containing Quantizing Noise or Additive White Noise," IEEE Trans. on Acoustics, Speech, and Signal Processing, vol. ASSP-24, no. 6, Dec. 1976, pp. 488-494.

[35] K. Steiglitz, "On the Simultaneous Estimation of Poles and Zeros in Speech Analysis," IEEE Trans. on Acoustics, Speech, and Signal Processing, vol. ASSP-25, no. 3, June 1977, pp. 229-234.

[36] K. Steiglitz and L. E. McBride, "A Technique for the Identification of Linear Systems," IEEE Trans. on Automatic Control, vol. AC-10, no. 4, July 1965, pp. 461-464.

[37] H. Tong, "Autoregressive Model Fitting with Noisy Data by Akaike's Information Criterion," IEEE Trans. on Information Theory, vol. IT-21, no. 4, July 1975, pp. 476-480.

[38] H. Tong, "More on Autoregressive Model Fitting with Noisy Data by Akaike's Information Criterion," IEEE Trans. on Information Theory, vol. IT-23, no. 3, May 1977, pp. 409-410.

[39] A. M. Walker, "Some Consequences of Superimposed Error in Time Series Analysis," Biometrika, vol. 47, no. 1, 1960, pp. 33-43.

[40] A. M. Walker, "Large Sample Estimation of Parameters for Moving Average Models," Biometrica, vol. 48, 1961, pp 343-357.

[41] B. Widrow, et. al., "Adaptive Noise Cancelling: Principles and Applications," Proc. of the IEEE, vol. 63, no. 12, Dec. 1975, pp. 1692-1716.

[42] B. Yegnanarayana, "Effect of Noise and Distortion in Speech on Parametric Extraction," 1976 IEEE International Conf. on Acoustics, Speech, and Signal Processing Record, April 1976, pp. 336-339.