

1988

The role of zeros in linear prediction of speech /

Christopher James Gosnell
Lehigh University

Follow this and additional works at: <https://preserve.lehigh.edu/etd>



Part of the [Electrical and Computer Engineering Commons](#)

Recommended Citation

Gosnell, Christopher James, "The role of zeros in linear prediction of speech /" (1988). *Theses and Dissertations*. 4871.
<https://preserve.lehigh.edu/etd/4871>

This Thesis is brought to you for free and open access by Lehigh Preserve. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of Lehigh Preserve. For more information, please contact preserve@lehigh.edu.

THE ROLE OF ZEROS IN
LINEAR PREDICTION OF SPEECH

by

Christopher James Gosnell

A Thesis

Presented to the Graduate Committee

of Lehigh University

in Candidacy for the Degree of

Master of Science

in

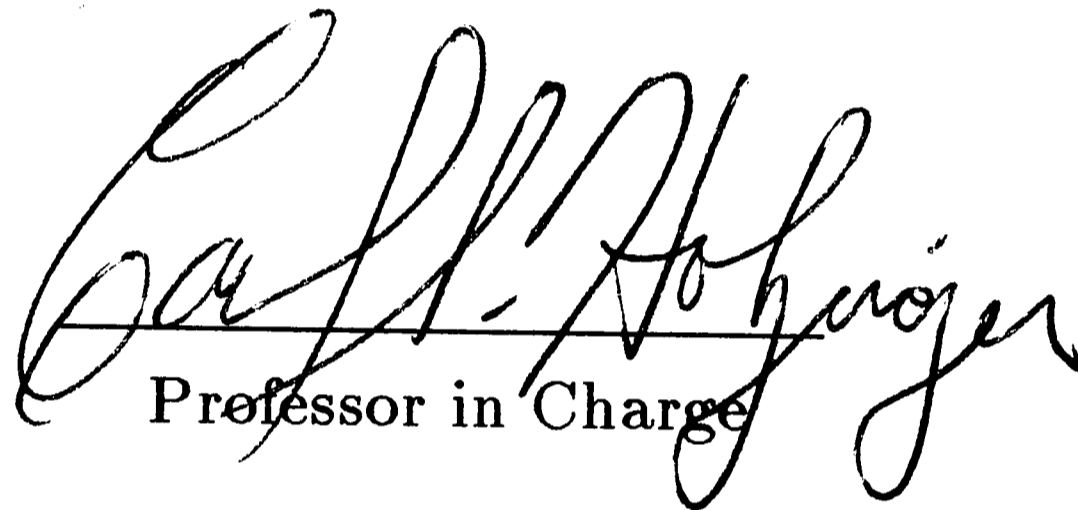
Electrical Engineering

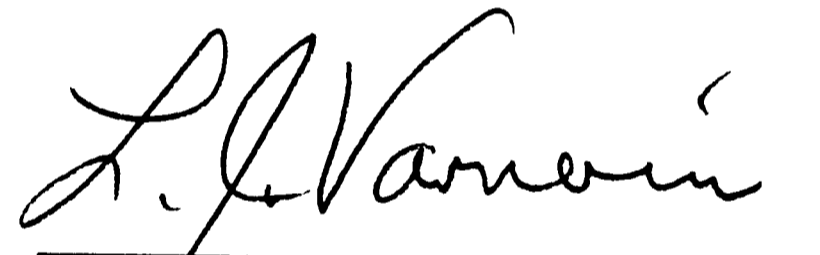
Lehigh University

1988

This thesis is accepted and approved in partial fulfilment of the requirements for the degree of Master of Science.

MAY 20, 1988
(date)


Professor in Charge


Chairman of Department

Acknowledgements

I would like to acknowledge the assistance, advice and motivation given to me by my advisor, Professor Carl S. Holzinger.

Table of Contents

Chapter 1 : Introduction	2
The Speech Model	3
Poles and Zeros	8
Performance Measures	12
Chapter 2 : LPC Filter Configuration	14
The All-Pole Configuration	14
The Pole-Zero Configuration	19
Chapter 3 : The Method of Steepest Descent	23
An Example	25
Application to Linear Predictive Coding	26
Chapter 4 : Experimental Background	35
Speech Data	35
Computer Programs	36
Chapter 5 : Simulations and Results	37
Test 1	37
Test 2	40
Test 3	42
Test 4	44
Test 5	46

Test 6	48
Test 7	50
Test 8	55

Chapter 6 : Conclusion

References	64
Appendix A	67
Appendix B	70
Appendix C	75

List of Figures

Fig 1.1	X-ray of the human vocal Apparatus	3
Fig 1.2	Model of speech production	5
Fig 1.3	Simplified model for speech production	5
Fig 2.1	Block diagram of an all-pole synthesis filter	15
Fig 2.2	Possible block diagram of an all-pole analysis filter	16
Fig 2.3	Block diagram of an all-pole analysis filter	17
Fig 2.4	All-pole analysis filter showing quantization noise	18
Fig 2.5	Block diagram of a pole-zero synthesis filter	20
Fig 2.6	Block diagram of a pole-zero analysis filter	21
Fig 3.1	Minimization by the method of steepest descent	23
Fig 3.2	LPC performance improvement with number of iterations	34
Fig 5.1	Gains for voiced speech	39
Fig 5.2	Gains for unvoiced speech	41
Fig 5.3	Gains for synthetic speech (glottal)	43
Fig 5.4	Gains for synthetic speech (impulse)	45
Fig 5.5	Gains for synthetic speech (impulse)	47
Fig 5.6	Gains for voiced speech	49
Fig 5.7	Reconstruction SNR for voiced speech	52
Fig 5.8	Reconstruction SNR for voiced samples 1 & 5	53
Fig 5.9	Reconstruction SNR for unclipped voiced speech	55
Fig 5.10	Reconstruction SNR for synthetic speech (glottal)	56

Abstract

This study investigates the nature of the role that zeros play in a linear predictive representation of speech. A configuration of pole-zero LPC filters is proposed and the equations are solved leading to near-optimal coefficients by the method of steepest descent.

It is shown that pole-zero representation of speech is superior to all-pole representations for the performance measures used. The optimal combination is found to be where one coefficient is used to model a single pole and the remaining coefficients represent zeros. The zeros that are important to linear prediction are traced to the glottal pulse and a method is suggested to incorporate this into a new approach to linear prediction of speech.

Chapter 1

Introduction

Speech waveforms are transmitted, stored and analyzed probably more than any other electronic signal. There is clearly a benefit in being able to represent these waveforms in a manner that is economical in terms of the storage space necessary to maintain a record of it, or in terms of the amount of information that must be passed from the source to the receiver of the speech signal. In 1971, B.S. Atal [Ata71] presented a most economical method of representing the speech waveform, known as Linear Predictive Coding or LPC. This paper details the effects of a proposed refinement to LPC.

Speech, when it is naturally produced, is a continuous sound waveform. This is easily converted into a continuous voltage signal using a microphone. Manipulation of this voltage waveform is often best done using digital computing techniques and thus the continuous, or analog, waveform is converted into a series of discrete values by sampling the waveform periodically. The Nyquist theorem shows that if the waveform is sampled at a frequency at least twice as great as the highest frequency present in the signal, then no information is lost by this process [Opp75].

Linear predictive analysis is built around the idea that the current value of a speech sample can be estimated by taking a weighted combination of some limited number of the previous samples. The primary problem in an LPC analysis is to find how much weight to attach to each of the previous samples; in other words, to find the predictor coefficients.

The Speech Model

An efficient analysis of speech signals at the waveform, or acoustic, level should take into account the actual method of speech production. Human speech sounds are produced by air being pushed from the lungs through the throat and out the mouth. A distinction is made between two classes of these sounds that is very important to many speech modelling techniques [Rab78]. When the air is forced through the glottis (see Fig. 1.1), it forces the vocal cords, which are under tension, to vibrate. This vibration introduces quasi-periodic pulses of air into the vocal tract, and causes what is known as *voiced* speech. When the vocal cords are not used, and sounds are produced by forcing

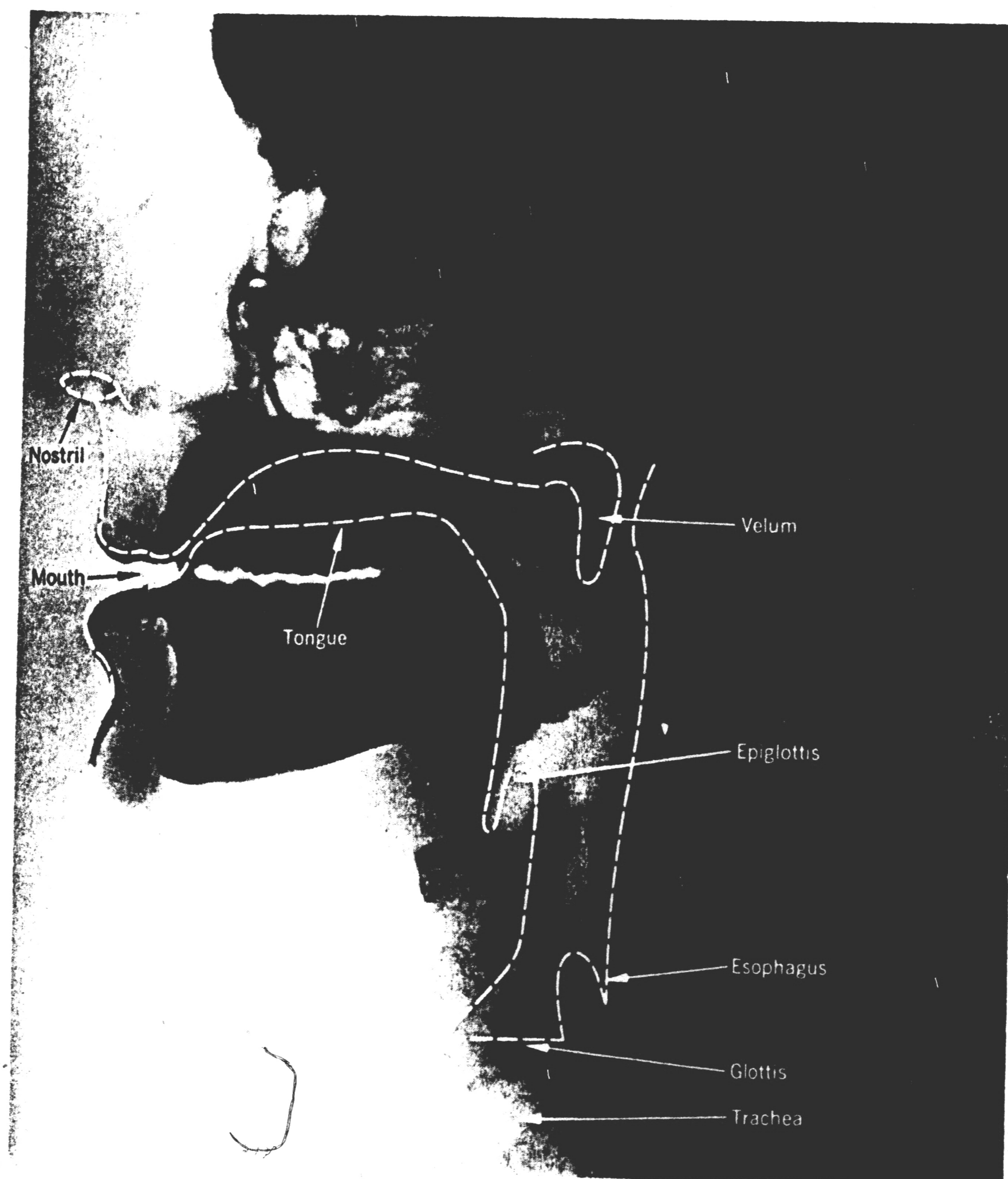


Fig 1.1 X-ray of the human vocal apparatus [Fla70]

air through a constriction at some point in the vocal tract, then these sounds are called *unvoiced*, and are known linguistically as fricatives. Sometimes both methods of sound production are used simultaneously, giving rise to what is known as a voiced fricative (eg. the *z* sound in *zebra*).

The speech production process can be divided into three elements that independently affect the type of sound produced. The first is the production of the glottal pulse, which only occurs during voiced speech, and the second is resonance within the vocal tract, which depends on the shape of the tract at the time of the utterance. This shape can be altered by movement of the tongue, changing the extension of the velum, which couples or uncouples the nasal tract and by moving the teeth and lips. The third element is the manner in which the sound is emitted or radiated, which involves the shape of the mouth and the degree to which the nostrils are used.

Considering the frequency spectrum of this speech sound, each of these elements makes its own contribution to the spectrum. Production of the glottal pulse is modelled as a filter, $G(z)$, operating on an impulse excitation. $G(z)$ is an all-zero filter, thus giving a finite impulse response [Opp75], which is the glottal pulse. The excitation for unvoiced sounds is white noise, which models the air turbulence caused by the constriction in the vocal tract. Resonance in the vocal tract is modelled as a filter, $V(z)$, while the spectral contribution of radiation is modelled as a filter given the transfer function $R(z)$.

Thus a simplified model of speech production can be constructed [O'S87]. Such a model is shown in Fig. 1.2. This model has the flaw that voiced fricatives cannot be accurately represented; some of them are regarded as voiced and some as unvoiced.

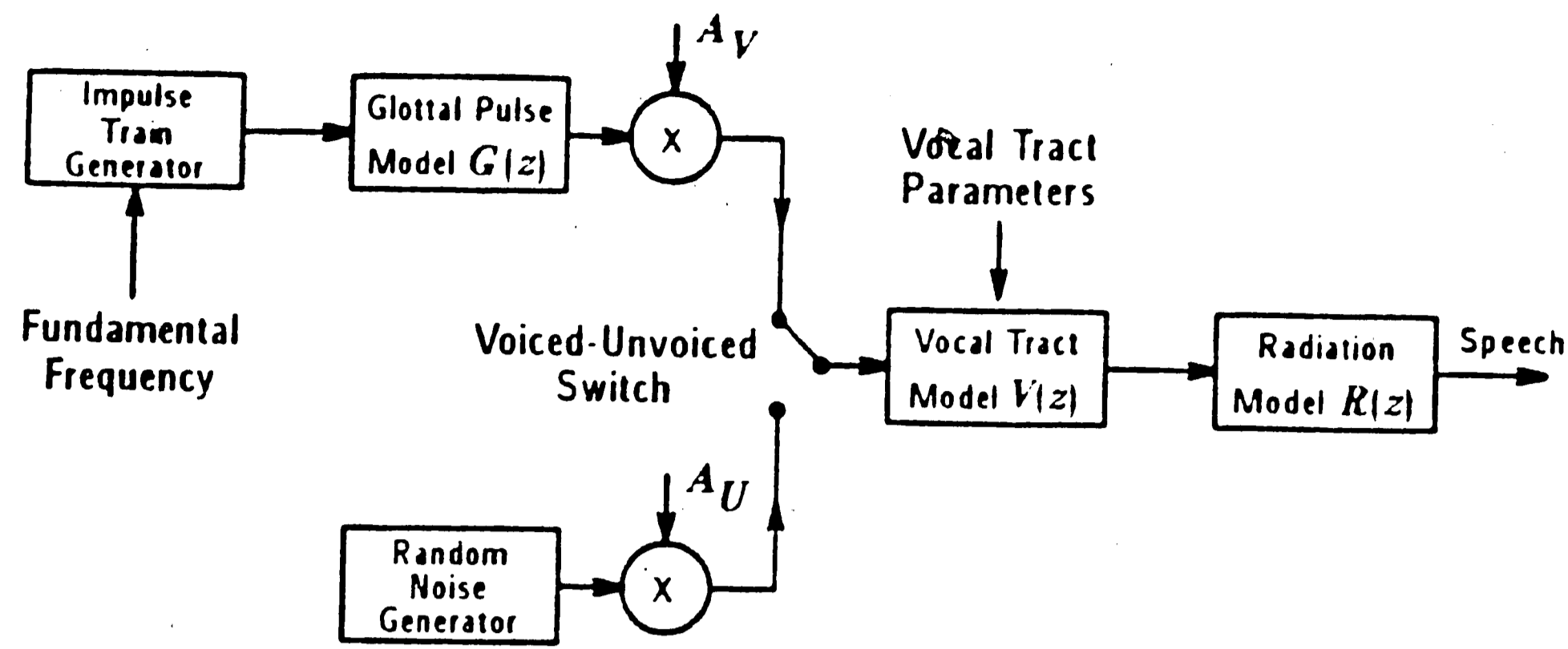


Fig 1.2 Model of speech production [O'Sh87]

For the purposes of Linear Prediction, the model shown in Fig. 1.2 can be simplified by combining the spectral effects of the glottis, the vocal tract and radiation into a single all-purpose filter (see Fig. 1.3). Since all speech analysis methods considered here use sampled signals, and because the parameters of the filter change as the configuration of the vocal apparatus is altered to produce different sounds, this filter is more formally defined as a time-varying digital filter [Rab78]. Since the filter synthesizes an approximation of a speech signal, this filter is called a *synthesis filter* and its transfer function is denoted $H(z)$.

$$H(z) = G(z) V(z) R(z) \quad (1.1)$$

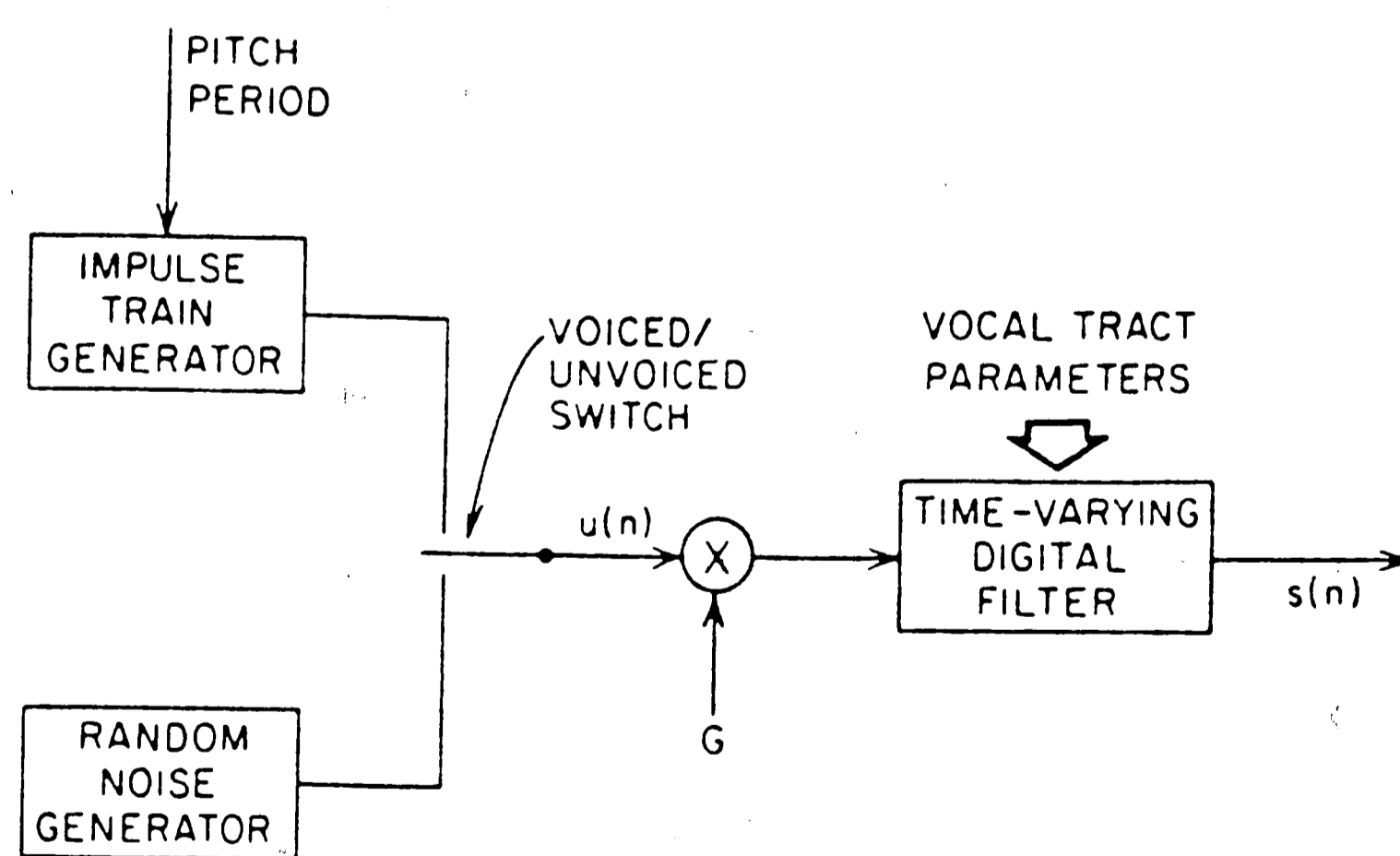


Fig 1.3 Simplified model for speech production [Rab78]

The task of Linear Prediction is now seen to be that of obtaining the coefficients of the filter $H(z)$. These coefficients can then be used to synthesize an approximation of the original signal, $\hat{s}(n)$, by exciting the filter with $u(n)$, which is either an impulse train or a noise source as appropriate.

$$\hat{S}(z) = H(z) U(z) \quad (1.2)$$

To simplify this task, a set of N samples, called a frame, is defined, the length of which is usually of the order of a few cycles of the glottal pulse. The predictor coefficients are assumed to be constant over such a frame; that is, we assume the speech waveform to be *stationary* over this period [Jay84]. $H(z)$ has, in the general case, p poles and q zeros. Allowing that the filter has a gain of G that compensates for the fact that $U(z)$ has a constant variance while the variances of different speech sounds may vary considerably, then:

$$H(z) = \frac{\hat{S}(z)}{U(z)} = G \frac{1 + \sum_{l=1}^q b_l z^{-l}}{1 - \sum_{k=1}^p a_k z^{-k}} \quad (1.3)$$

or, in the time domain:

$$\hat{s}(n) = \sum_{k=1}^p a_k \hat{s}(n-k) + G \left\{ 1 + \sum_{l=1}^q b_l u(n-l) \right\} \quad (1.4)$$

Recall that this filter constructs an approximation of the speech waveform from an excitation function. In order to find the coefficients of the filter (i.e. a_k and b_l) one must pass the speech signal one is trying to model through the inverse of $H(z)$ and find coefficients such that this inverse filter produces a signal as near as possible to the

excitation function, $U(z)$. The numerator G in eqn. 1.5 allows for the difference in variance between the output of the inverse filter and the desired output, $U(z)$. This inverse filter is known as an *analysis filter* and its transfer function is defined as $A(z)$.

$$A(z) = \frac{G}{H(z)} \quad (1.5)$$

When the speech signal is passed through $A(z)$, the resulting signal, $e(n)$, is known as the *residual* or *error* signal, with a z-transform $E(z)$. Thus:

$$E(z) = S(z) A(z) \quad (1.6)$$

and so,

$$S(z) = \frac{E(z) H(z)}{G} \quad (1.7)$$

If one chooses the coefficient values of $A(z)$ such that the residual $e(n)$ has a minimum variance, then this residual will be very similar to an impulse train for voiced speech. The impulses will arise from the fact that the arrival of another glottal pulse cannot be predicted¹ and will thus result in a large error signal at that point. For unvoiced speech, this large error will not occur, and then $e(n)$ will be similar to a small white noise source. For this reason random noise is used as the excitation for unvoiced sounds. In either case, the output of the analysis filter $A(z)$, is a signal that approximates $U(z)$ multiplied by the gain factor G .

Now, since

$$S(z) = \frac{H(z) E(z)}{G} \quad \text{from (1.7)}$$

¹ Attempts to predict speech based on a prediction of the arrival of the next glottal pulse have not been successful.

and, $\hat{S}(z) = H(z) U(z)$ *from (1.2)*

furthermore, $E(z) \approx G \cdot U(z)$

then: $\hat{S}(z) \approx S(z)$ (1.8)

Thus the goal of LPC analysis is now reduced to minimizing the variance of the residual signal when the speech waveform is passed through $A(z)$. The smaller that this variance is, in principle, the closer the synthesized waveform will be to the original one.

Note that the benefit of this procedure is that only the filter coefficients need to be passed from the analysis end to the receiving end - the speech waveform itself does not have to be transmitted. It is usually sufficient to re-adjust these coefficients only every 10msec [Ata85]. This obviously results in a tremendous reduction in the bit rate needed to transmit or store the signal. Study of the coefficients and the residual signal have also led to many other uses for LPC. For example, the pitch period can be found from the residual signal. The LPC coefficients can also be used as an efficient template for waveform matching in a speech recognition system.

Poles and Zeros

Before any attempt can be made to find an algorithm that optimally reduces the variance of the error signal, the form of the filter $H(z)$ must be decided upon. Recall that, in the general case, both poles and zeros are present:

$$H(z) = G \frac{1 + \sum_{l=1}^q b_l z^{-l}}{1 - \sum_{k=1}^p a_k z^{-k}}$$

Virtually all LPC used commercially today uses an all-pole filter to predict the speech signal [O'Sh87]. This is for two reasons: Firstly, the poles (or, more correctly, the filter coefficients that model the poles) can be found efficiently and accurately by any of a number of methods. The most popular of these are the covariance method [Ata71], the autocorrelation formulation [Mar73] and the lattice method [Mak77]. The computation of all-pole filter coefficients is efficient because linear equations can be found which, when solved, determine the optimal predictor coefficients. When zeros are included in the filter, however, the equations to be solved are no longer linear and are thus far more difficult to solve [Miy86]. The second reason for using an all-pole filter is primarily a justification for using the simpler computation method - the effect of the zeroes can be approximated by modifying the poles - a process that is automatically performed in the solution methods currently in use [Ata85].

Consider the representation of some arbitrary transfer function in the z-plane. A pole represents a peak in the magnitude of the transfer function at the point in the z-plane defined as the pole position. This magnitude increases the nearer one gets to the pole. In fact, the pole causes an increase in the magnitude of the transfer function at all points in the z-plane, although this effect is small at large distances from the pole position. A zero, on the other hand produces a local null in the spectrum and it tends to reduce the transfer function magnitude at all other points. When a number of poles and/or zeros occur simultaneously in the z-plane, they will each have two effects. Firstly they will

cause local peaks and nulls near their z-plane positions and secondly their effects at other points will interplay to bring about what is known as the *spectral balance*.

Now, if the LPC filter $H(z)$ is an all-pole filter, then it will only mark the positions of the poles in the z-plane. Modification of these poles can effectively account for the change in the spectral balance caused by the missing zeros [Ata85], but $H(z)$ will still fail to represent the local dips in the spectrum. Atal maintains that, perceptually, the overall spectral balance is the more important effect [Ata78].

There are two primary reasons why nulls occur in the natural speech spectrum. These nulls give rise to zeros in the transfer function of the filter representing speech production. The driving glottal pulse contains many nulls in its spectrum, and whenever the vocal tract contains two or more paths through which air can flow, its transfer function, $V(z)$, will contain more zeros. This splitting of the vocal tract occurs most obviously in nasal sounds, but can result from a fricative constriction or by the tongue position. Zeros may also result from the radiation load at the lips and from the low-pass filtering performed before sampling to ensure compliance with the Nyquist criterion [Ata78]. Since,

$$H(z) = G(z) V(z) R(z) \quad \text{from (1.1)}$$

any zeros in any one of the component transfer functions will result in zeros in $H(z)$.

Some attempts have been made to incorporate zeros into the LPC filter. The most efficient of these performs a two-part analysis [Fri83]. Firstly an all-pole analysis is performed giving rise to a residual signal $E(z)$. Since,

$$E(z) = \frac{S(z) \cdot G}{H(z)} \quad \text{from (1.7)}$$

the spectrum of this residual will contain the nulls that were present in $S(z)$ and could not be represented by $H(z)$. In Friedlander's procedure, the spectrum of the residual is inverted so that the nulls become peaks and then this *spectrally inverted* residual is analyzed once more, again with an all-pole filter. This second analysis finds the position of the peaks in the spectrally inverted residual, which correspond to the nulls of the original residual and thus the nulls of $S(z)$. The procedure is inaccurate, however, in that when the first analysis is done, the pole positions are modified to account for the spectral balance effects of the zeros, and thus when the zeros are actually located, the poles are now in the wrong position. An optimal solution for the coefficients of $H(z)$ must, therefore, solve for the locations of the poles and zeros simultaneously.

The purpose of this study is to demonstrate the effect that zeros have on linear prediction. By monitoring the improvement that the introduction of zeros brings about in LPC performance, it will allow determination of whether there is sufficient gain in performance to warrant continued effort toward finding an efficient means to simultaneously calculate the positions of both poles and zeros.

The method used to find these zeros is not an explicit solution of the equations from which the coefficients can be found. As stated previously, these equations are not linear for the general case in which both poles and zeros are present in $H(z)$. Instead, it attempts to minimize the predictor error by adjusting the coefficients until the best combination is found. This adjustment is made based on the gradient of the residual

variance against the coefficient values, a procedure known as the method of steepest decent [Jay84]. The advantage of this method is that it can be used to find the coefficients for any combination of poles and zeros but, being a numerical method, it will not necessarily always find the optimum coefficients. Thus the most valid comparisons that can be made in this study are between the all-pole results using this method of steepest descent and the pole-zero combinations also using this method. It is not expected that either will perform as well as the explicit solution methods mentioned earlier. The aim of this study is, however, to quantify, analyze and describe the effects that zeros have in LPC, not to present a better method of obtaining the coefficients.

Performance Measures

A major factor in any comparison of systems is, of course, the measure by which the system performances are gauged. There are many methods one could use to measure how well an LPC filter is performing; the best measure depends on the particular application for which the LPC is being used. This investigation concentrates on transmission/storage aspects of LPC and makes no attempt to decide whether the inclusion of zeros assists in the use of LPC for speech recognition or speaker identification.

One measure that is commonly used to determine LPC performance is to calculate the variance of the residual signal. This is particularly appropriate when the residual signal is to be sent along with the LPC coefficients. When the entire residual is quantized and transmitted, this is known as *differential predictive pulse code modulation* or DPCM [Jay84] and if only the perceptually important parts of the residual are sent it is called a *residually excited linear prediction* or RELP [O'Sh87]. In the case of DPCM, the only

errors introduced into the system are quantization errors and if adaptive quantization is used, then the reduction in residual variance is a direct measure of the improvement in the *Signal to Noise Ratio* (SNR), since the quantization noise introduced is directly proportional to the variance of the quantized signal [Rab78].

However, in many of the extremely low bit-rate applications that LPC is used for, the residual signal is not sent at all, and the filter is excited at the receiving end by either an impulse train (for voiced speech) or a noise source for unvoiced sounds. In such cases, a system giving rise to a residual having the smallest variance could be out-performed by one in which the residual power is concentrated in the 'impulses' and is small elsewhere. In this type of application, a better measure of the system performance is a comparison of the reconstructed signal with the original one. This is done most effectively in the frequency domain since the phase insensitive human ear detects only the magnitude of a sound's spectrum, and not its time domain waveform. Both these performance measures are used in this work.

Chapter 2

LPC Filter Configuration

Recall that an LPC system consists of two filters - the analysis filter that is used to determine the coefficients of the transfer function $H(z)$, and the synthesis filter that reconstructs the speech signal at the receiving end.

The All-Pole Configuration

A standard configuration for these filters exists for the case of the all-pole filters (see Fig. 2.1 and Fig. 2.3). Note that since the analysis and synthesis filters are inverses of each other, only one, the synthesis filter, is actually an all-pole filter. The analysis filter contains an all-pole filter in its feedback path but is not, itself, an all-pole filter. The transfer function of the "all-pole" analysis filter, it is found, does contain zeros, but these zeros are dependent on the pole positions and the filter is still known in the engineering literature as an all-pole filter [Koo86]

The transfer function of an all-pole synthesis filter is:

$$H(z)_{\text{all-pole}} = \frac{\hat{S}(z)}{U(z)} = \frac{G}{1 - \sum_{k=1}^p a_k z^{-k}} \quad (2.1)$$

which converts to the time-domain form:

$$\hat{s}(n) = \sum_{k=1}^p a_k \hat{s}(n-k) + G \cdot u(n) \quad (2.2)$$

This can be constructed in terms of hardware building blocks as follows:

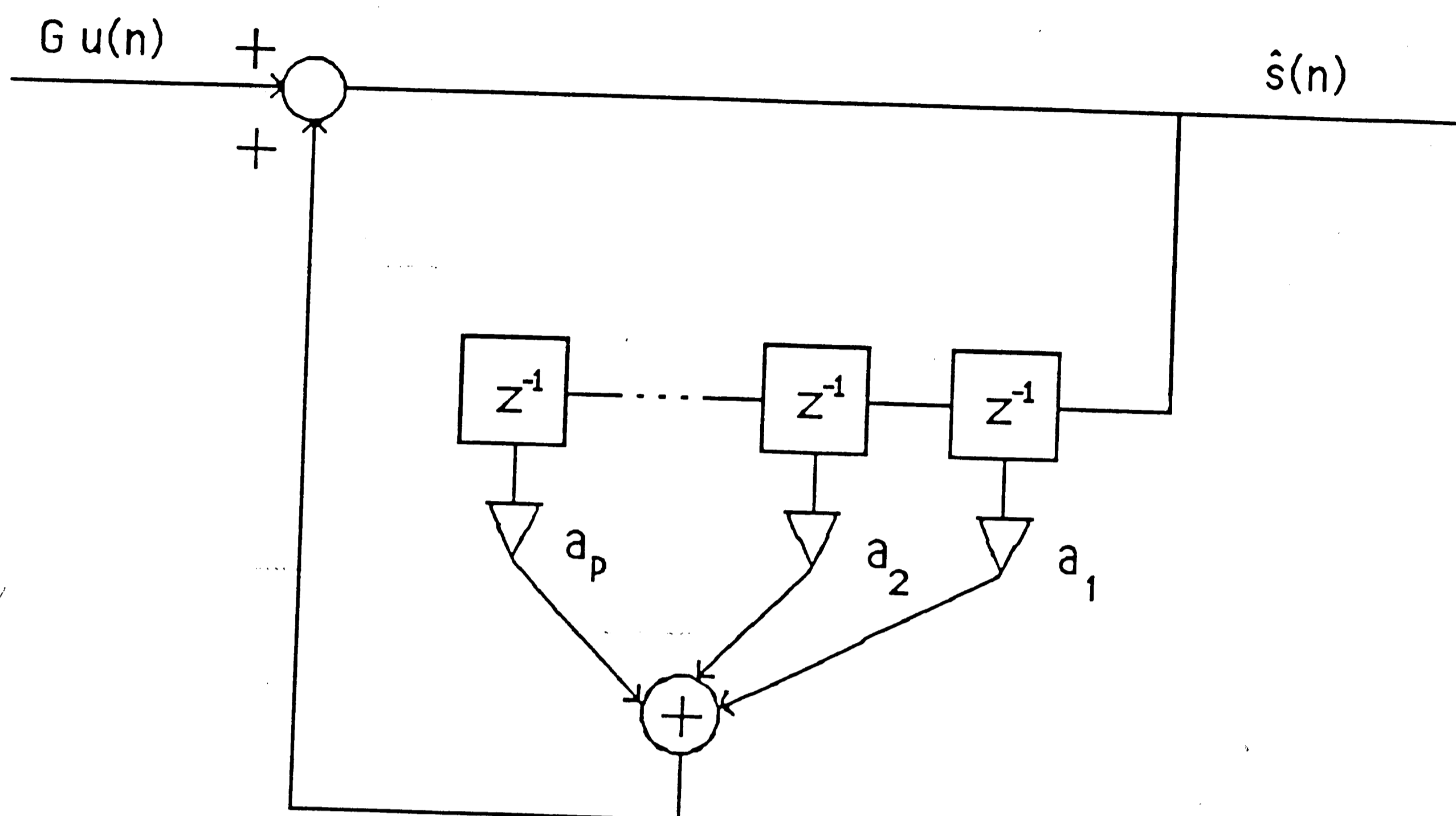


Fig. 2.1 Block diagram of an all-pole synthesis filter

The all-pole analysis filter must be, as discussed earlier, the inverse of $H(z)$. Thus:

$$A(z)_{\text{all-pole}} = \frac{E(z)}{S(z)} = 1 - \sum_{k=1}^p a_k z^{-k} \quad (2.3)$$

and so, the output $e(n)$ as function of time is:

$$e(n) = s(n) - \sum_{k=1}^p a_k s(n-k) \quad (2.4)$$

Defining a quantity $\tilde{s}(n)$, which assists in relating the equations to the diagram.

$$\tilde{s}(n) = \sum_{k=1}^p a_k s(n-k) \quad (2.5)$$

then:

$$e(n) = s(n) - \tilde{s}(n) \quad (2.6)$$

Such a filter could be constructed as follows:

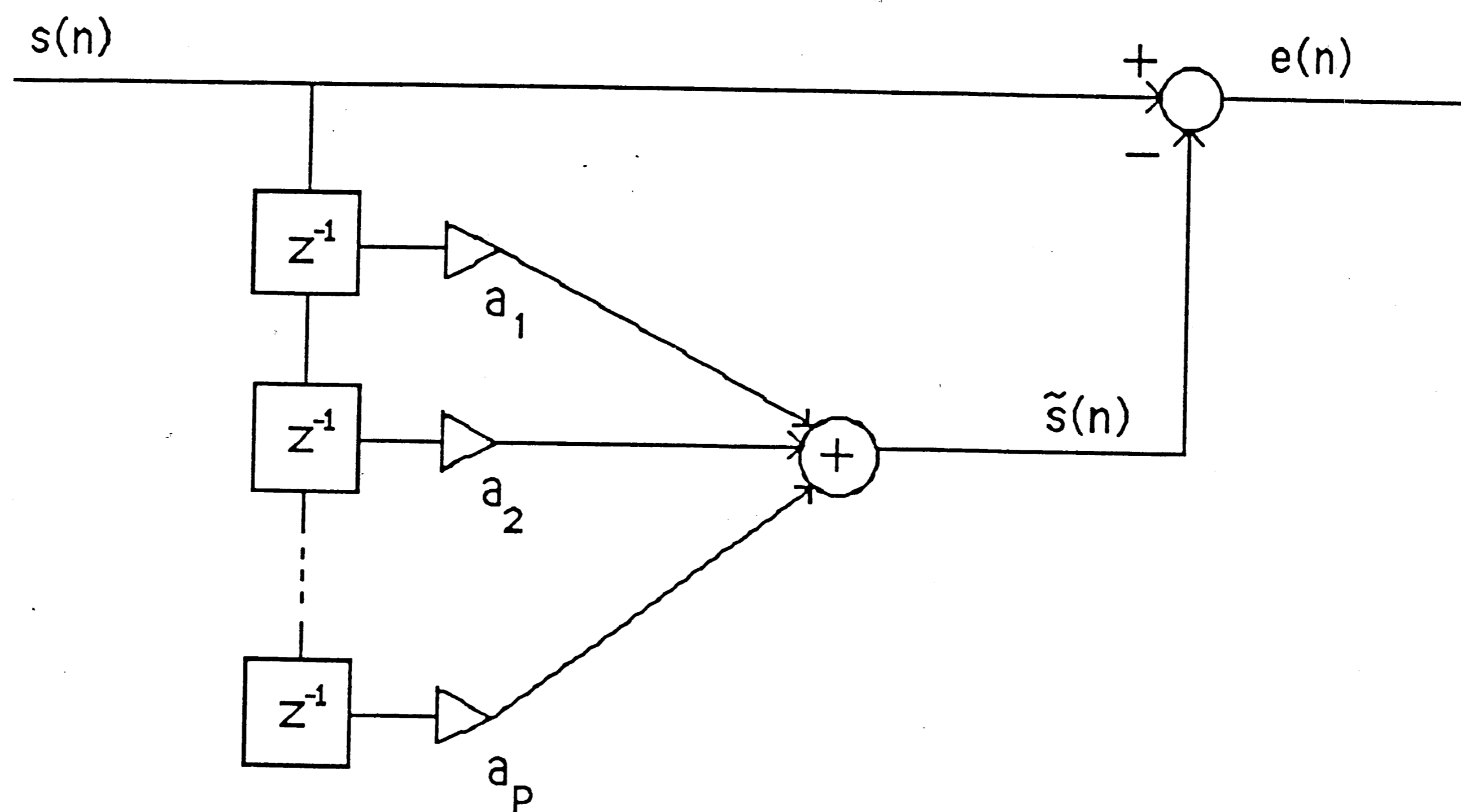


Fig. 2.2 Possible block diagram of an all-pole analysis filter

In practice, however, it turns out that a problem exists with this configuration caused by the fact that in a digital system (as opposed to the *discrete* systems that are a result of sampling) the waveform must, at some point, be quantized into digital strings of numbers so that they can be mathematically manipulated. If the system contained only real-time delay units and analog methods for multiplication and addition, then Fig. 2.2 would work correctly. Considering a quantized system, however, it is desirable that the analysis filter operates on quantized values of $s(n)$ so that the analysis and synthesis filter are operating on the same data.

The configuration shown in Fig. 2.3 of the all-pole analysis filter is adapted to minimize

the effects of quantization. Note the following relationships that occur in Fig 2.3.

$$e(n) = s(n) - \tilde{s}(n) \quad (2.7)$$

$$s'(n) = \tilde{s}(n) + e(n) \quad (2.8)$$

and so

$$s'(n) = \tilde{s}(n) + s(n) - \tilde{s}(n) = s(n) \quad (2.9)$$

Thus the relationship described in eqn 2.5 still holds even though $H(z)$ now operates on $s'(n)$ instead of $s(n)$.

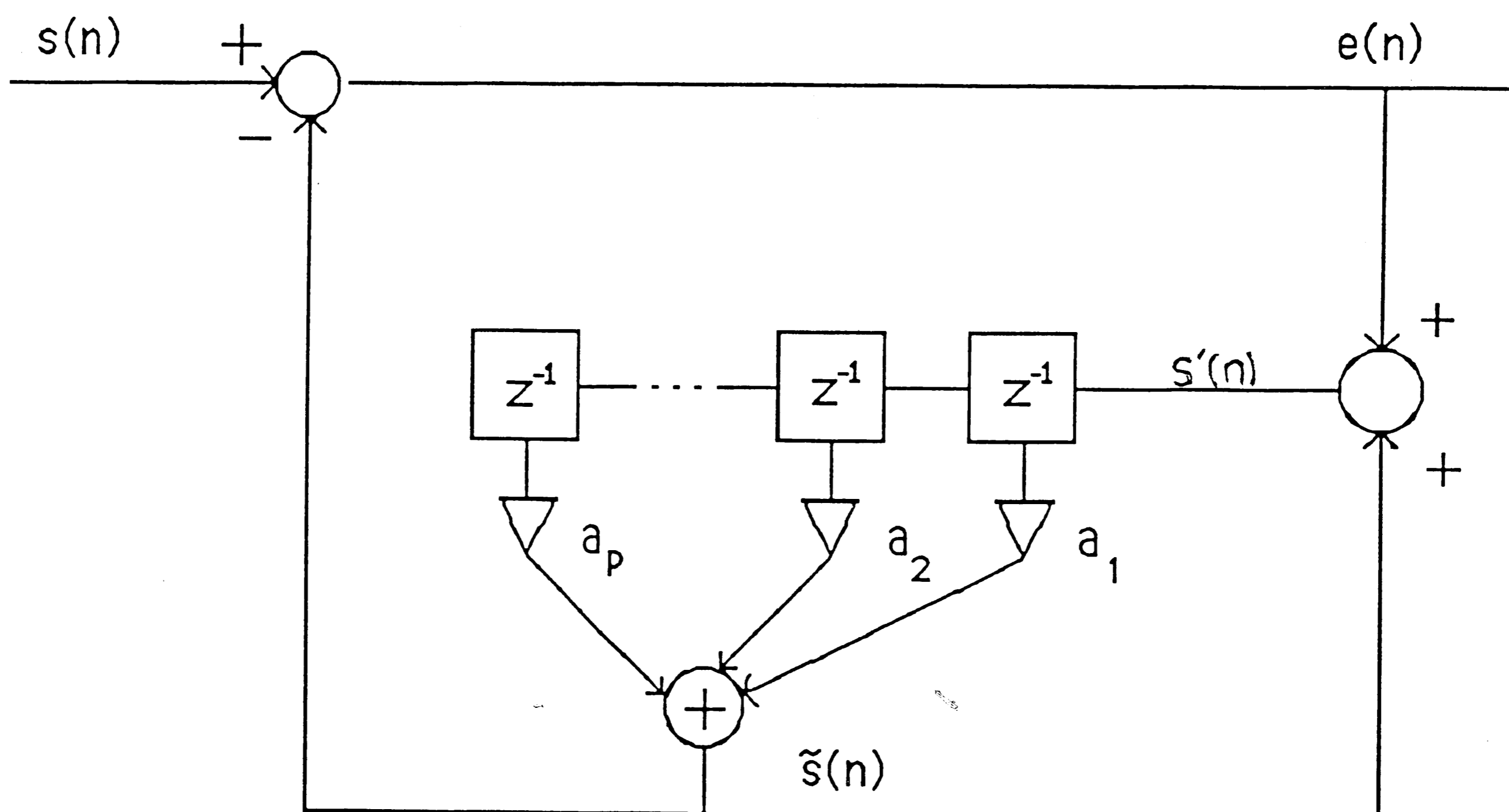


Fig. 2.3 Block diagram of an all-pole analysis filter

Now, when the quantization is shown, the analysis filter can be viewed as in Fig. 2.4. The effect of quantization is modelled as a source introducing a quantization noise signal, $q(n)$. Thus the residual signal becomes $e_q(n)$ where:

$$e_q(n) = e(n) + q(n) \quad (2.10)$$

and

$$e(n) = s(n) - \tilde{s}(n) \quad \text{from (2.4)}$$

Therefore:
$$e_q(n) = s(n) - \tilde{s}(n) + q(n) \quad (2.11)$$

$s_q(n)$ is now defined as the sum of $e_q(n)$ and $\tilde{s}(n)$.

$$s_q(n) = e_q(n) + \tilde{s}(n) \quad (2.12)$$

so:
$$s_q(n) = s(n) - \tilde{s}(n) + q(n) - \tilde{s}(n) \quad (2.13)$$

or
$$s_q(n) = s(n) + q(n) \quad (2.14)$$

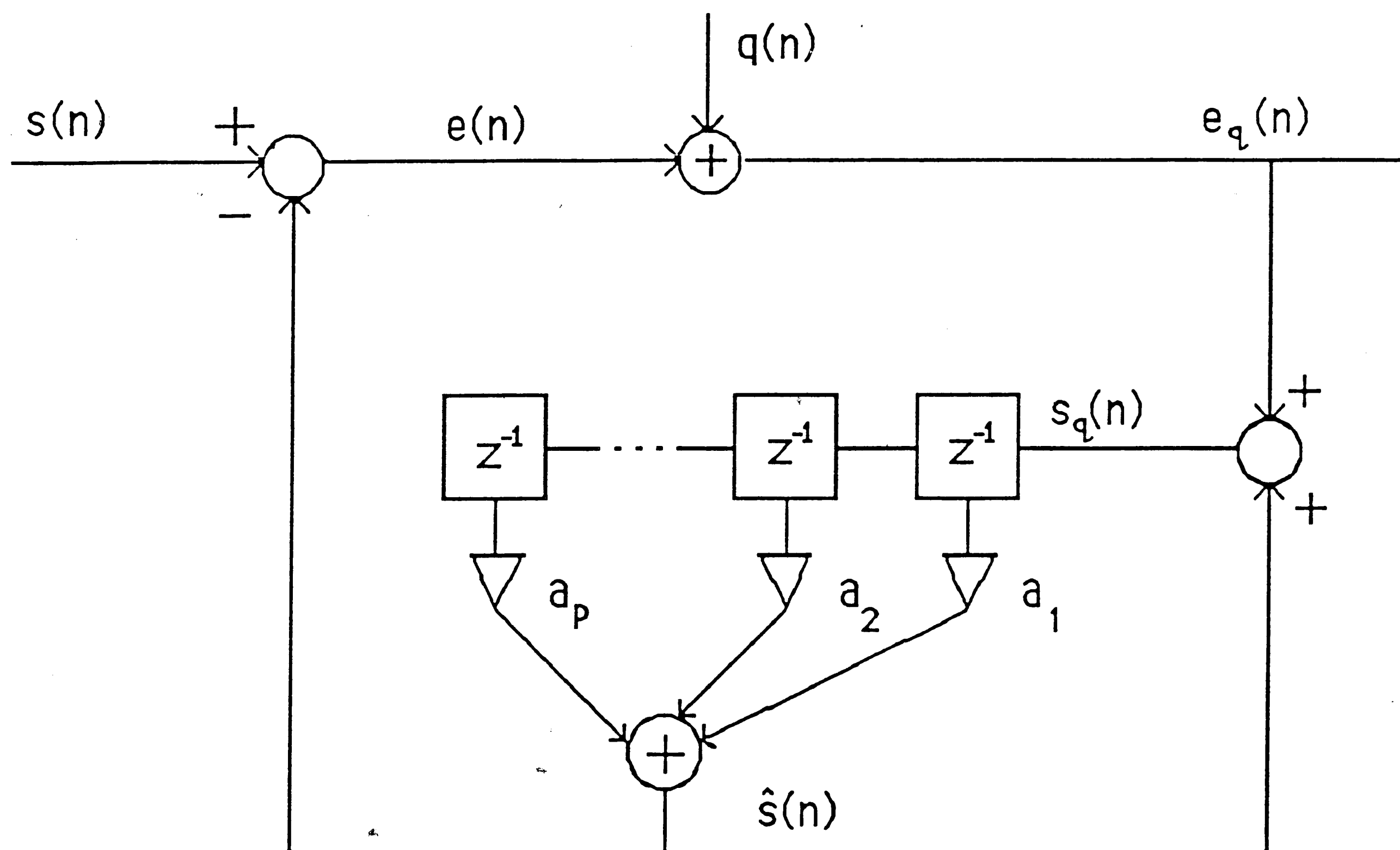


Fig. 2.4 Block diagram of all-pole analysis filter showing the effect of quantization noise

Thus the filter $H(z)$, operating on the quantized version of $e(n)$, can construct a replica of $s(n)$ that is different from the original $s(n)$ only by the addition of $q(n)$. This is important since at the receiving end, only the quantised version of $e(n)$ would be available, even in an ADPCM system.

Although this study takes no further account of quantization, it does recognize that this problem occurs in implementation and thus the analysis filters are designed to account for the noise introduced by quantization error.

The Pole-Zero Configuration

The configurations used to build synthesis and analysis filters for the case where $H(z)$ contains only poles can be extended to the situation in which $H(z)$ contains both poles and zeros.

The transfer function of the synthesis filter is now:

$$H(z) = \frac{\hat{S}(z)}{U(z)} = G \frac{1 + \sum_{l=1}^q b_l z^{-l}}{1 - \sum_{k=1}^p a_k z^{-k}} \quad (2.15)$$

and so the time-domain representation of the relationship of the output to the input is:

$$\hat{s}(n) = \sum_{k=1}^p a_k \hat{s}(n-k) + G \left\{ u(n) + \sum_{l=1}^q b_l u(n-l) \right\} \quad (2.16)$$

It can be seen that the configuration in Fig 2.5 gives rise to such an output $\hat{s}(n)$, given the input $u(n)$.

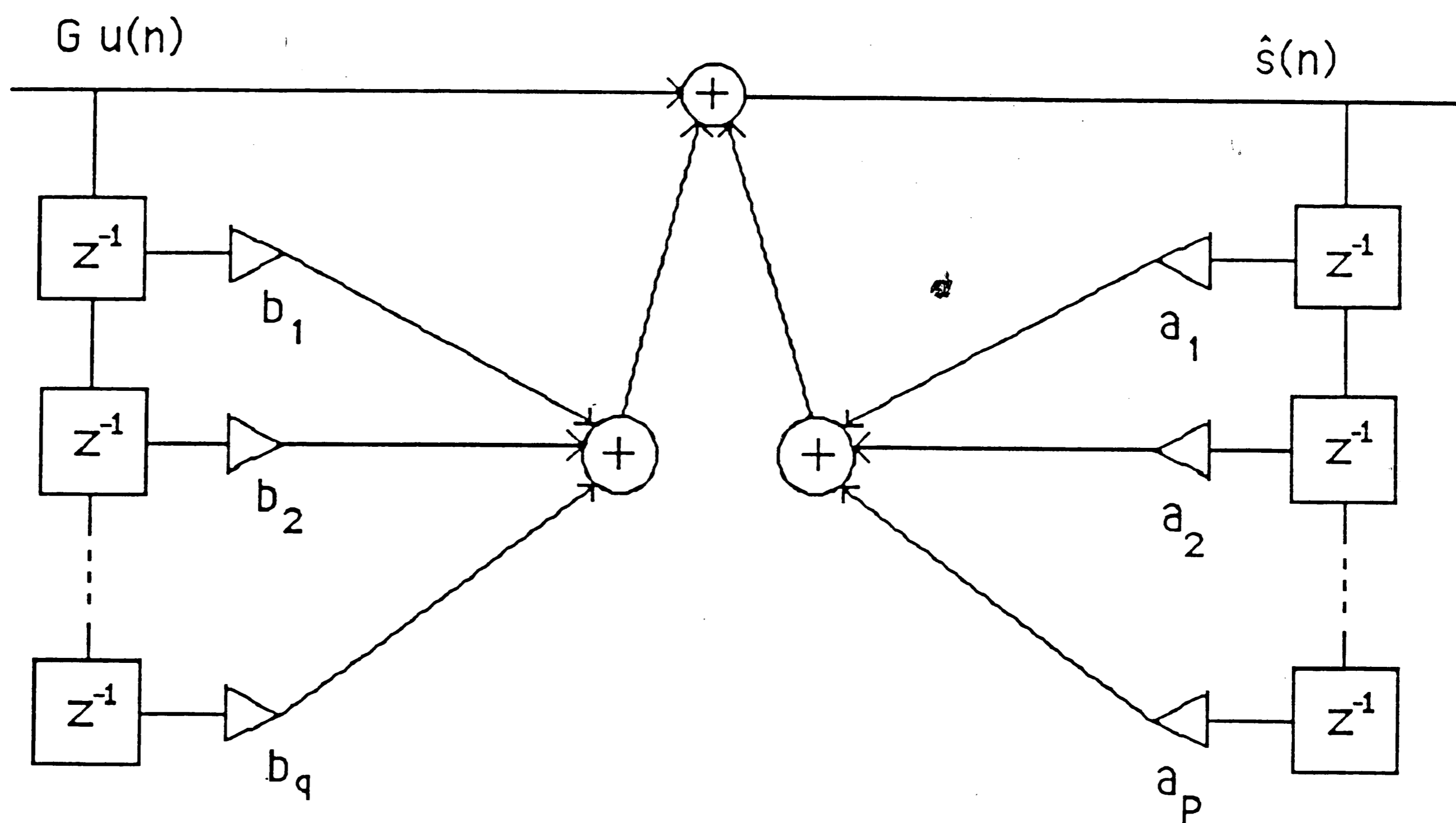


Fig 2.5 Block diagram of a pole-zero synthesis filter

In order to find a configuration for the analysis filter, the transfer function $A(z)$ must be found, where:

$$A(z) = \frac{G}{H(z)} \quad \text{from (1.5)}$$

Therefore, with $H(z)$ defined as in eqn. 2.15, we have:

$$A(z) = \frac{E(z)}{S(z)} = \frac{1 - \sum_{k=1}^p a_k z^{-k}}{1 + \sum_{l=1}^q b_l z^{-l}} \quad (2.17)$$

In order to design a system with such a transfer function the output and input should be related in the time-domain:

$$e(n) = s(n) - \sum_{k=1}^p a_k s(n-k) - \sum_{l=1}^q b_l e(n-l) \quad (2.18)$$

Now it can be seen that the system proposed in Fig. 2.6 has a relationship between its input and output as described in eqn 2.18. This filter has a design based on the modification included in Fig. 2.3 so that it, too, will work correctly in the presence of quantization noise.

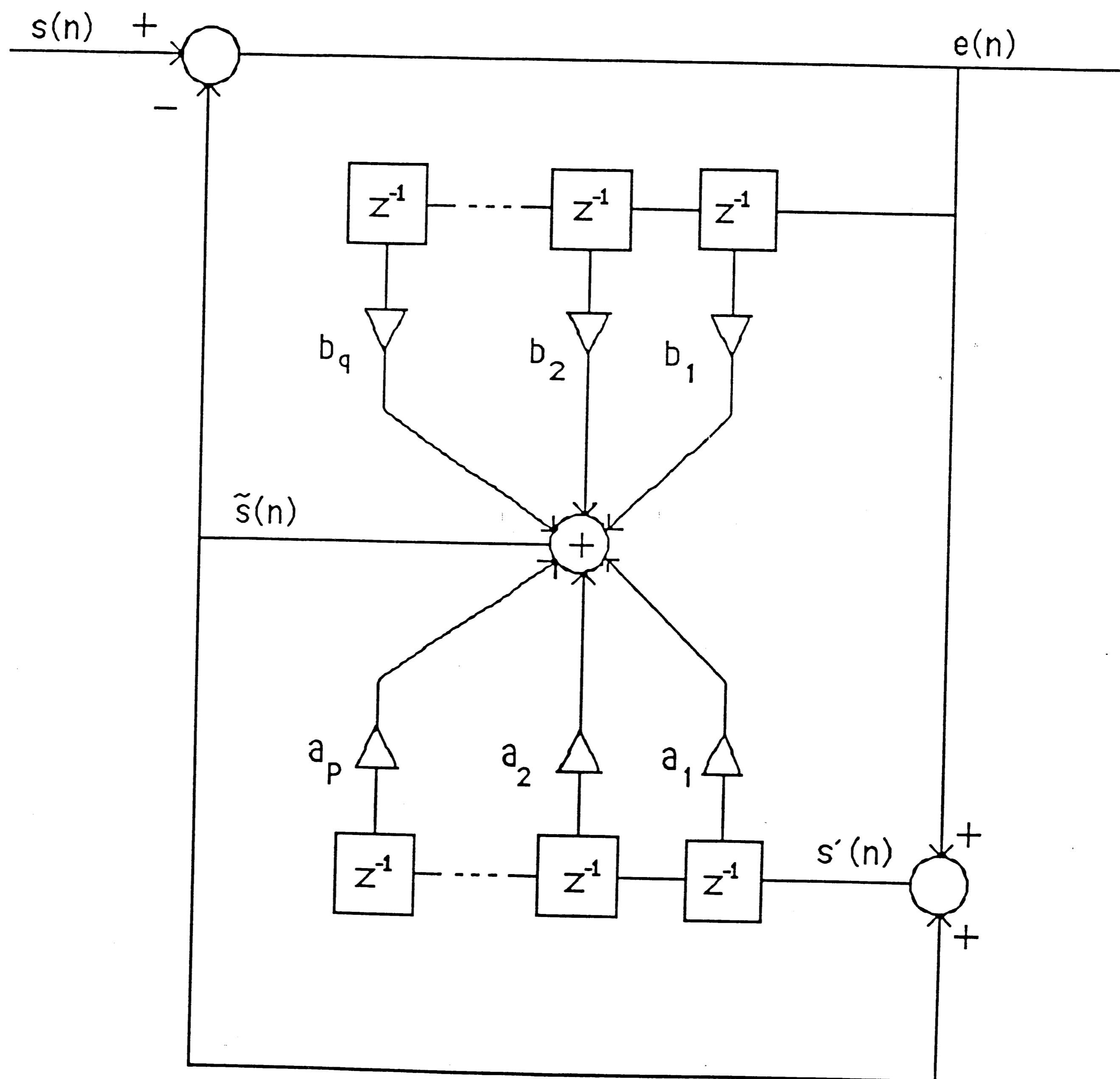


Fig. 2.6 Block diagram of a pole-zero analysis filter

It was shown in Chapter 1 that the procedure one follows to obtain the optimal filter

coefficients is to minimize $e(n)$. A procedure to do this is presented in Chapter 3. The residual signal that is to be minimized is the output of the pole-zero analysis filter. Thus, from eqn 2.18.

$$e(n) = s(n) - \sum_{k=1}^p a_k s(n-k) - \sum_{l=1}^q b_l e(n-l)$$

is the function that must be minimized.

Chapter 3

The Method of Steepest Descent

The method of steepest descent, first proposed by Cauchy in 1847, is a means to solve for the minimization of any function, including functions of many variables. It does this by finding the gradient of the function with respect to each of the variables and then adjusting that variable proportionally to the *negative* of the gradient (see Fig. 3.1). As long as the function decreases monotonically towards its minimum, this method will converge on the absolute minimum of the function; if not, it may be 'trapped' in a local minimum [Wil67].

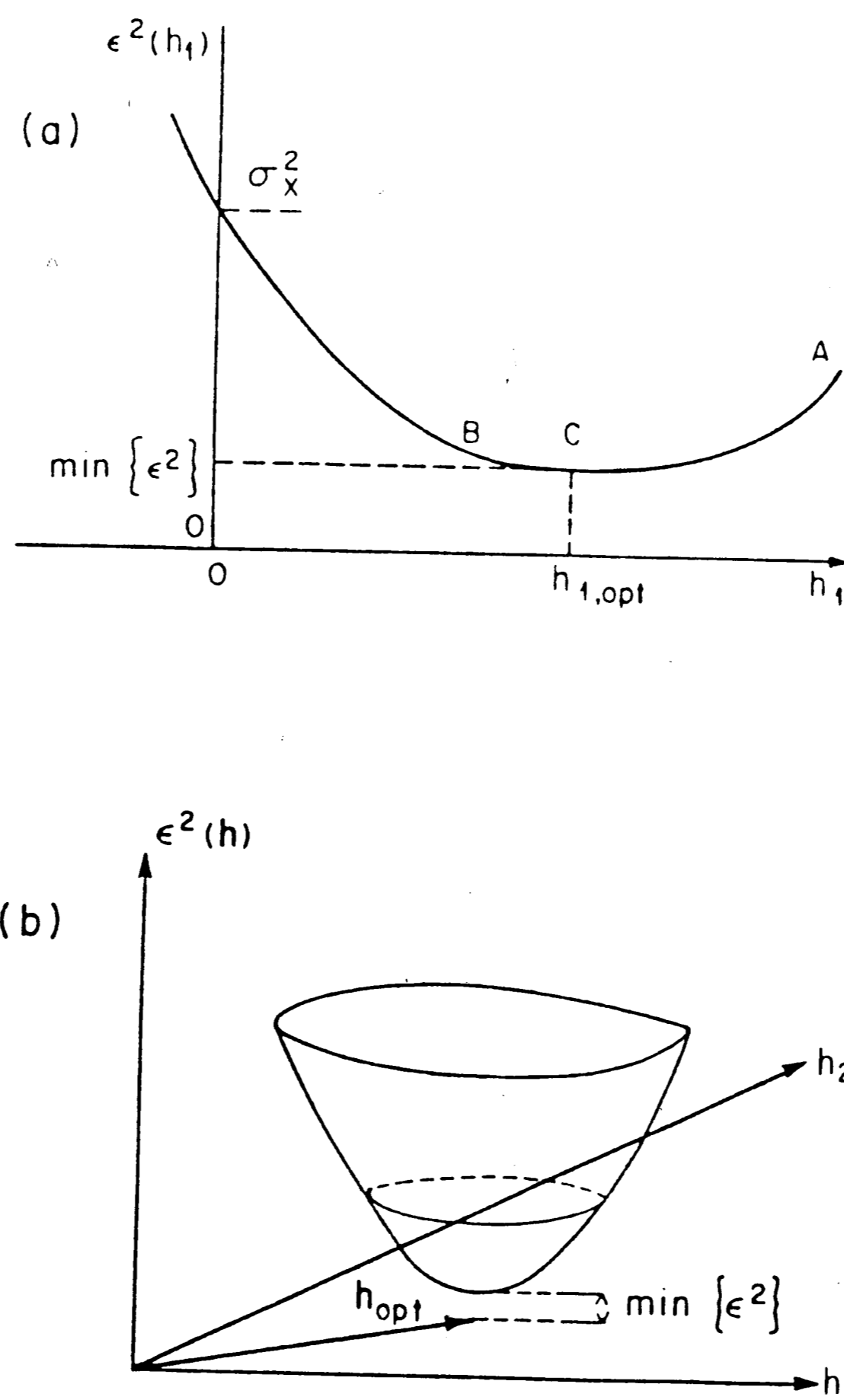


Fig 3.1 Minimization by the method of steepest descent [Jay84]

Consider some function.

$$y = f(x_1, x_2, \dots, x_i, \dots, x_n) \quad (3.1)$$

where $(x_1 \dots x_n)$ are independent variables.

The gradient of this function, ∇ , is defined as a vector in n-dimensional space, where

$$\nabla = \left(\frac{\partial y}{\partial(x_1)}, \frac{\partial y}{\partial(x_2)}, \dots, \frac{\partial y}{\partial(x_n)} \right) \quad (3.2)$$

It has been shown [Kre83] that this gradient vector has the direction of the steepest descent of the function y . Thus each variable, x_i , should be adjusted in the direction of $-\frac{\partial y}{\partial(x_i)}$

Thus, in general, the adjustment that must be made to the variable x_i when trying to find the minimum of the function is:

$$x_i^{(j+1)} = x_i^{(j)} - K \cdot \left[\frac{\partial y}{\partial x_i} \right]^{(j)} \quad (3.3)$$

where: $x^{(j)}$ is the value of the variable x after j adjustments

and: K is a proportionality constant

Each time that this adjustment is made, the function will move closer to its minimum. Steepest descent is thus an iterative method in which a compromise must be made between computation time and the desired accuracy.

An Example of the Method of Steepest Descent

To see how this works in a simple case, consider the function:

$$y(x) = (x - 3)^2$$

Now, finding the gradient with respect to x :

$$\frac{\partial y}{\partial x} = 2(x - 3)$$

Picking some initial value of the variable from which to start the analysis:

$$x^{(0)} = 0$$

then:
$$\left[\frac{\partial y}{\partial x} \right]^{(0)} = -6$$

and so x must be changed:
$$x^{(1)} = x^{(0)} - \{ K \cdot (-6) \}$$

where K is, say 0.4.

then:
$$x^{(1)} = 0 - \{ 0.2 \cdot (-6) \} = 2.4$$

now,
$$\left[\frac{\partial y}{\partial x} \right]^{(1)} = 2 \cdot (2.4 - 3) = -1.2$$

and so:
$$x^{(2)} = x^{(1)} - \{ K \cdot (-1.2) \} = 2.88$$

and
$$x^{(3)} = x^{(2)} - \{ K \cdot (-0.24) \} = 2.976$$

In this way, it can be seen, the variable x will approach the value, 3 at which point $y(x)$ is at a minimum.

It should be noted that in this simple example, two assumptions were made which will be explored later in greater detail. These were the initial value of the variable and the value of the proportionality constant K . The same assumptions must be made in the more complex application for which the method of steepest descent is used in this study - the minimization of the residual variance.

The Application of Steepest Descent to Linear Predictive Coding

In Chapter 2 it was shown that in a pole-zero analysis filter, the residual signal can be expressed as:

$$e(n) = s(n) - \sum_{k=1}^p a_k s(n-k) - \sum_{l=1}^q b_l e(n-l) \quad (3.4)$$

The discussion in Chapter 1 showed that to obtain the best results from linear prediction, the variance of this residual must be minimised.

Defining the variance of the residual, σ^2 , as the expected value of the square of $e(n)$.

$$\sigma^2 = E [e^2(n)] \quad (3.5)$$

then,
$$\sigma^2 = E [e^2(n)] = E \left[\left(s(n) - \sum_{k=1}^p a_k s(n-k) - \sum_{l=1}^q b_l e(n-l) \right)^2 \right] \quad (3.6)$$

This is a function that has $(p + q)$ variables. To solve the differentiation, the use of the following general calculus result is required:

$$\frac{\partial}{\partial(x)} (f(x))^n = n \cdot (f(x))^{(n-1)} \cdot \frac{\partial}{\partial(x)} (f(x)) \quad (3.7)$$

so, for the case of $n=2$:

$$\frac{\partial}{\partial(x)} (f(x))^2 = 2 \cdot f(x) \cdot \frac{\partial}{\partial(x)} (f(x)) \quad (3.8)$$

Therefore:

$$\frac{\partial(\sigma^2)}{\partial(a_k)} = 2 \cdot E \left[\left(s(n) - \sum_{k=1}^p a_k s(n-k) - \sum_{l=1}^q b_l e(n-l) \right) \cdot \frac{\partial}{\partial(a_k)} \cdot \left(s(n) - \sum_{k=1}^p a_k s(n-k) - \sum_{l=1}^q b_l e(n-l) \right) \right] \quad (3.9)$$

Since this is a solution for the partial differential, all other variables are considered to be constant for this differentiation.

$$\frac{\partial(\sigma^2)}{\partial(a_k)} = 2 \cdot E \left[\left(s(n) - \sum_{k=1}^p a_k s(n-k) - \sum_{l=1}^q b_l e(n-l) \right) \cdot \frac{\partial}{\partial(a_k)} \cdot (a_k s(n-k)) \right] \quad (3.10)$$

Noting that the expression in the large round brackets is simply $e(n)$ from eqn 3.4 and solving the partial differential on the right hand side of the equation:

$$\frac{\partial(\sigma^2)}{\partial(a_k)} = 2 \cdot E [e(n) \cdot s(n-k)] \quad (3.11)$$

A similar result can be obtained for the coefficients modelling the zeros:

$$\frac{\partial(\sigma^2)}{\partial(b_l)} = 2 \cdot E \left[\left(s(n) - \sum_{k=1}^p a_k s(n-k) - \sum_{l=1}^q b_l e(n-l) \right) \cdot \frac{\partial}{\partial(b_l)} \cdot \left(s(n) - \sum_{k=1}^p a_k s(n-k) - \sum_{l=1}^q b_l e(n-l) \right) \right]$$

(3.12)

giving,

$$\frac{\partial(\sigma^2)}{\partial(b_l)} = 2 \cdot E [e(n) \cdot e(n-1)] \quad (3.13)$$

The residual variance can now be minimized by adjusting each variable as described in eqn 3.3.

$$a_k^{(j+1)} = a_k^{(j)} - K_a \cdot \left[\frac{\delta(\sigma^2)}{\delta a_k} \right]^{(j)} \quad (3.14)$$

Therefore:

$$a_k^{(j+1)} = a_k^{(j)} - K_a \cdot E [e(n) \cdot s(n-k)] \quad (3.15)$$

and:

$$b_l^{(j+1)} = b_l^{(j)} - K_b \cdot \left[\frac{\delta(\sigma^2)}{\delta b_l} \right]^{(j)} \quad (3.16)$$

so:

$$b_l^{(j+1)} = b_l^{(j)} - K_b \cdot E [e(n) \cdot e(n-1)] \quad (3.17)$$

A simple method of calculating the quantities indicated in eqn 3.15, for the all-pole case, has been proposed by [Wid76] in which the difference signal used for minimization is the *instantaneous* difference, as opposed to the expected value used in the derivation above. This gives the simplification that

$$a_k^{(j+1)} = a_k^{(j)} - K_a \cdot e(n) \cdot s(n-k) \quad (3.18)$$

and this can be readily applied to the zero-modelling coefficients, giving

$$b_l^{(j+1)} = b_l^{(j)} - K_b \cdot e(n) \cdot e(n-1) \quad (3.19)$$

As in the steepest descent example worked through above, when the adaption equations 3.18 and 3.19 are put into practice, assumptions must be made regarding the initial values of the coefficients and the value of K .

The initial values of the variables affects the performance of the steepest descent procedure in two ways. Most importantly, if a local minimum occurs between the 'initial guess' and the optimal value, then the procedure may be 'trapped' in this local minimum. Clearly, if the initial guess is close to the optimal value, then there is less chance that a local minimum will occur between the two. Secondly, the iterative procedure will converge on the optimal value more quickly if the initial guess is good. Thus, for a set number of iterations, the final value obtained will be closer to the optimal value.

At the start of some particular frame, a good initial guess for the coefficient value would be the final value found at the end of the previous frame since the optimal coefficient values are known to be vary slowly with time. In fact, many systems do not even recalculate the coefficients for each frame, adjusting the coefficients only every 10ms or so [Ata85]. This is because, these coefficients model the shape of the vocal tract - a physical system that can only vary slowly. A better guess for the initial value might even be some linear combination of the previous values of that coefficient!

In this study, however, the initial value of each coefficient at the start of each frame is

chosen to be zero. This increases the simplicity of the study considerably since a single frame can be analysed in depth independently of the surrounding frames. Since the method of steepest descent is not presented as an optimal solution for the coefficient values, but rather as the means by which the effect of the zeros can be studied, it is felt that this simplification is justified. The value of zero was selected since the coefficient values can be either positive or negative.

The value selected for the proportionality constant K also has important effects on the performance of the steepest descent method. If K is set at a very small value, then more iterations will be needed to eventually reach the optimal value. Again, for a set number of iterations, this means that the final value reached will not be as close to the optimal value as it could be. If K is chosen to be too large, however, then the method of steepest descent may become unstable. This is because K relates the adjustment in the variable value to the gradient. If K is too large then the variable may be adjusted, not just beyond the optimal value (in which case it would be adjusted back in the next iteration) but so far beyond the optimal value that it is further away than it was before the adjustment. Now, in the next iteration, the coefficient will be adjusted by an even larger amount (assuming, not unreasonably, that the gradient is larger further from optimal value), back over the optimal value and even further away. In this situation, the method of steepest descent diverges from the optimal value rather than converging towards it. The choice of K may be compared to that facing a golfer making a stroke near the green. If he swings too softly then he will need many strokes to reach the green. If he uses too much power, on the other hand, he will overshoot the green, possibly ending up further away from the hole than when he started.

It should be noted that the instability caused by a proportionality constant that is too large, is entirely different from the filter instability that results from encountering poles that are outside the unit circle in the z-plane.

This value for K is simply determined experimentally. Certain constraints are placed on it, however, so that in this study it is not really a constant but rather a proportionality value that is re-determined for each frame and even for each iteration.

The most important adjustment of K concerns a compensation for the variance of the input signal. Note that both the equation for adjusting a_k and for b_l (eqns. 3.18 and 3.19) have an adjustment that is directly proportional to the input variance, since the variance of $e(n)$ is increased proportionally to that of $s(n)$, following from

$$E(z) = S(z) A(z) \qquad \text{from (1.6)}$$

The optimal coefficients, however, are not affected by the input variance and so the size of the adjustment made to them should not be changed. Thus K in this study, is some experimentally determined base value, divided by the input variance.

A second adjustment that is made helps prevent a steady-state oscillation of the variable around the optimal value. This is done by reducing the value of K in each iteration - accomplished by dividing K by the iteration number. This is justified since as the method converges on the optimal value, large adjustments will no longer be necessary - a principle is akin to the golfer alluded to earlier, selecting a higher club for shots nearer the hole.

These two refinements are the only ones made for K_a . Even with these adjustments, however, it was found that when the method of steepest descent was used to find the b coefficients which model the zeros, it could still become unstable. This instability was most pronounced when the number of zeros was small and when the number of samples in the frame was large. Thus K_b should be reduced for these two cases, an effect introduced by dividing K_b by the frame length and multiplying it by the number of zeros.

Thus, in this study, the following proportionality values are used in eqns 3.17 and 3.18.

$$K_a = \frac{0.05}{E[s(n)^2] \cdot IT} \quad (3.20)$$

and

$$K_b = \frac{0.05 \cdot q}{E[s(n)^2] \cdot IT \cdot N} \quad (3.21)$$

where:

$E[s(n)^2]$ = the variance of the input signal $s(n)$

IT = the iteration number of the the method of steepest descent

q = the number of zeros

N = the number of samples in the frame

Finally, it is necessary to determine how many iterations should be made for this

application of the method of steepest descent. Since the optimal values for the coefficients are unknown, it is impossible to measure directly how close the method has come to this optimal solution. With many other numerical methods, for example Newton's method to solve for the roots of a polynomial, substitution of the current "best value" into the original equation will yield a number that is a direct measure of how close the solution is to the optimal solution. In Linear Predictive Coding, however, even the optimal values will not yield perfect prediction (and thus a residual with zero variance) - they will merely yield a residual variance that is the minimum possible.

Since the method of steepest descent will converge on the minimum solution (assuming it is stable) an indirect, yet accurate, measure of how close the solution is to the optimal one is how much further improvement each iteration yields. When the improvement becomes negligible, the present solutions can be assumed to be as close to the optimum as the system will approach. A trade-off must be made here between computation time and the desired accuracy.

The number of iterations necessary was determined experimentally by plotting results after each iteration and thus determining when improvement becomes negligible.

This experiment was carried out on a samples of real speech, both voiced and unvoiced, for a 12-pole filter, a 6-pole-6-zero filter and a 12-zero filter. The results are presented in Appendix A. They have been averaged and this average is presented below. The "Gain" plotted on the y-axis is the ratio of the input variance to the residual variance on a decibel scale, a common measure of LPC performance.

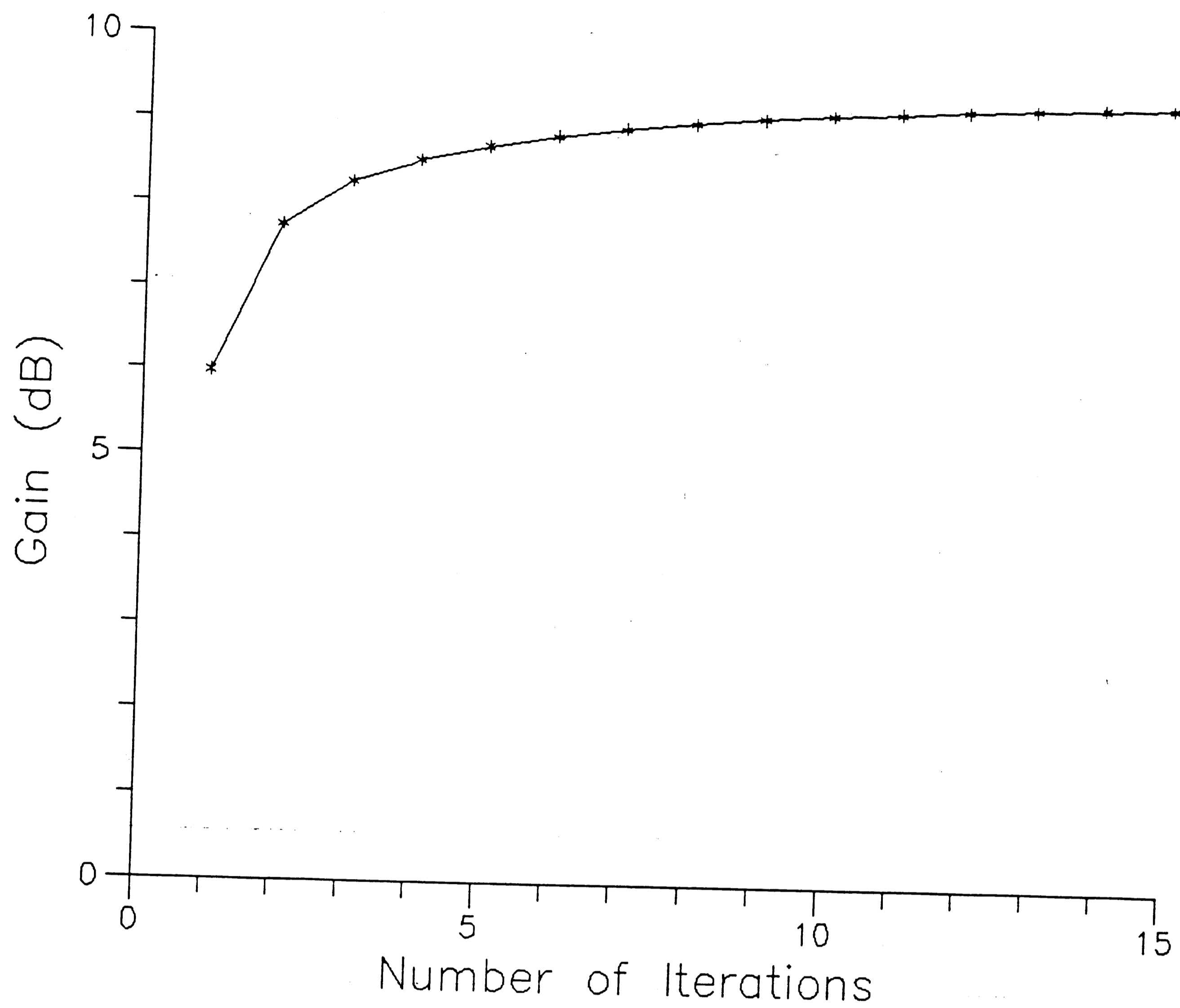


Fig 3.2 LPC Performance improvement with number of iterations

Based on the graphical information shown in Fig 2.3, it was decided to use six iterations for the rest of this study, as a good compromise between performance and computation time.

U

Chapter 4

Experimental Background

In this study, Linear Predictive filters are simulated using a digital computer. The simulation programs were written in BASIC since rapid run-time is not a consideration in this study, and BASIC allows the algorithms developed to be quickly and easily converted into program code.

Speech Data

Two types of speech data are operated on in this study. The first is synthetic speech produced by C.S. Holzinger [Hol88]. This speech waveform is derived by modelling the vocal tract as a concatenated series of lossless acoustic tubes [Rab78]. This model leads to a vocal tract that has a transfer function that contains only poles. The lossless tube model is excited by a signal that resembles a glottal pulse. This "glottal pulse" has a finite length, and thus must be the result of some Finite Impulse Response (FIR), leading one to conclude that the spectrum of the "glottal pulse" contains only nulls [Opp75]. The spectrum of this synthetic speech thus contains peaks due to the vocal tract response and nulls due to the glottal excitation. A second type of synthetic speech was obtained by modifying Holzinger's program to derive the synthetic speech above so that the lossless tube model is now excited by an impulse. The spectrum of this (less realistic) synthetic speech should thus contain no nulls at all - a property that will be useful to demonstrate some of the effects that zeros in the LPC filter responses does have on the system performance. Examples of both types of synthetic speech may be located in Appendix B.

Real speech data is also used in this study. This speech was captured and quantized by C. Woloszynski [Wol86], whose programs have been adapted in this study to ensure the data is compatible with the Basic programs that are used to analyze them. From the voluminous data captured by Woloszynski, sixteen samples have been selected: ten of these are examples of voiced speech and six are examples of unvoiced speech. These samples were selected to represent a wide range of waveshapes, amplitudes and, for the case of the unvoiced samples, zero-crossing rates. The samples are not further differentiated into the speech sounds that produced them since at this stage in an LPC analysis, the only information usually available about the sample is whether it is voiced or unvoiced. The speech is low-pass filtered (to prevent aliasing) at 34 KHz and is sampled at 15 KHz. The examples selected for processing are each 128 samples long and thus correspond to 8.5 milliseconds of speech. These samples can be found in Appendix B. As was explained in Chapter 1, it is expected that the real speech samples will have spectra containing both peaks and valleys.

Computer Programs Used

Two previously written BASIC programs have been adapted for use in this study. A plotting routine is based on a program originally written by Holzinger [Hol88] and a program to perform the Digital Fourier Transform of time-domain waveforms incorporates much of a program written by Wagener. The program used to find the optimal all-pole coefficients is based on Durbin's recursive method as presented in [Rab78].

Chapter 5

Simulations and Results

At this stage, it should be recalled that the purpose of this study is to investigate the effect that incorporating zeros into the transfer function of the synthesis and analysis filters has on the performance of the LPC system. This is done by simulating the pole-zero filters proposed in Figures 2.5 and 2.6. These general-case filters can be easily converted into all-pole or all-zero filters by setting to zero the number of zeros and poles respectively. A test was performed to ensure that the pole-zero model with no zeros behaved in exactly the same manner as the all-pole model as described in Figures 2.1 and 2.3, and this was found to be true.

The first of the tests described here establishes the effect that replacing some or all of the coefficients that model poles with coefficients that model zeros.

Test 1

(P + Q) = 12 : Ratio varied

Voiced Speech

The simulation program was written so that each time an analysis was done, the same data was analyzed using differing ratios of poles to zeros. The sum of the number of poles and zeros, however, was held constant so that the total amount of predictive information is unchanged. A common standard for the number of poles in an all-pole model is twelve [O'Sh87] and thus in the first simulation the sum of the number of poles and zeros is held at twelve. A test is performed for each combination of poles and zeros

within this limit, on each of the ten samples of voiced speech shown in Appendix B. As a comparison, an all-pole analysis was performed on each sample using Durbin's method to obtain the optimal coefficients.

The performance measure used for these tests is the ratio of the variance of the input signal to the variance of the residual waveform. This is known as the gain and is expressed here in decibels. A comparatively large gain indicates that the system is performing well in predicting the current speech sample.

The results of these tests are presented in Appendix C. The results have been averaged separately and are graphed in Figure 5.1.

The results presented here show clearly that, at least for the case of voiced speech, there is a definite advantage in using pole-zero LPC filters instead of all-pole LPC filters. This is not entirely surprising in light of the discussion in Chapter 1 on the production of natural speech. A more surprising aspect of the results is that the optimal combination of poles and zeros is *one* pole and eleven zeros. Detailed examination of the results in Appendix C show that this combination was the best in 80% of the tests conducted and a ratio of two poles to ten zeros was most successful in 10% of cases, as was the ratio of three poles to nine zeros. All cases where the best combination was *not* one pole and eleven zeros occurred when the input waveform had been clipped because the amplitude was larger than the quantizer used could account for. The significance of this will be discussed later.

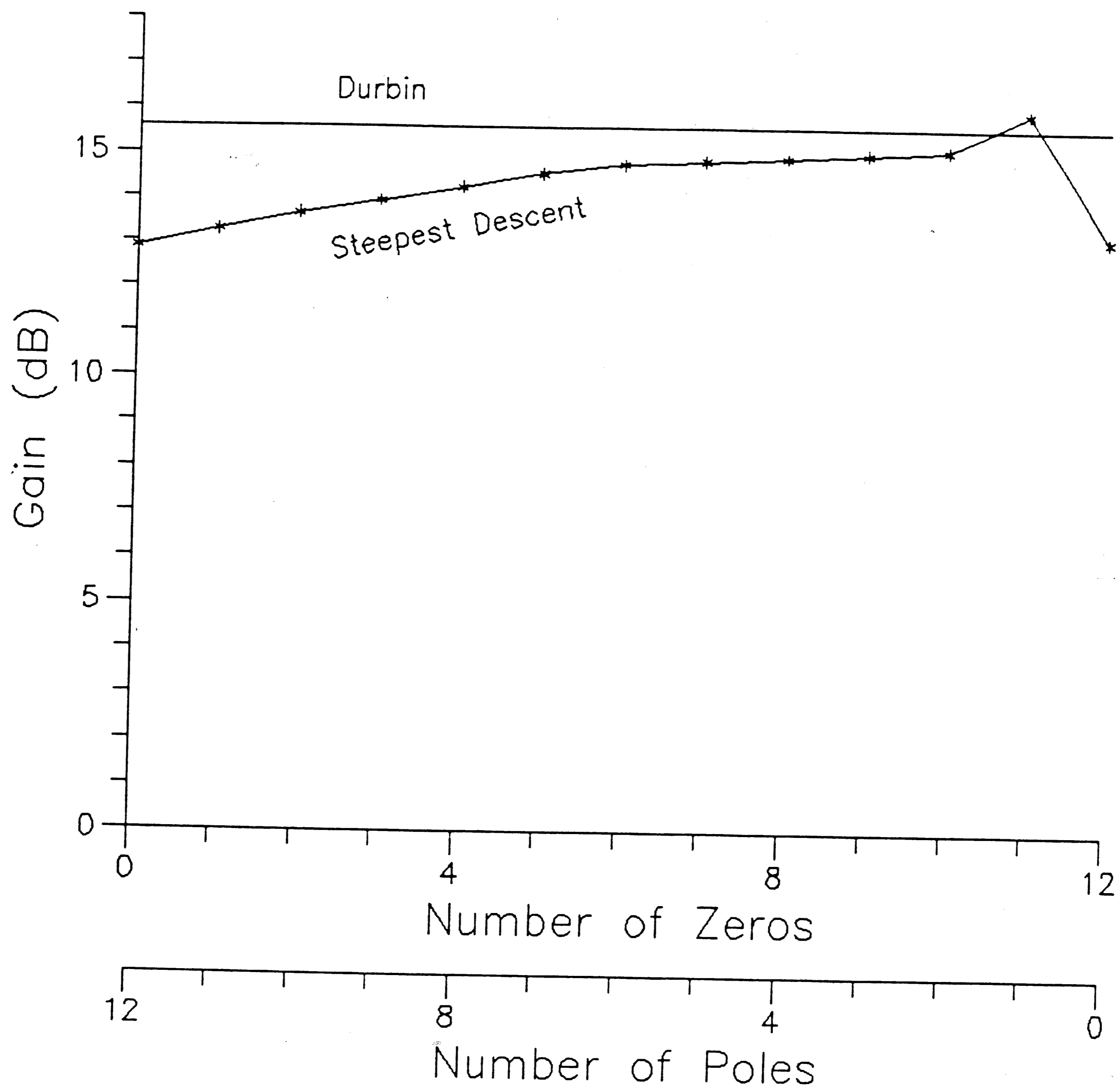


Fig 5.1 Gains for Voiced Speech

Another important fact that can be drawn from Fig. 5.1 is that *any* combination of poles and zeros performs better than the all-pole case including the all-zero model. This would seem to give lie to any justification for using the all-pole model other than the speed and accuracy with which the coefficients can be obtained. This all-zero model, however, performed significantly worse than when just one pole was included.

The last preliminary conclusion from Fig. 5.1 is that the best combinations of the pole-

zero model outperform even the optimal all-pole models derived from Durbin's equations. This occurred even though the method of steepest descent used for the pole-zero model is not an optimum solution and only a strictly limited number of iterations have been used! It should be noted, however, that computation time to find the pole-zero coefficients was significantly longer than that needed to find the Durbin coefficients.

Test 2

(P + Q) = 12 : Ratio varied

Unvoiced Speech

A second test was performed under exactly the same conditions as Test 1, except that the speech data analyzed are now samples of real unvoiced speech (see Appendix B for plots of these samples). As in Test 1, the sum of the number of poles and zeros is twelve and the performance measure used is the ratio of input variance to residual variance. The test results are presented in Appendix C and have been averaged to form the graph in Fig 5.2.

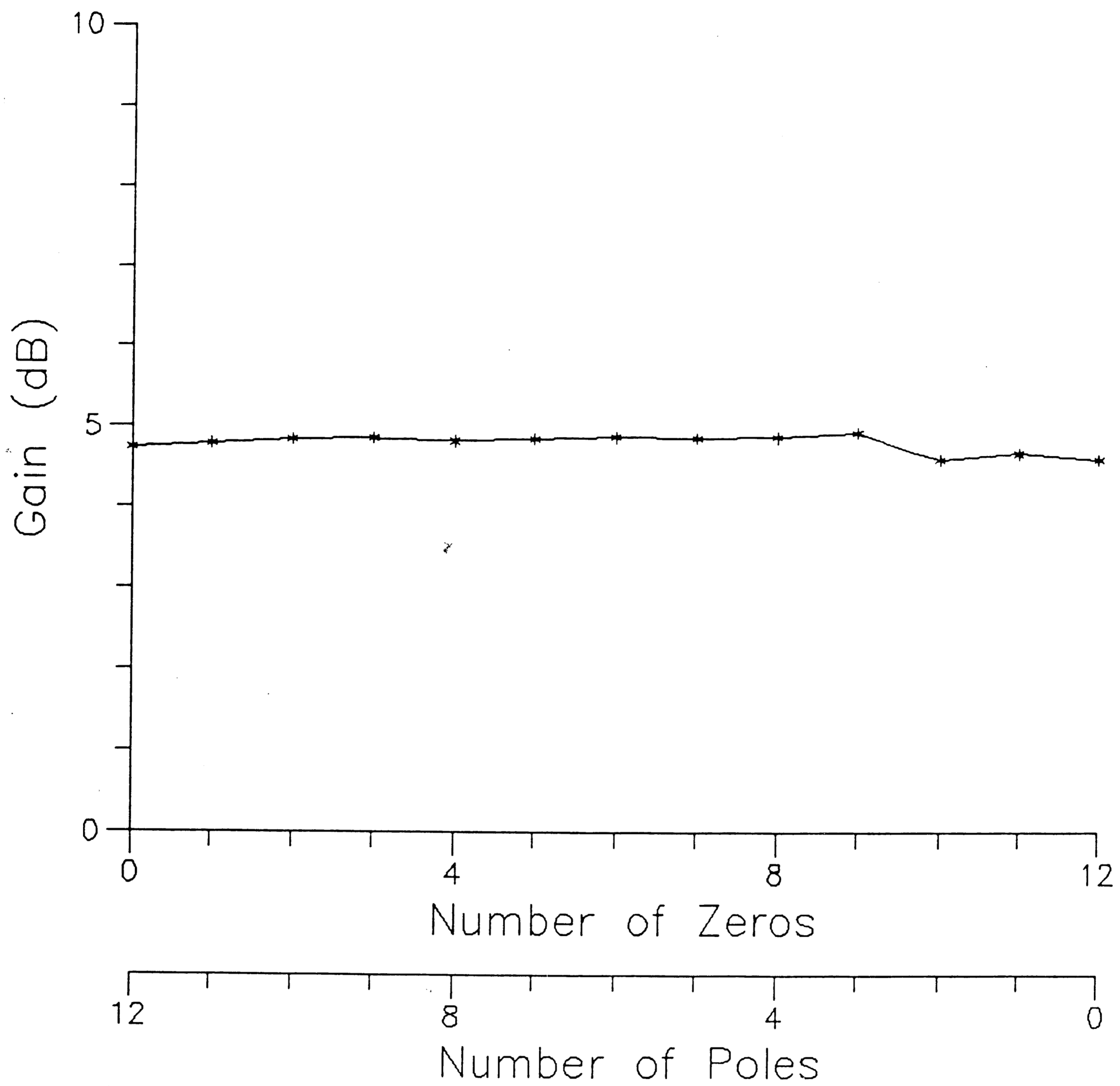


Fig 5.2 Gains for Unvoiced Speech

The results obtained here for unvoiced speech are not nearly as clear or as conclusive as those obtained in Fig 5.1 for voiced speech. It does seem that some tiny advantage exists in using some form of pole-zero combination but it is not even clear what this combination should be. A combination of three poles and nine zeros produced the best result on average but this was the optimal combination in only 33% of the samples taken.

The effective conclusion drawn from Fig. 5.2 is that all-pole and pole-zero analyses perform approximately as well as each other for unvoiced speech and thus only voiced speech is considered further in this study.

The results of Test 2 also hint that the improved performance of the pole-zero combination lies primarily with its ability to model the zeros that occur as a result of the glottal pulse, and which are thus not present in unvoiced sounds, as opposed to the zeros due to the vocal tract transfer function (or the effects of radiation or low-pass filtering), which are present in unvoiced speech. This possibility is explored further in Chapter 6.

Test 3

(P + Q) = 12 : Ratio varied

Synthetic Speech (Glottal excitation)

The procedures used in Tests 1 and 2 are used again in Test 3, except that the speech sample operated on is an example of voiced synthetic speech. This speech has been synthesized using an excitation resembling a glottal pulse. The parameters and performance measures are as before, the raw results can be found in Appendix C and they are presented graphically in Fig. 5.3.

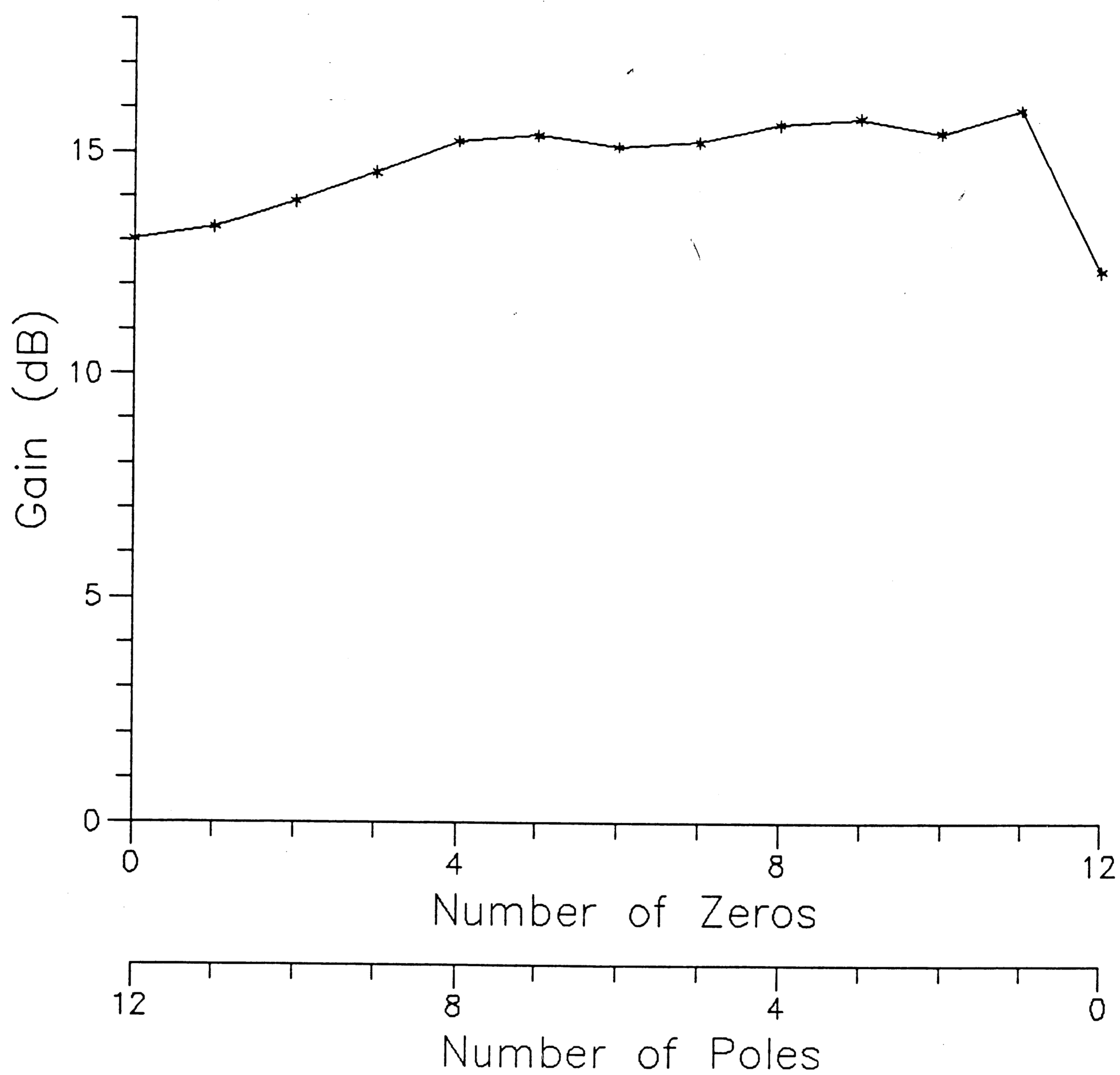


Fig 5.3 Gains for Synthetic Speech (Glottal)

The results of Test 3 again show a significant improvement in performance when zeros are included in the transfer functions of the LPC filters. The improvement is very similar to, although slightly less marked than it was in the case of real voiced speech and the optimum combination of coefficients is still found to be one pole and eleven zeros. To understand the particular significance of this experiment, it should be recalled from

Chapter 4 that this synthetic speech is created with an *all-pole* model of the vocal tract. Thus the only spectral valleys in the synthetic waveform occur as a result of the glottal pulse excitation, but still the modelling of these zeros seemed to be of significantly more importance than modelling the poles that occur due to the response of the vocal tract. This result thus concurs well with the initial conclusion drawn from the unvoiced speech results, that the most important zeros to model are those that occur due to the glottal pulse.

Test 4

(P + Q) = 12 : Ratio varied

Synthetic Speech (Impulse excitation)

The final test in this set performs the same analysis as in Tests 1-3, except that the data now analyzed is synthetic speech, where the vocal tract model is identical to that used to produce the synthetic speech of Test 3, but the excitation is now an impulse. It was found that the proportionality constant used in the method of steepest descent had to be reduced from the value presented in eqn 3.21 to prevent instability in this special case where no the data analyzed has no nulls. Again the raw data may be found in Appendix C, and is presented in graphical form in Fig. 5.4.

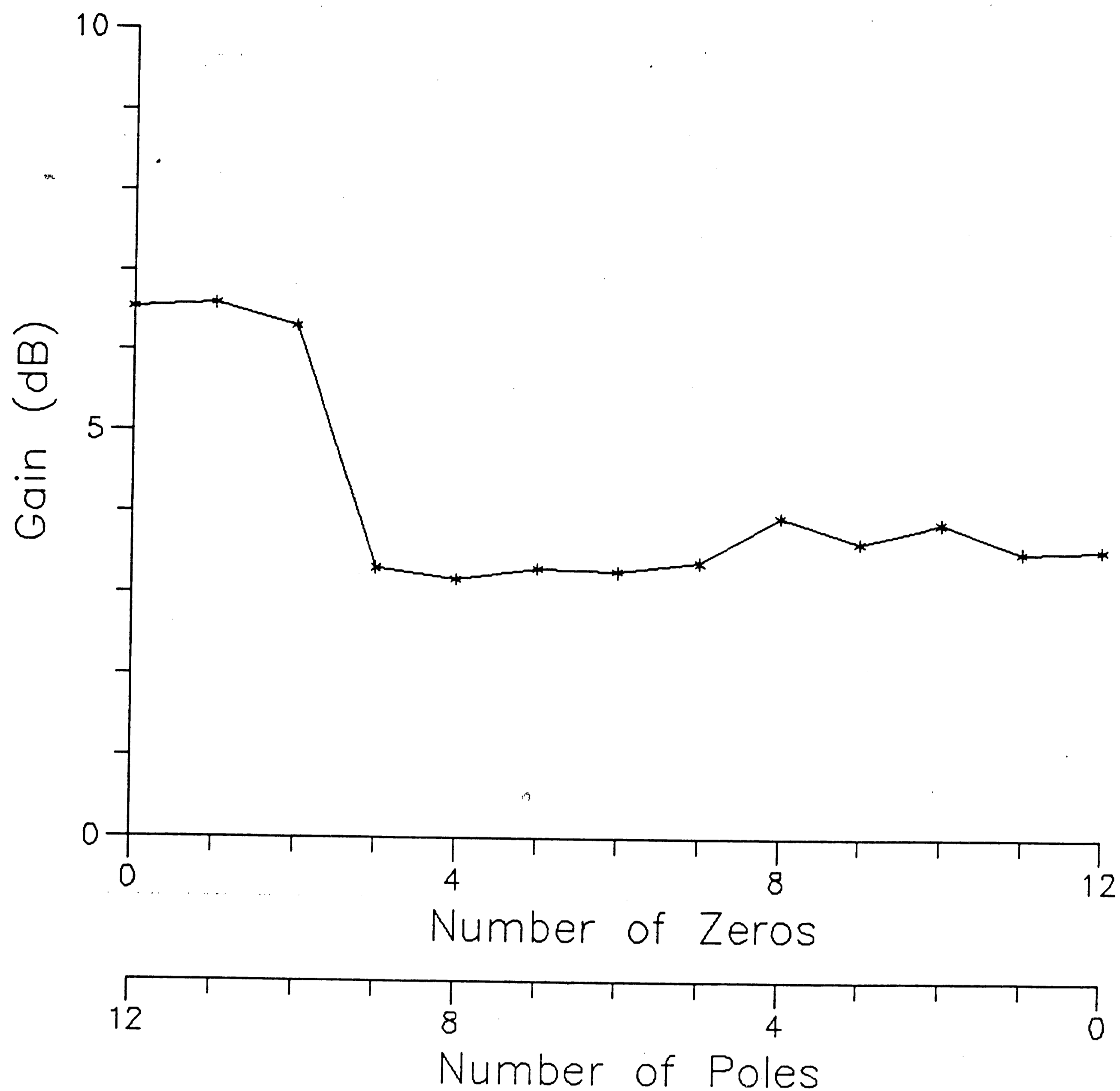


Fig 5.4 Gains for Synthetic Speech (Impulse)

The graph in Fig 5.4 shows that, for this particular example of synthetic speech, a pole-zero combination performs worse than the all-pole filters, even when these all-pole coefficients are found using the sub-optimal method of steepest descent. The slightly better result for the 11-pole-1-zero case is explained later by the results of Test 5. The overall result comes as no surprise when it is recalled that the impulse excited synthetic speech should have no nulls at all in its spectrum. The excitation function (an impulse) has a flat frequency spectrum and the vocal tract is modelled as an all-pole filter. When

compared with the results obtained in Test 3 for the glottally-excited synthetic speech, a more substantial basis is built for the contention that the zeros that are important in LPC analysis are those that derive from the glottal pulse.

The large difference in performance between the nine-pole and ten-pole cases can be attributed to the fact that the vocal tract model used in the synthesis of this speech has exactly ten poles in its transfer function.

Test 5

Q = 0 : P varied

Synthetic Speech (Impulse excitation)

The fact that a pole-zero combination fared more poorly than the all-zero model in Test 4 does not mean that the zeros, themselves, degraded the performance of the system, or even that they had no effect. Each zero in Test 4, displaced a pole from the combination and thus the result shows that, in Test 4, the addition of a zero did not fully compensate for the loss of a pole. To obtain a clearer idea of what the independent effect of the zeros was, Test 5 shows the system performance with no zeros at all, but for a decreasing number of poles, again using the impulse excited synthetic speech. When this result is compared to that obtained in Test 4, the effect of the zeros can be clearly seen. To assist in this comparison, the results from Test 4 are replotted in Fig. 5.5, along with the new, all-pole data.

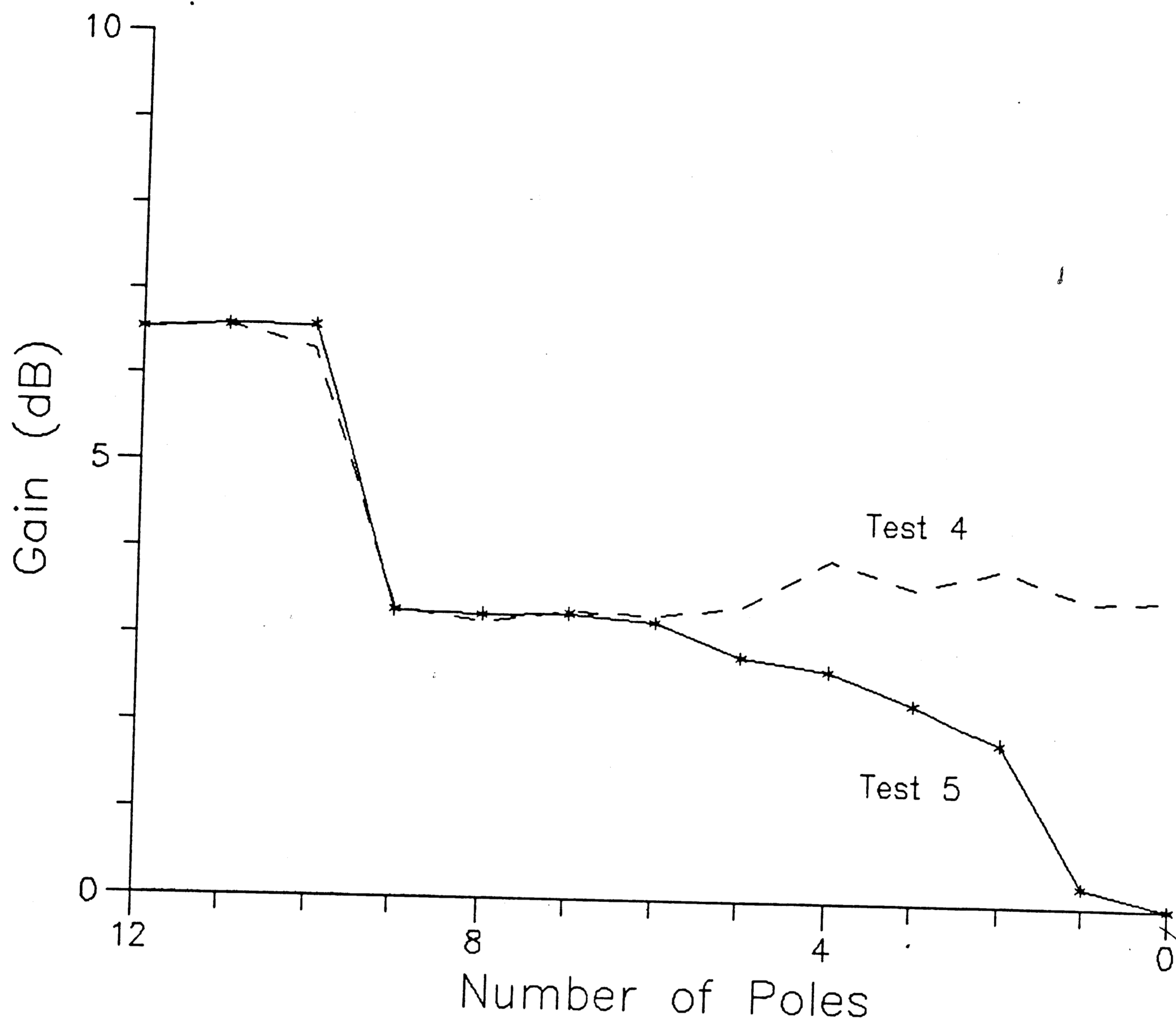


Fig 5.5 Gains for Synthetic Speech (Impulse)

Figure 5.5 shows that when a relatively large number of poles are used to model $H(z)$, then the addition of zeros has little effect. When only a small number of poles is used, however, the addition of zeros does show significant improvement, even though there are theoretically no zeros in the signal. This can be explained in terms of the spectral balance - a large number of zeros can be used to model missing poles in the same way that an all-pole LPC filter uses additional poles to model the missing zeros. In each case the spectral balance is preserved better than the local peaks and nulls.

The simulation also shows the ten-pole case performing slightly better than the twelve-pole case which contains more predictive information. This is because the vocal tract model used to synthesize the speech has only ten resonances and thus exactly ten poles occur in its transfer function. The flat spectrum of the impulse excitation will add no further poles to the system and thus coefficients to represent additional poles serve no purpose and should ideally be set to zero or else they will degrade performance. This result, it should be observed, is specific to impulse excited synthetic speech but it provides an insight into the accuracy with which the method of steepest descent is able to model the speech production mechanism.

Test 6

(P + Q) = 6 : Ratio varied

Voiced Speech

In this test, the same conditions and data are used as in Test 1. The difference here is that the combined number of poles and zeros is reduced to six. This is done to ensure that the important results obtained in Test 1 are not specific to the use of twelve coefficients. Again, the performance test is the ratio of the input variance to the variance of the residual.

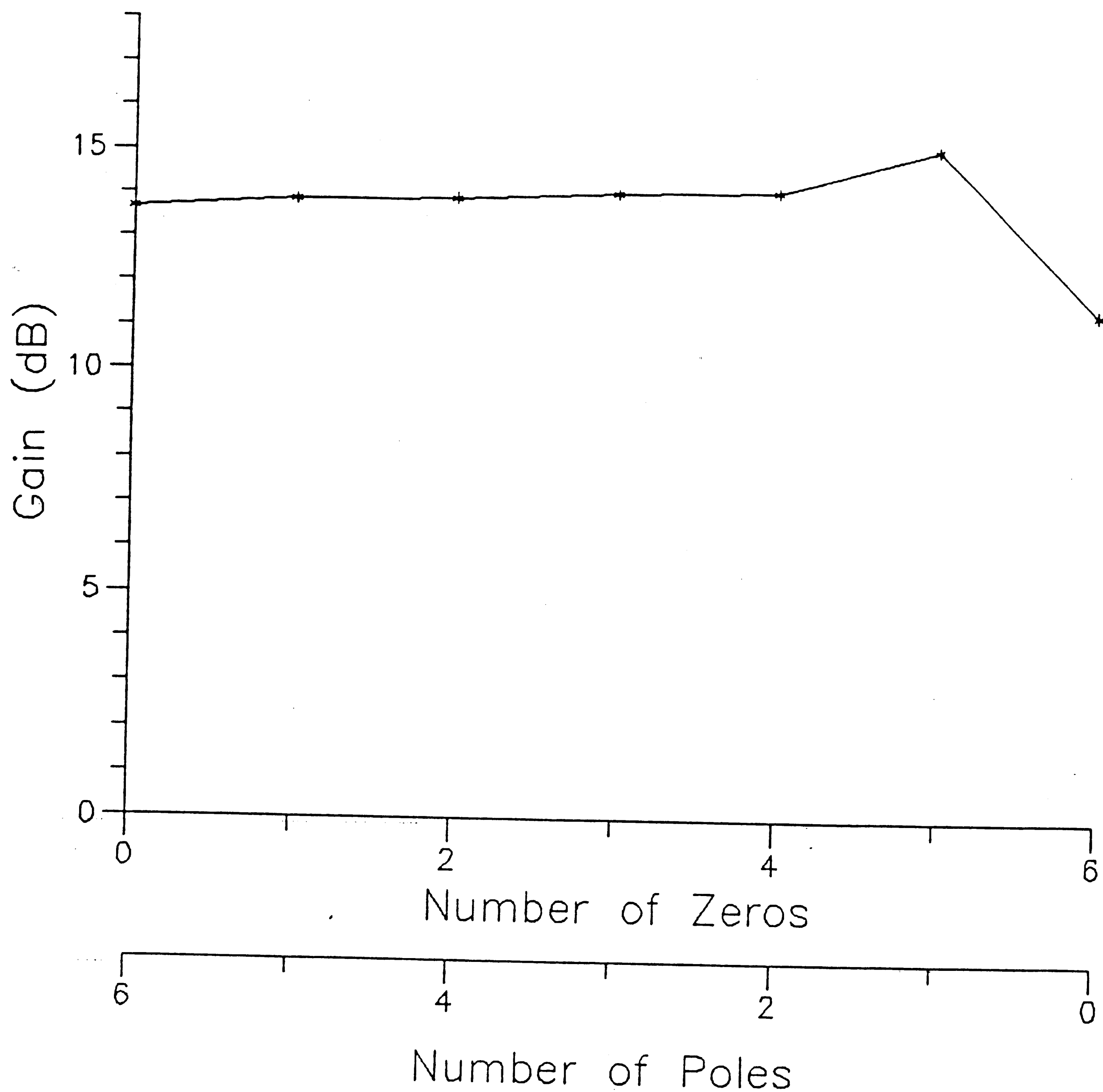


Fig 5.6 Gains for Voiced Speech

The graph in Fig. 5.6 shows results that are very similar to those obtained in Test 1. Again, the optimal combination of poles and zeros is to include only one pole and to use the remaining coefficients to represent zeros.

Similar results were obtained when the sum of the poles and zeros was set to other values but it is felt that repetition of these does not serve to illuminate the matter

further.

Test 7

(P + Q) = 12 : Ratio varied

Voiced Speech

In Test 7, and in all the remaining tests conducted in this study, the measure used to gauge the performance of the system is changed. As described in Chapter 1, the ratio of the input variance to the residual variance is a highly accurate measure of the performance of an ADPCM system, but it may not best represent the performance of an LPC system (in which the residual is not used at all).

In this test, the entire LPC system is simulated, as opposed to simply the analysis filter that was sufficient in Tests 1 to 6. The waveform is analyzed to determine the optimal coefficients, and then an approximation of the waveform is re-created by exciting the synthesis filter with an impulse train (since voiced speech is being analyzed). The original and reconstructed waveforms should be compared in the frequency domain, not the time domain, since this parallels more closely the human hearing process in which only the magnitude of the spectrum is detected. (As it turns out, the time-domain waveforms of the reconstructed signal and the original signal are usually very different, even when their spectra are similar.) Thus Digital Fourier Transforms are then taken of both the reconstructed waveform and the original data sample (as a reference) to obtain their spectra, and the magnitudes of these are then compared.

The gain of the system is accounted for by ensuring that the variance of the

reconstructed signal matches that of the original signal, thus making the height of the impulse excitation irrelevant. (It should be noted that this is clearly impossible to do in a real system where the original signal is not available to the receiver and many schemes have been developed to determine the height of the excitation impulse train [Rab78], but these are felt to be irrelevant to this study). Problems occurred in the detection of the pitch period from the residual (in reality, the pitch is determined from a number of sources, including the residual [O'Sh87]) and again, since the pitch period is only tangential to this study, a crude simplifying assumption was made and the pitch was set to a constant of 128 samples. This value of 128 was determined by inspection of the data waveforms used. This assumption is expected to degrade overall system performance but it is not expected that it should affect the balance between the contributions of the poles and the zeros. The actual performance measure used is the ratio of the variance of the original spectrum to the variance of the difference between the original and reconstructed spectra. This ratio is expressed in decibels. As before, this test was performed on all ten samples of real, voiced speech. A total of twelve coefficients are used, and they are used to represent varying ratios of poles and zeros. The full results are presented in Appendix C, and have been averaged to obtain the data shown in Fig. 5.7.

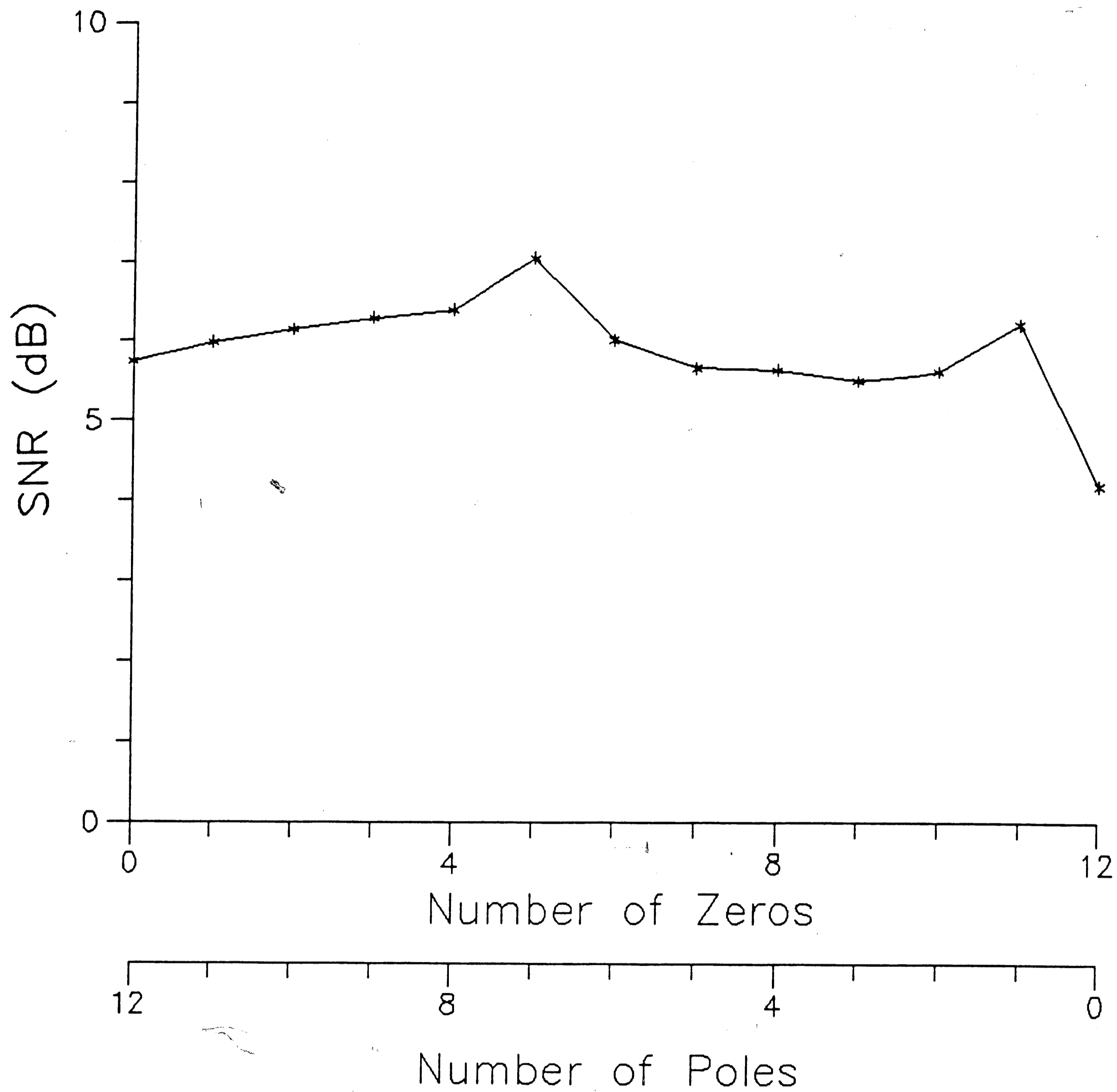


Fig 5.7 Reconstruction SNR for Voiced Speech

Figure 5.7 would indicate, once again, that there is a definite advantage to be had in using a combination of poles and zeros. An interesting feature of the graph is that the optimal combination is found to be no longer one-pole and and eleven zeros as occurred in Test 1, (which uses the same data and constraints, only a different performance measure) but now a more evenly balanced mixture, peaking at seven poles and five zeros. Once again, there is considerable degradation for the all-zero case. Figure 5.7 is

misleading, however, in that few of the voice samples analyzed showed this relationship between gain and pole-zero combination - in fact only 10% of the samples had a best ratio of seven poles and five zeros. Most of the voice samples were of one of two types, either peaking at the all-pole case (20%) or at the 1-pole-11-zero combination (30%). This gave rise to individual graphs falling, broadly, into one of the two types shown in Fig. 5.8. The graphs in Fig. 5.8 are the gain against pole-zero combination plots for the voiced samples 5 and 1 shown in Appendix B.

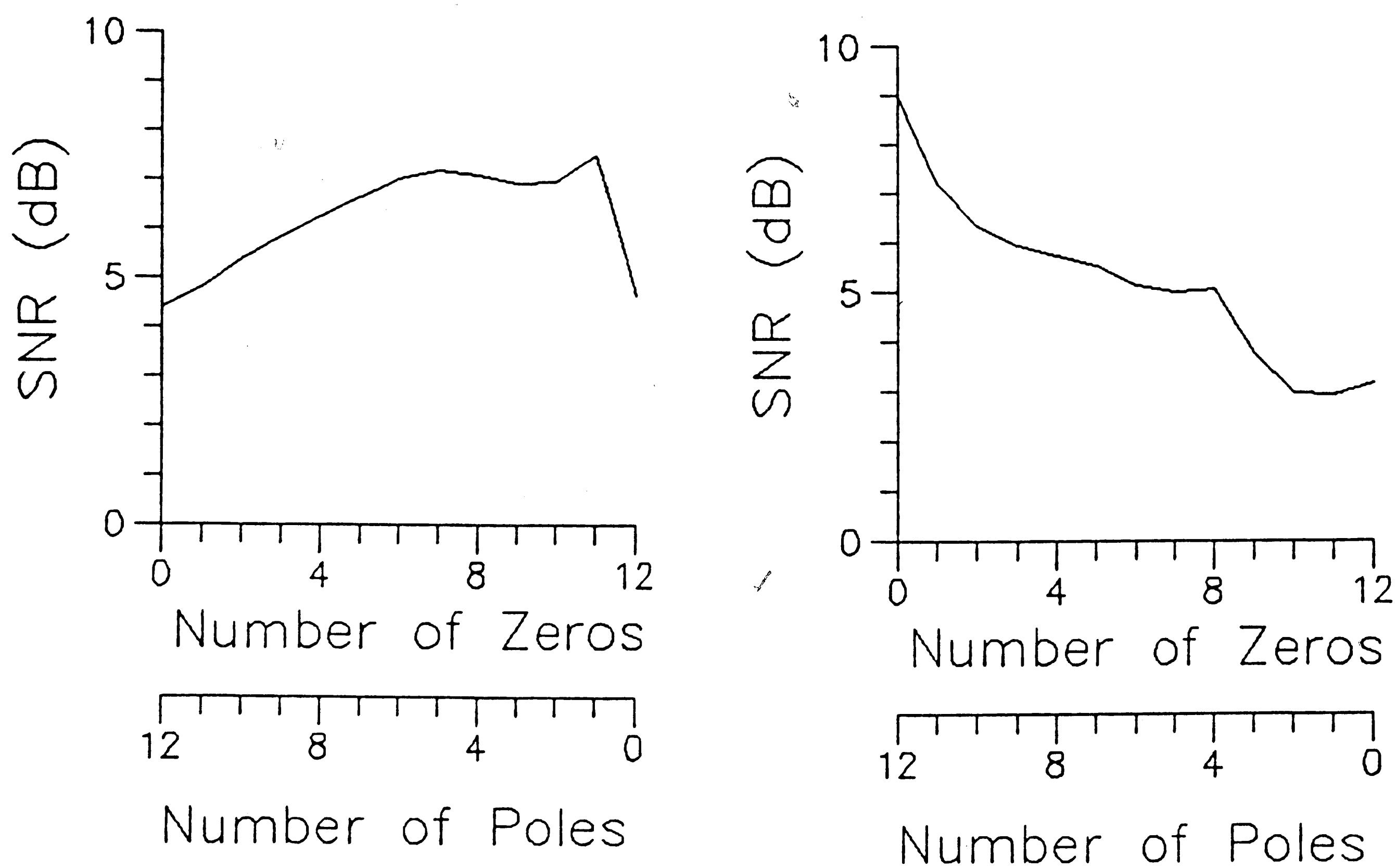


Fig 5.8 Reconstruction SNR for Voiced Samples 1 and 5

When plots of these types were averaged, they *combined* to form the relationship shown in Fig. 5.7.

Examination of the voice samples showed that the all-pole model was superior for those voice samples which had been clipped due to the input waveform being too large, while

the single-pole-many-zero model performed better when no clipping occurred. Note that amplitude itself did not account for this change - it was the presence of clipping that caused the better all-pole performances. A reasonable explanation of this is that the clipping introduces erroneous high frequency components into the spectrum of the input waveform and these erroneous components may well be better represented by poles than by zeros. It should be recalled that a similar change occurred using the earlier performance measure (Test 1) when the single-pole-eleven zero model could be bettered only when in the presence of clipping. When only the samples in which no clipping occurred are considered, the average of the results is now shown in Fig 5.9.

These results, for unclipped input waveforms, now show the result that is expected from Test 1 - that a definite advantage occurs when both poles and zeros are used and that the optimal combination of these poles and zeros is, again, one pole and eleven zeros. This has shown that the simpler measure of performance, using the ratio of the input variance to that of the residual, gives rise to essentially the same results as the more complex reconstruction measure.

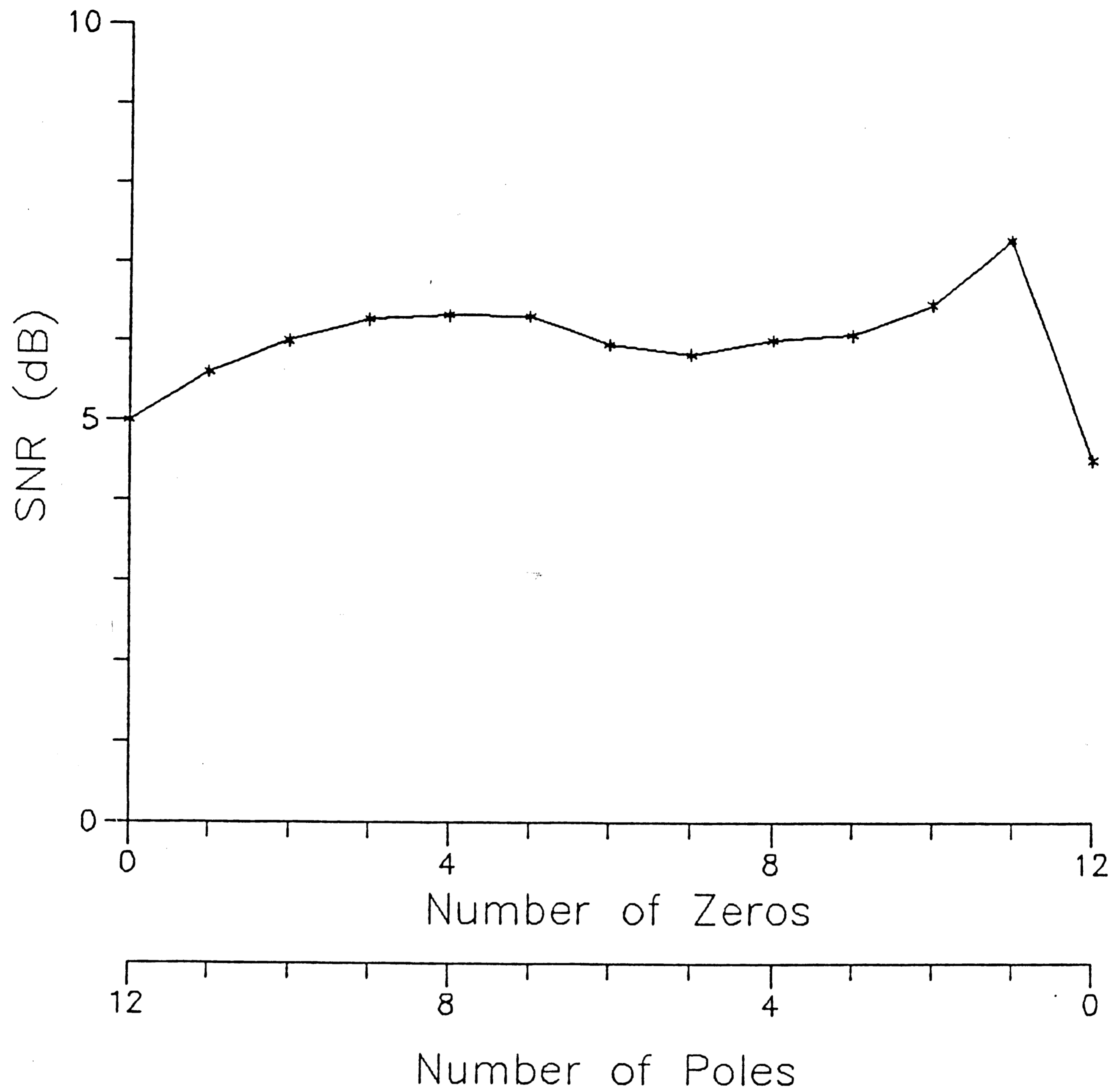


Fig 5.9 Reconstruction SNR for Unclipped Voiced Speech

Test 8

(P + Q) = 12 : Ratio varied

Synthetic Speech (Glottal excitation)

The same procedure as was followed in Test 7, is now performed in this test, with the

exception that the data examined is the glottally excited synthetic speech. This speech is not clipped at any point. The results are to be found in Appendix C and are plotted in Fig. 5.10.

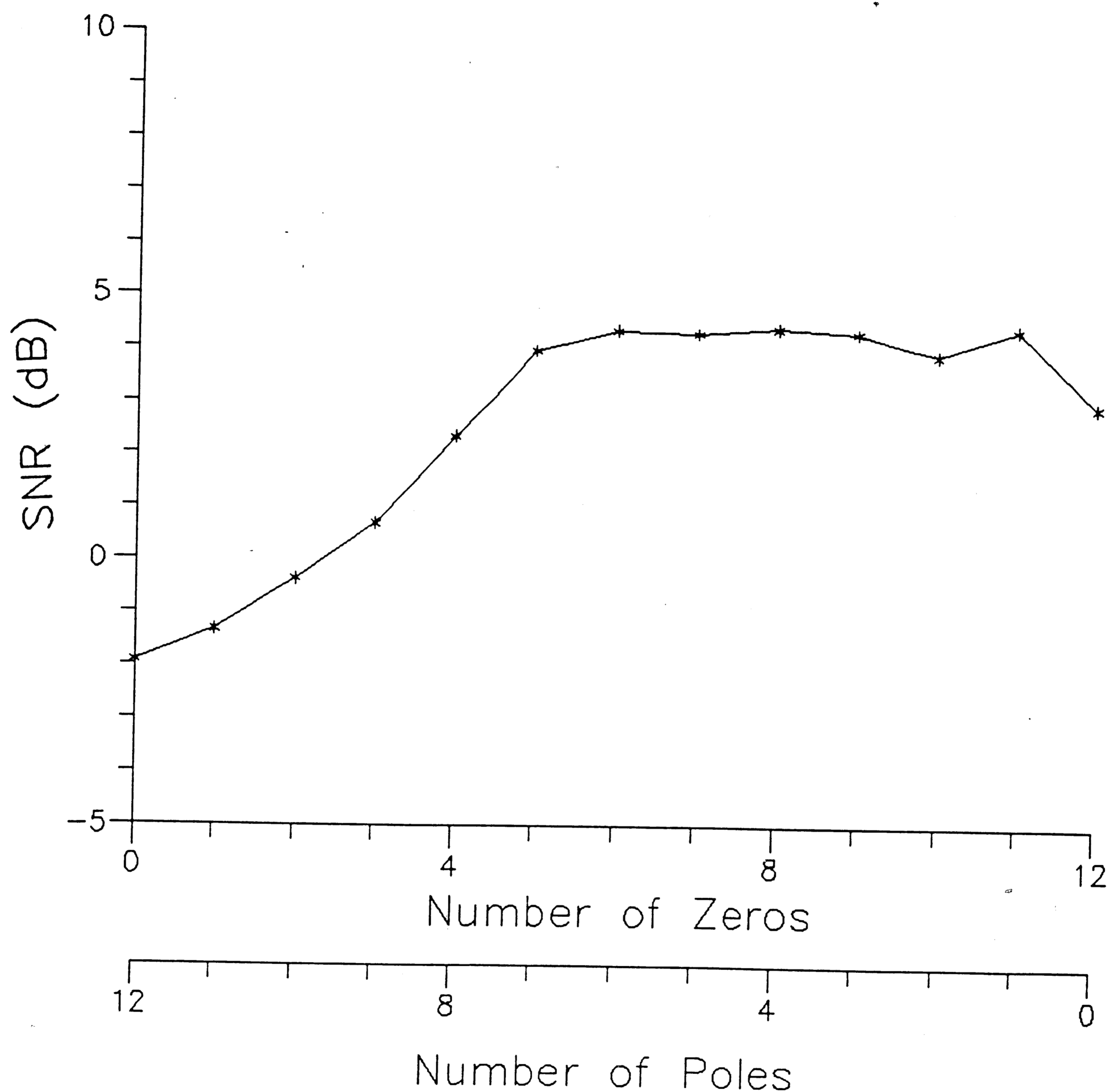


Fig 5.10 Reconstruction SNR for Synthetic Speech (Glottal)

Once again it is shown that the results obtained in Test 3 using the simple performance measure are repeated when the performance measure is a comparison of the reconstructed waveforms.

Chapter 6

Conclusion

The results of Test 1 (described in Chapter 5) provide the most important conclusion of this study, viz. *a pole-zero representation of speech can out-perform the classic all-pole representation.*

It is clear that the addition of zeros to an all-pole model will improve performance, and this has been superbly illustrated by B. Atal in [Ata78] in which reconstructed spectra are compared graphically with the input spectra. Such a result is no surprise - the addition of zeros has increased the amount of predictive information available. It is also well known that the performance of an all-pole LPC filter improves with every pole added [Ata71] - again the result of an increase in predictive information. Atal's studies raised the question of whether the addition of a zero increases performance as much as the addition of another pole. Put another way, will the replacement of a pole by a zero improve or degrade performance? This study shows that the answer is that improvement will occur.

This conclusion was also arrived at by Koo and Gibson [Koo86] but these researchers failed to address the question of *how many* poles could be replaced by zeros while still maintaining improved performance. To quote from their paper "it was not clear which of the combinations of pole and zero orders ... would yield the best results." The results from Test 1 show that the best combination of poles and zeros is one pole and the remaining predictive information modelling zeros. This is surprising in its suggestion that modelling the zeros is more important than modelling the poles when almost all LPC

systems use all-pole representations.

A conclusion about whether an all-pole or all-zero predictor is more efficient is difficult to arrive at from the data gathered in this study. Test 1 shows a slightly better performance from the all-pole predictor, but the reverse was found to be true in Test 6 for a smaller number of predictor taps, and in Test 7 where reconstruction is considered. Koo found a slightly improved performance for the all-zero case [Koo86] and then proceeded to show how the all-zero predictor performs far better when transmission occurred over a noisy channel since the finite response of an all-zero filter means that the effect of a channel error is more quickly eradicated. This advantage of the all-zero predictor in a noisy environment can be expected to be passed on, in part, to the pole-zero predictors, giving them a further advantage to that described in this study.

Test 2 repeats the study undertaken in Test 1 except that the data analyzed is unvoiced speech as opposed to the voiced speech of Test 1. For this case, it was found that the all-pole, pole-zero and all-zero predictors all performed at a very similar level. This discrepancy indicates that the advantage found in pole-zero representation of voiced speech stems from the spectral nulls that occur as a result of the glottal excitation, which is not present in unvoiced speech. It should be noted that a theoretical study of the shape of the vocal tract indicates that the sounds for which zeros occur most prominently in the transfer function of the vocal tract are nasal sounds and fricatives [O'Sh87]. Unvoiced sounds are almost exclusively fricatives, and so, if the vocal tract was contributing significantly to the spectral nulls, then a pole-zero representation should show an advantage for fricatives. Since this was found to be *not* the case, it can be concluded that the vocal tract can be adequately modelled by an all-pole filter, but

that the glottal excitation is best represented by a transfer function that includes zeros

This hypothesis is further investigated in Test 3, in which synthetic data is examined. This synthetic data been produced using a finite glottal pulse (which thus has an all-zero transfer function) and an all-pole model of the vocal tract. The results of this test are very similar to those obtained for real voiced speech indicating once again that the glottal pulse is the source of the representationally important zeros. By this reasoning, if the same vocal tract model is used but the excitation no longer contains spectral nulls, then a pole-zero representation should perform comparatively poorly. This is shown to be the case in Test 4 where the vocal tract model is excited by an impulse. The significance of this test is that it traces the pole-zero representation improvement in the glottally excited synthetic speech *directly* to the glottal pulse since this is the only change in test conditions.

The clarity of the results shown in these tests is tempered by the fact that these tests, in an effort to reduce the influence of external factors, are a simplification of the complete LPC process. The performance measure used is the ratio of the residual variance to that of the input waveform. The results obtained thus will correlate directly to an improvement in an ADPCM system (where the synthesis filter is excited by the residual) in which quantization of the coefficients and channel noise is negligible. Koo's work has shown an improvement in the performance of an all-zero ADPCM system over an all-pole one in the presence of channel noise and so it is reasonable to assume that a pole-zero ADPCM system will perform better than an all-pole one in the same noisy conditions. Quantization of the coefficients (for low bit-rate transmission) has been found to have two degrading effects for all-pole analysis filters. By shifting the pole

positions they make the model of the speech production process less accurate, but more importantly, if the poles are shifted outside the unit circle on the z -plane, then the all-pole filter will become unstable. No evidence has been gathered here to suggest that zero positions are affected by quantization more or less than pole positions and no reasoning has been proposed in the literature why either case should occur. Stability should be less of a concern for the pole-zero representations since if the synthesis filter is still stable when zeros occur outside the unit circle. Thus it is indicated that the pole-zero representations will show an even larger gain over the all-pole representations when 'real-life' factors are considered, although experimental confirmation of this is beyond the scope of this study.

Further complications occur when the ADPCM system is replaced by an LPC one, in which the synthesis filter is excited by an impulse train or a noise source rather than the residual. Now it is no longer sufficient that the residual is minimised - it must be brought as close to the shape of an impulse train as is possible (for voiced speech). It is also no longer clear exactly how the performance of the system should be measured. Perhaps the best measure is a listening test, while some researchers have used visual comparison of the spectra. Both these measures are, however, subjective and thus repeatable results can only be obtained by taking the statistical average of a large number of tests. The measure used here is a ratio of the variance of the input spectrum (or the reconstructed spectrum since the gain factor ensures these are the same) to the variance of the difference between the input and the reconstructed spectra. This is called the Signal to Noise Ratio.

When reconstruction was performed using an impulse train excitation, in Tests 6 and 7,

it was found that the pole-zero representation still performed better than the all-pole case, for the SNR measure described above. As explained in Chapter 5, this result is most pronounced for unclipped input waveforms. That the pole-zero improvement can be found in the reconstructed spectrum indicates that the shape of the residual is not an important factor here. It does not necessarily follow that these same results will occur in subjective listening tests. A pole-zero representation minimizes the *numerical* difference between the input and reconstructed waveforms but studies have shown [Ata85] that, in subjective listening tests, the human ear is more responsive to spectral peaks and the overall spectral balance than it is to the spectral valleys. Zeros represent these valleys and so it is quite conceivable that even though the SNR has been reduced, this will not result in improved subjective results. Another way of looking at this is to recall from the results presented above that the spectral valleys result primarily from the glottal pulse. This pulse is quasi-periodic and so it does not provide the listener with much new information whereas the spectral peaks result mostly from the shape of the vocal tract - the changing of which is what differentiates the phonetic sounds of our language. Thus information about the vocal tract is more salient to discrimination between language sounds: the measures used in this study treat the speech signal without regard to its function as a complex encoder of meaning. Determination of whether pole-zero gains could be extended as far as subjective listening tests was, again, beyond the scope of this study and no evidence has been presented in the engineering literature to prove the matter one way or the other.

Finally, this study takes only passing regard of the reason why all-pole prediction is the widely preferred method at this time - that the all-pole coefficients can be found quickly and accurately by the solution of linear equations, while pole-zero and all-zero

coefficients must be found by the solution of non-linear equations. This is done in this study by the numerical method of steepest descent. The same procedure is followed for the all-pole case so that valid comparisons can be made and, indeed, the gains made by exchanging some poles for zeros more than compensated, in some cases, for the losses caused by the inaccuracy with which the coefficients could be found. (These losses should be reduced when quantization of the coefficients is included because in that case there is only a set limit of accuracy *possible*.) Comparisons of the all-pole, pole-zero and all-zero systems are done at this lower level of accuracy under the assumption that when more accurate methods are developed to find the pole-zero coefficients, the relationships found at the level of this study will continue to hold. This study has shown that *the gains are there to be made*.

The tracing of the important valleys in the speech spectrum to the glottal pulse also suggests a different approach to the modelling of the speech system. If the synthesis filter were to be excited by a pseudo-glottal pulse instead of an impulse, then the LPC analysis need only be responsible for modelling the vocal tract which, it has been shown in this study, can be accurately done by an all-pole filter. This has been attempted, although with limited success [Hed84]. The problem is that minimization of the residual by the analysis filter drives the residual to an impulse, not a glottal pulse and so it is incorrect to now excite the synthesis filter with a glottal pulse. The spectrum of the glottal pulse must be removed from the speech spectrum (by deconvolution in the time-domain) before the signal reaches the LPC analysis filter, and then the addition of this spectrum by exciting the synthesis filter with a glottal pulse is logical. Such a scheme avoids the need to include zeros in the LPC filters and deserves further attention.

In summary, this study shows that improved performance is possible by the inclusion of zeros in linear predictive representations of speech and describes the limits and possible counterbalancing effects of this pole-zero representation. An optimal combination of one pole and eleven zeros is shown for the twelve coefficient system. The effect is traced primarily to the glottal pulse and a scheme for the representation of speech based on this cause is suggested.

References

- [Ata70] B.S. Atal and M.R. Schroeder, "Adaptive Predictive Coding of Speech Signals", *Bell Sys. Tech. Journal*, v10 39 no.8 pp 1973-1986, Oct 1970
- [Ata71] B.S. Atal and S.L. Hanauer, "Speech Analysis and Synthesis by Linear Prediction of the Speech Wave" *J. Acoust. Soc. Am.*, Vol 50, pp 637-655, 1971
- [Ata78] B.S. Atal and M.R. Schroeder, "Linear Predictive Analysis of Speech Based on a Pole-Zero Representation", *J. Acoust. Soc. Am.*, Vol 64 no.5, Nov 1978
- [Ata85] B.S. Atal, "Linear Predictive coding of Speech" in *Computer Speech Processing* (ed. F. Fallside), Prentice-Hall, 1985
- [Fla70] J.L. Flanagan, C.H. Coker, L.R. Rabiner, R.W. Schafer and N. Umeda, "Synthetic Voices for Computers" *IEEE Spectrum*, Vol 7 no.10, pp 22-45, 1970
- [Fri83] B. Friedlander, "Efficient algorithm for ARMA spectral estimation", *IEE Proc.* 130, Part F, pp 195-201, 1983
- [Hed84] P. Hedelin, "A Glottal LPC-Vocoder", *International Conference on Acoustics, Speech and Signal Processing (ICASSP-84)*, San Diego, March 1984
- [Hol88] C.S. Holzinger, private communication, 1988
- [Hon82] M. Honig and D. Messerschmitt, "Comparison of Adaptive Linear Prediction Algorithms in ADPCM", *IEEE Trans. Communications*, vol COM-30 no.7, July 1982
- [Jay84] N.S. Jayant and P. Noll, *Digital Coding of Waveforms: Principles and Applications to Speech and Video*, Prentice-Hall, 1984
- [Koo86] B. Koo and J.D. Gibson, "Experimental comparison of all-pole, all-zero and pole-zero predictors for ADPCM speech coding", *IEEE Trans. Comm.*, Vol COM-34 pp285-290, 1986

- [Kre83] E. Kreyzig, *Advanced Engineering Mathematics*, John Wiley & Sons, 1983
- [Mak77] J. Makhoul, "Stable and Efficient Lattice Methods for Linear Prediction" *IEEE Trans. Acoustics, Speech and Signal Proc.*, Vol ASSP-25 no.5, pp 423-428, October 1977
- [Mar73] J.D. Markel and A.H. Gray Jr., "On Autocorrelation Equations as Applied to Speech Analysis" *IEEE Trans. on Audio and Electroacoustics*, Vol AU-21, pp69-79, April 1973
- [Mar76] J.D. Markel and A.H. Gray, "*Linear Prediction of Speech*", Springer-Verlag, 1976
- [Mor82] H. Morikawa and H. Fujisaki, "Adaptive Analysis of Speech Based on a Pole-Zero Representation", *IEEE Trans. Acoustics, Speech and Signal Proc.*, Vol ASSP-30 no. 1, 1982
- [Miy86] Y. Miyanaga, N. Miki and N. Nagai, "Adaptive identification of a time-varying ARMA model and its evaluation", *IEEE Trans. Acoustics, Speech and Signal Proc.*, Vol ASSP-34 pp 423-433, 1986
- [Opp75] A.V. Oppenheim and R.W. Schaffer, *Digital Signal Processing*, Prentice-Hall, 1975
- [O'Sh87] D. O'Shaughnessy, *Speech Communication - Man and Machine*, Addison-Wesley, 1987
- [Rab78] L.R. Rabiner and R.W. Schaffer, *Digital Processing of Speech Signals*, Prentice-Hall, 1978
- [Ste77] K. Steiglitz "On the Simultaneous Estimation of Poles and Zeros in Speech Analysis", *IEEE Trans. Acoustics, Speech and Signal Proc.*, vol ASSP-25 no.3, June 1977
- [Wid76] B. Widrow, J.M. McCool, M.G. Larimore and C.R. Johnson, "Stationary and

Nonstationary Learning Characteristics of the LMS Adaptive Filter", *Proc.*

IEEE pp1151-1162, August 1976

[Wil67] D.J. Wilde and C.S. Beightler, "*Foundations of Optimization*", Prentice-Hall, 1967

[Wol86] C. Woloszynski, "*Voice Sampling Hardware and Software Design Report & User's Guide*", unpublished, 1986

Appendix A

LPC Simulation Results showing Performance against Number of Iterations

VOICED SAMPLE				
Data Points	Poles	Zeros	Iterations	Gain(dB)
128	12	0	1	10.499
128	12	0	2	11.539
128	12	0	3	12.141
128	12	0	4	12.625
128	12	0	5	12.985
128	12	0	6	13.254
128	12	0	7	13.462
128	12	0	8	13.626
128	12	0	9	13.760
128	12	0	10	13.871
128	12	0	11	13.965
128	12	0	12	14.045
128	12	0	13	14.116
128	12	0	14	14.177
128	12	0	15	14.232

VOICED SAMPLE				
Data Points	Poles	Zeros	Iterations	Gain(dB)
128	6	6	1	10.834
128	6	6	2	13.903
128	6	6	3	14.383
128	6	6	4	14.636
128	6	6	5	14.807
128	6	6	6	14.936
128	6	6	7	15.039
128	6	6	8	15.124
128	6	6	9	15.195
128	6	6	10	15.257
128	6	6	11	15.311
128	6	6	12	15.359
128	6	6	13	15.402
128	6	6	14	15.440
128	6	6	15	15.476

VOICED SAMPLE				
Data Points	Poles	Zeros	Iterations	Gain(dB)
128	0	12	1	8.7037
128	0	12	2	12.379
128	0	12	3	13.058
128	0	12	4	13.424
128	0	12	5	13.663
128	0	12	6	13.836
128	0	12	7	13.969
128	0	12	8	14.075
128	0	12	9	14.162
128	0	12	10	14.235
128	0	12	11	14.297
128	0	12	12	14.351
128	0	12	13	14.398
128	0	12	14	14.440
128	0	12	15	14.478

UNVOICED SAMPLE				
Data Points	Poles	Zeros	Iterations	Gain(dB)
128	12	0	1	1.891
128	12	0	2	2.968
128	12	0	3	3.291
128	12	0	4	3.491
128	12	0	5	3.619
128	12	0	6	3.708
128	12	0	7	3.772
128	12	0	8	3.821
128	12	0	9	3.859
128	12	0	10	3.890
128	12	0	11	3.915
128	12	0	12	3.936
128	12	0	13	3.954
128	12	0	14	3.969
128	12	0	15	3.982

UNVOICED SAMPLE				
Data Points	Poles	Zeros	Iterations	Gain(dB)
128	6	6	1	2.365
128	6	6	2	3.029
128	6	6	3	3.315
128	6	6	4	3.452
128	6	6	5	3.534
128	6	6	6	3.589
128	6	6	7	3.629
128	6	6	8	3.658
128	6	6	9	3.682
128	6	6	10	3.701
128	6	6	11	3.716
128	6	6	12	3.730
128	6	6	13	3.741
128	6	6	14	3.751
128	6	6	15	3.759

UNVOICED SAMPLE				
Data Points	Poles	Zeros	Iterations	Gain(dB)
128	12	0	1	1.600
128	12	0	2	2.522
128	12	0	3	3.265
128	12	0	4	3.415
128	12	0	5	3.509
128	12	0	6	3.570
128	12	0	7	3.612
128	12	0	8	3.644
128	12	0	9	3.668
128	12	0	10	3.687
128	12	0	11	3.703
128	12	0	12	3.716
128	12	0	13	3.726
128	12	0	14	3.735
128	12	0	15	3.743

Appendix B

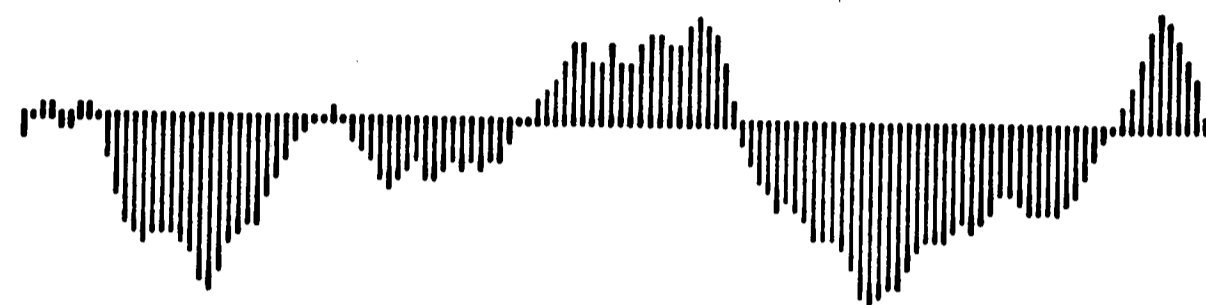
Plots of the Real and Synthetic Speech Samples used in this study

VOICED SAMPLE 1

Variance = 4.7×10^4

Zero-crossing rate = 8%

Clipping = 0%

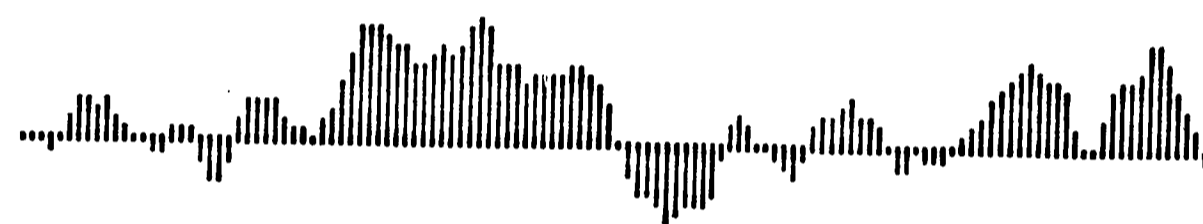


VOICED SAMPLE 2

Variance = 2.1×10^4

Zero-crossing rate = 11%

Clipping = 0%

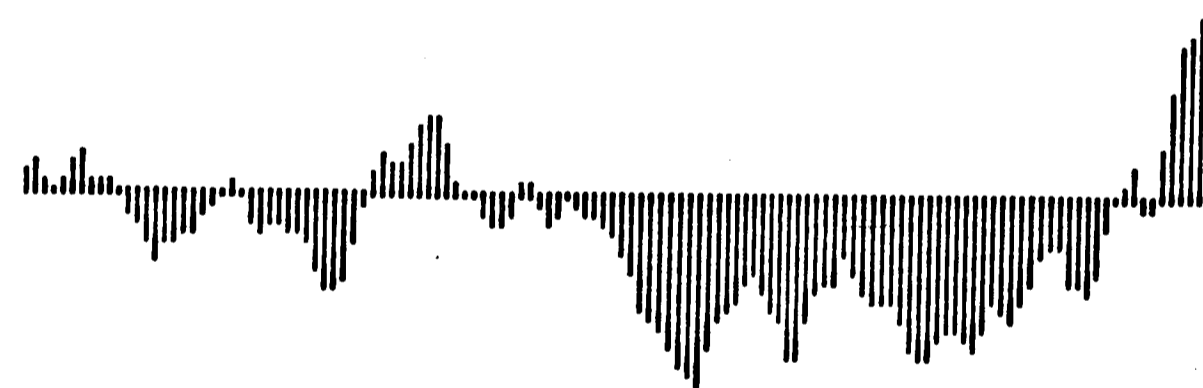


VOICED SAMPLE 3

Variance = 5.1×10^4

Zero-crossing rate = 9%

Clipping = 0%

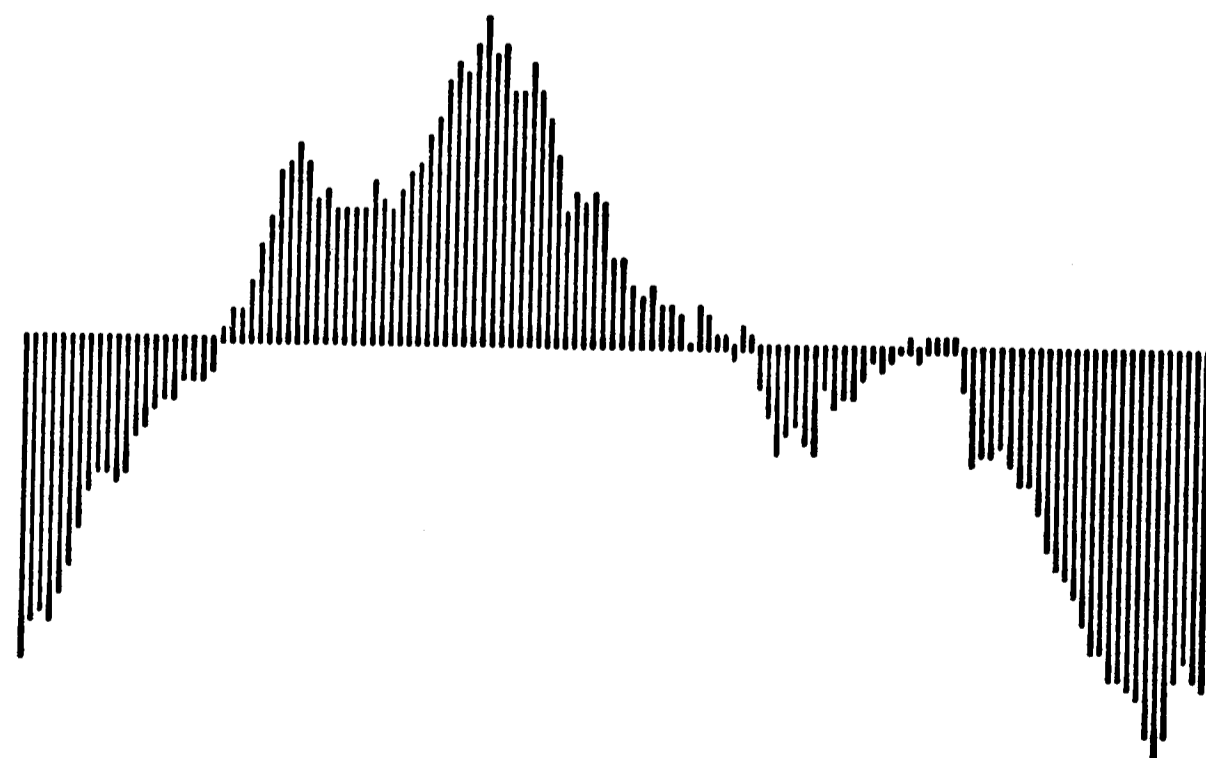


VOICED SAMPLE 4

Variance = 2.3×10^5

Zero-crossing rate = 7%

Clipping = 0%

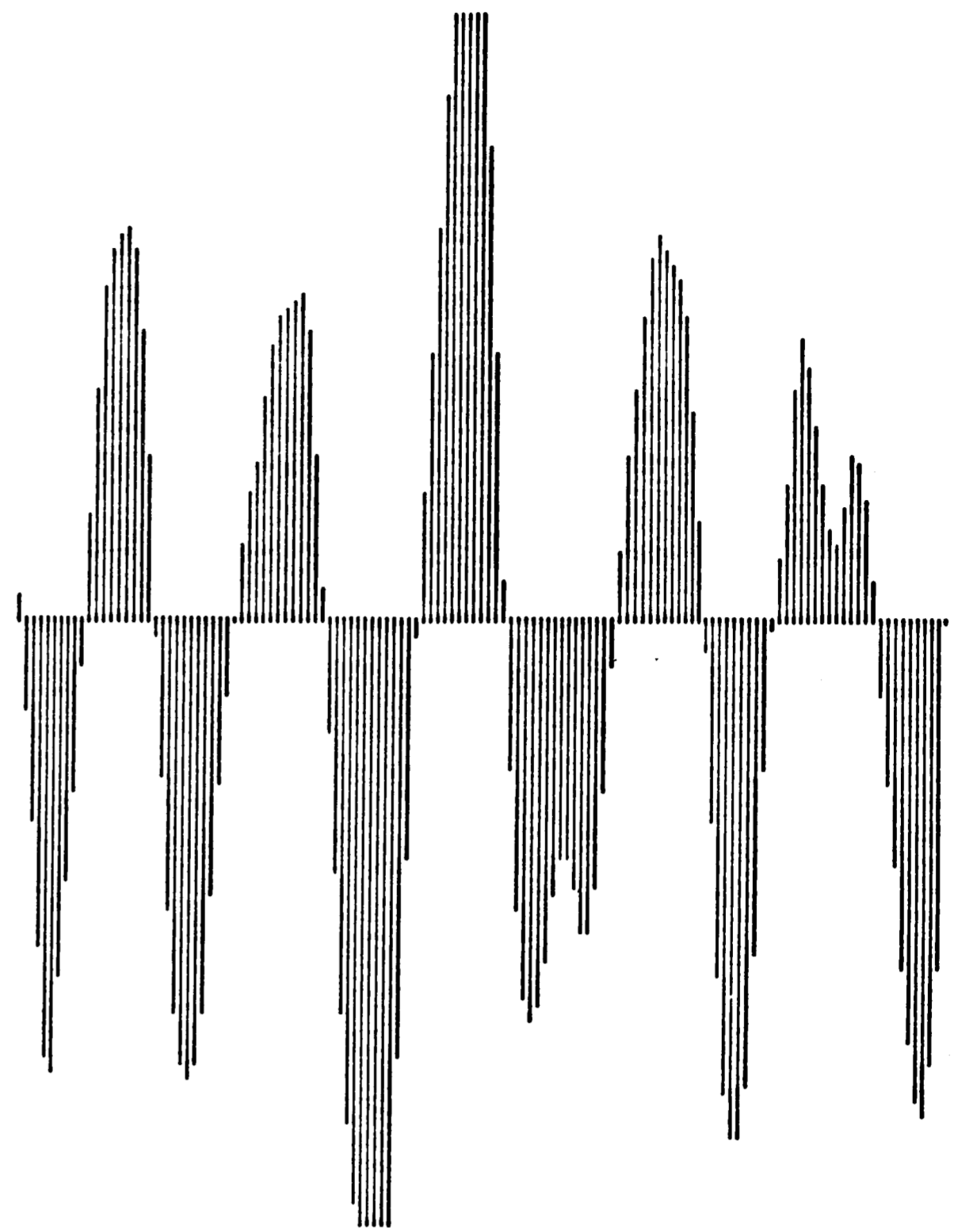


VOICED SAMPLE 5

Variance = 1.2×10^6

Zero-crossing rate = 9%

Clipping = 8%

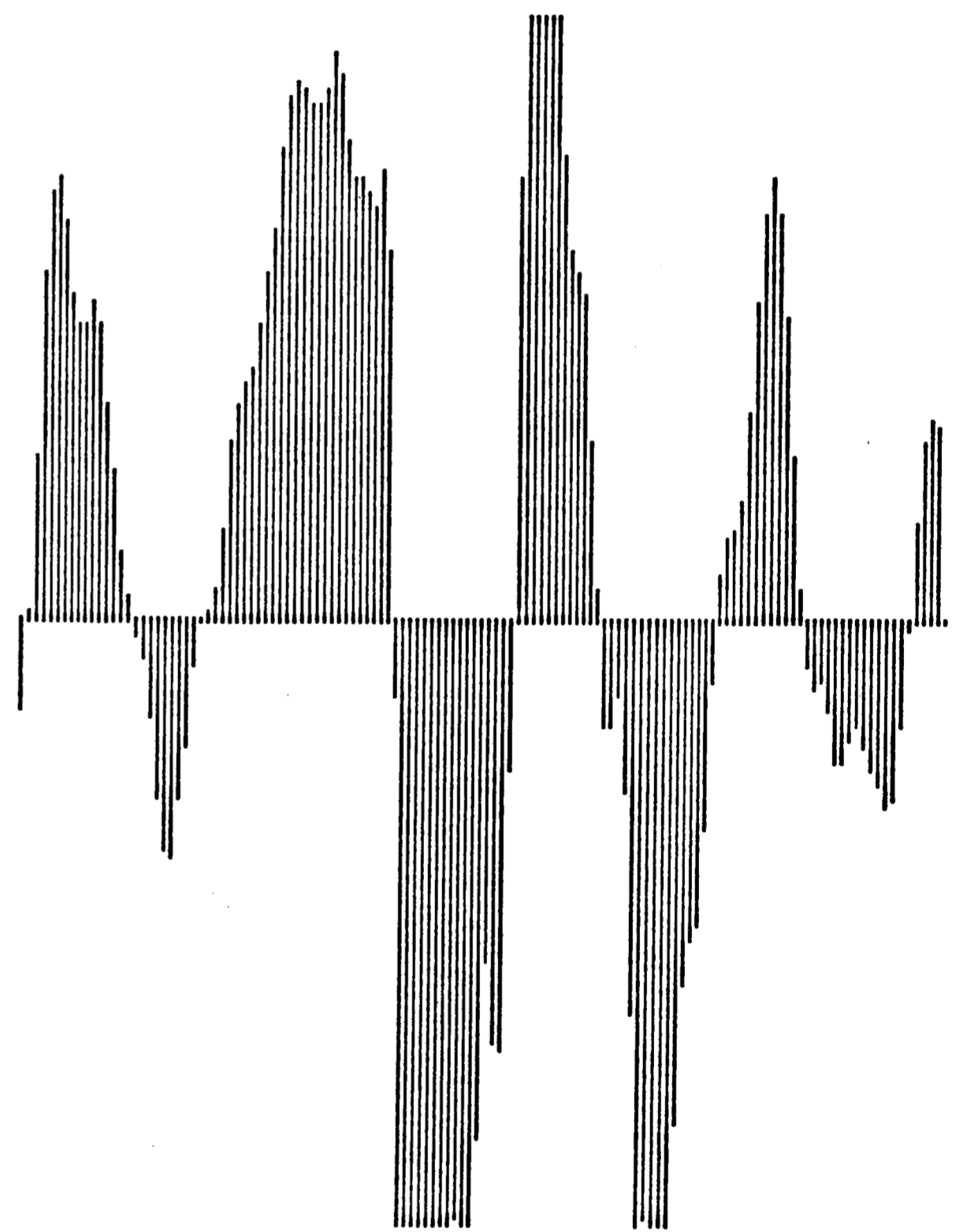


VOICED SAMPLE 6

Variance = 1.5×10^6

Zero-crossing rate = 7%

Clipping = 15%

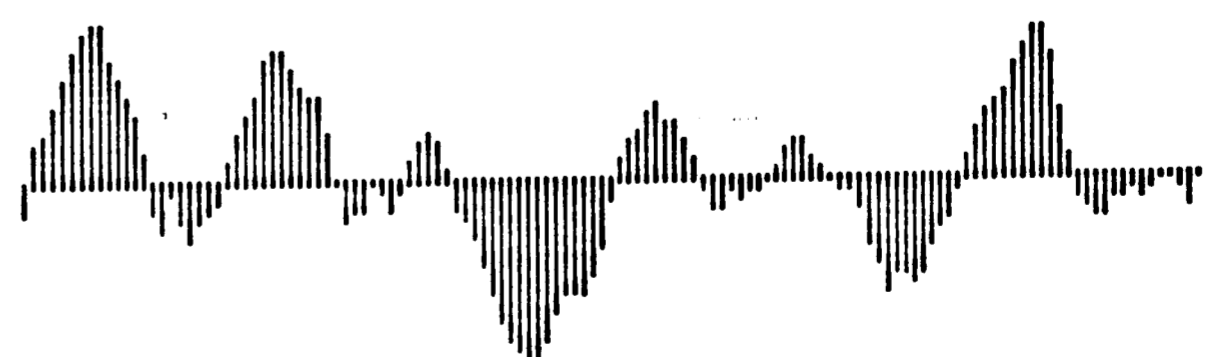


VOICED SAMPLE 7

Variance = 4.1×10^4

Zero-crossing rate = 11%

Clipping = 0%

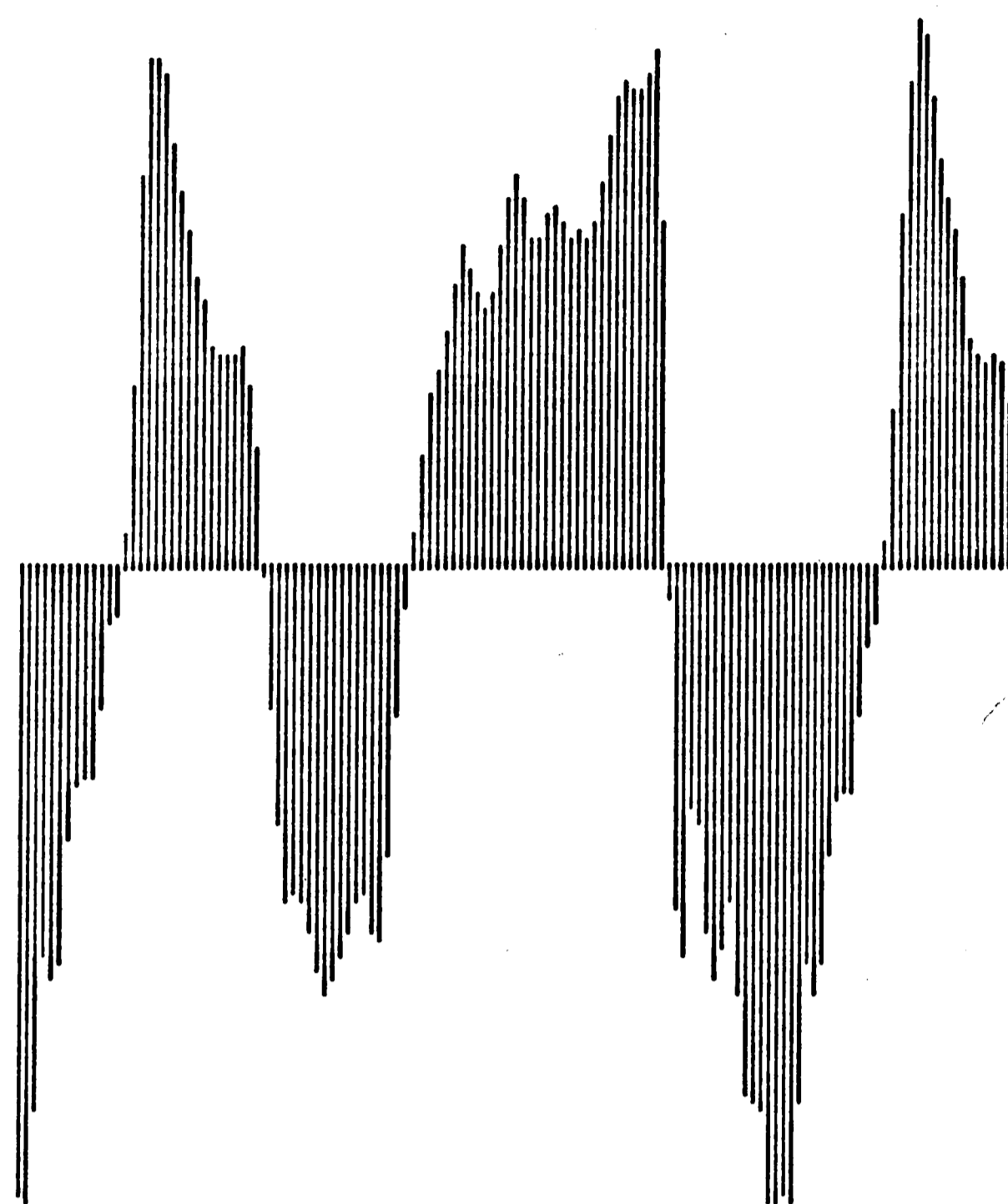


VOICED SAMPLE 8

Variance = 1.2×10^6

Zero-crossing rate = 4%

Clipping = 3%

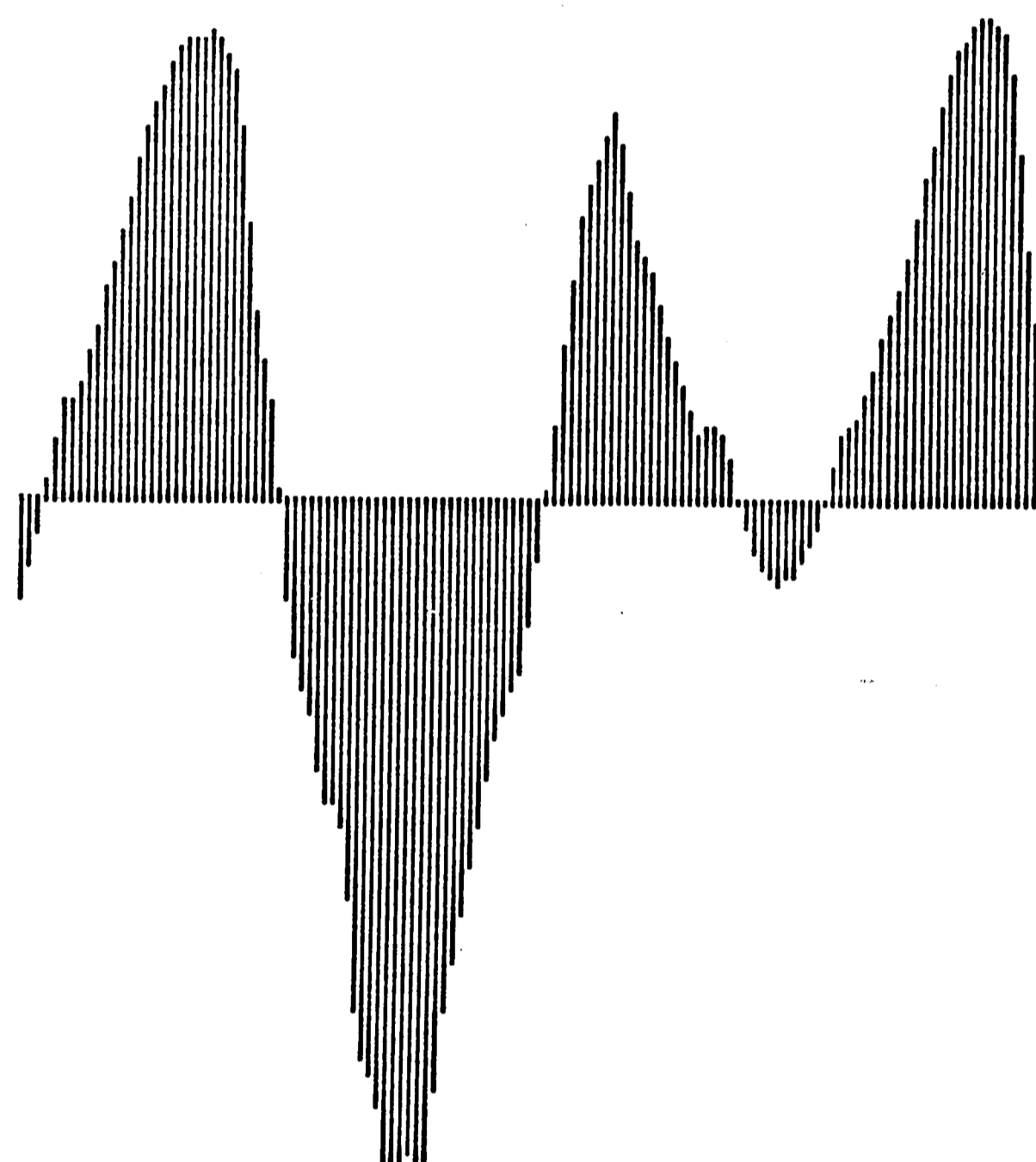


VOICED SAMPLE 9

Variance = 9.8×10^5

Zero-crossing rate = 4%

Clipping = 3%

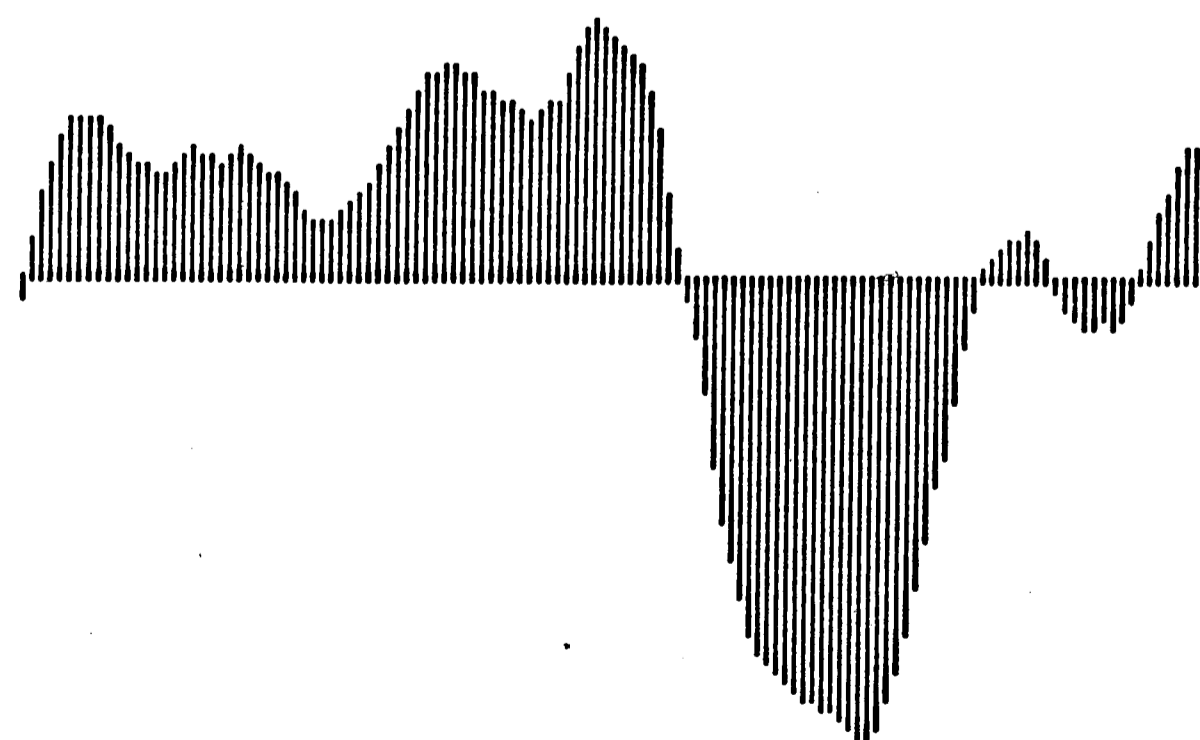


VOICED SAMPLE 10

Variance = 3.0×10^5

Zero-crossing rate = 4%

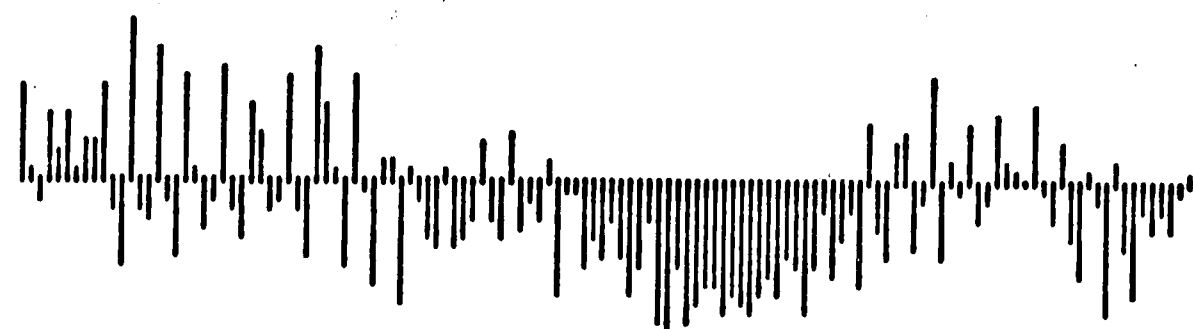
Clipping = 0%



UNVOICED SAMPLE 1

Variance = 3.6×10^4

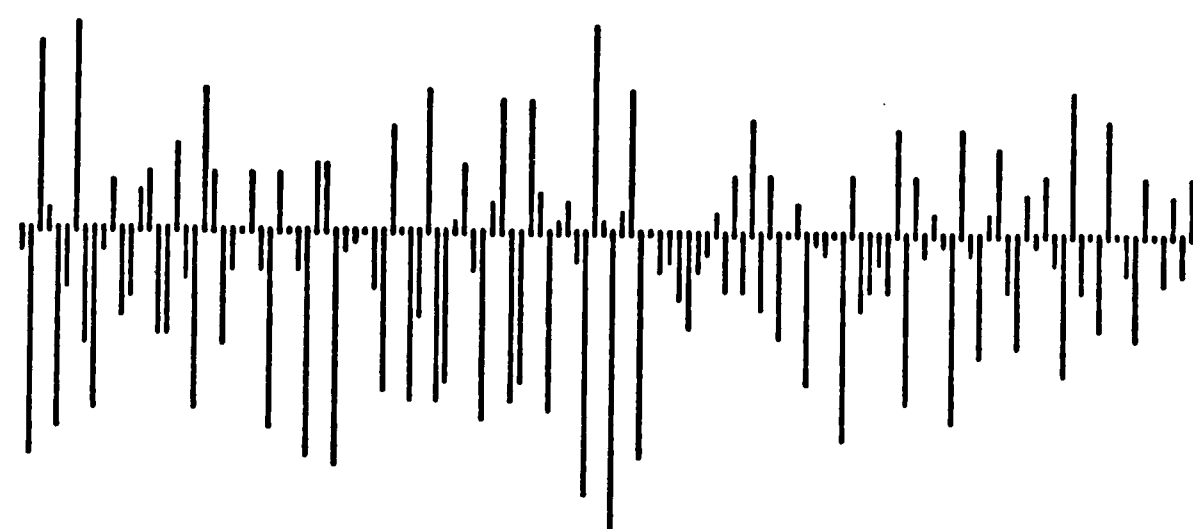
Zero-crossing rate = 39%



UNVOICED SAMPLE 2

Variance = 7.9×10^4

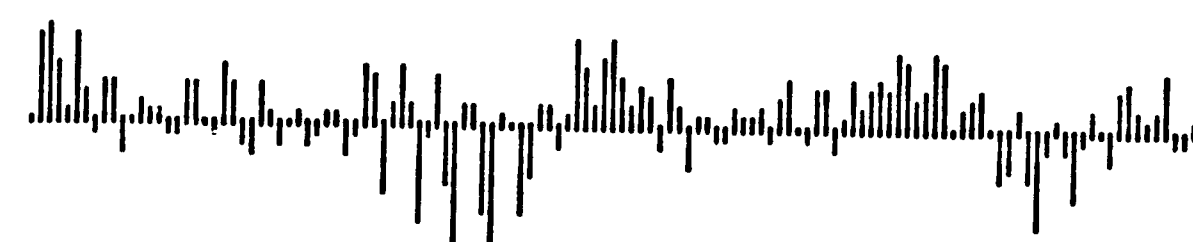
Zero-crossing rate = 55%



UNVOICED SAMPLE 3

Variance = 1.1×10^4

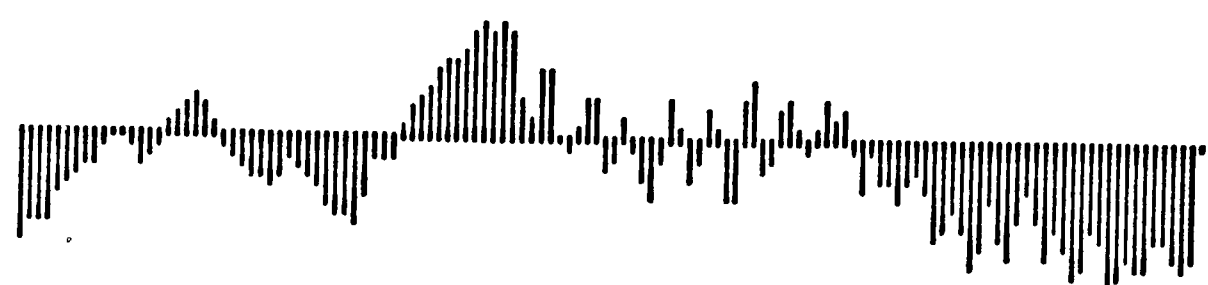
Zero-crossing rate = 40%



UNVOICED SAMPLE 4

Variance = 2.9×10^4

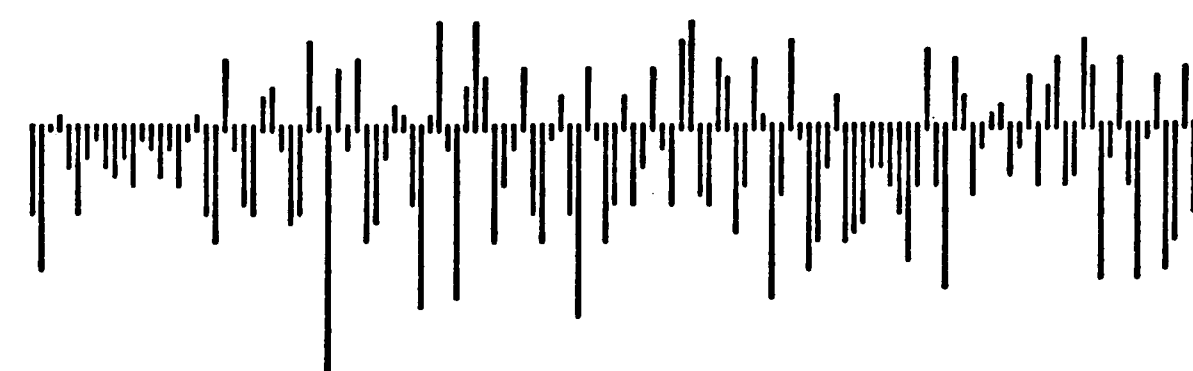
Zero-crossing rate = 14%



UNVOICED SAMPLE 5

Variance = 4.4×10^4

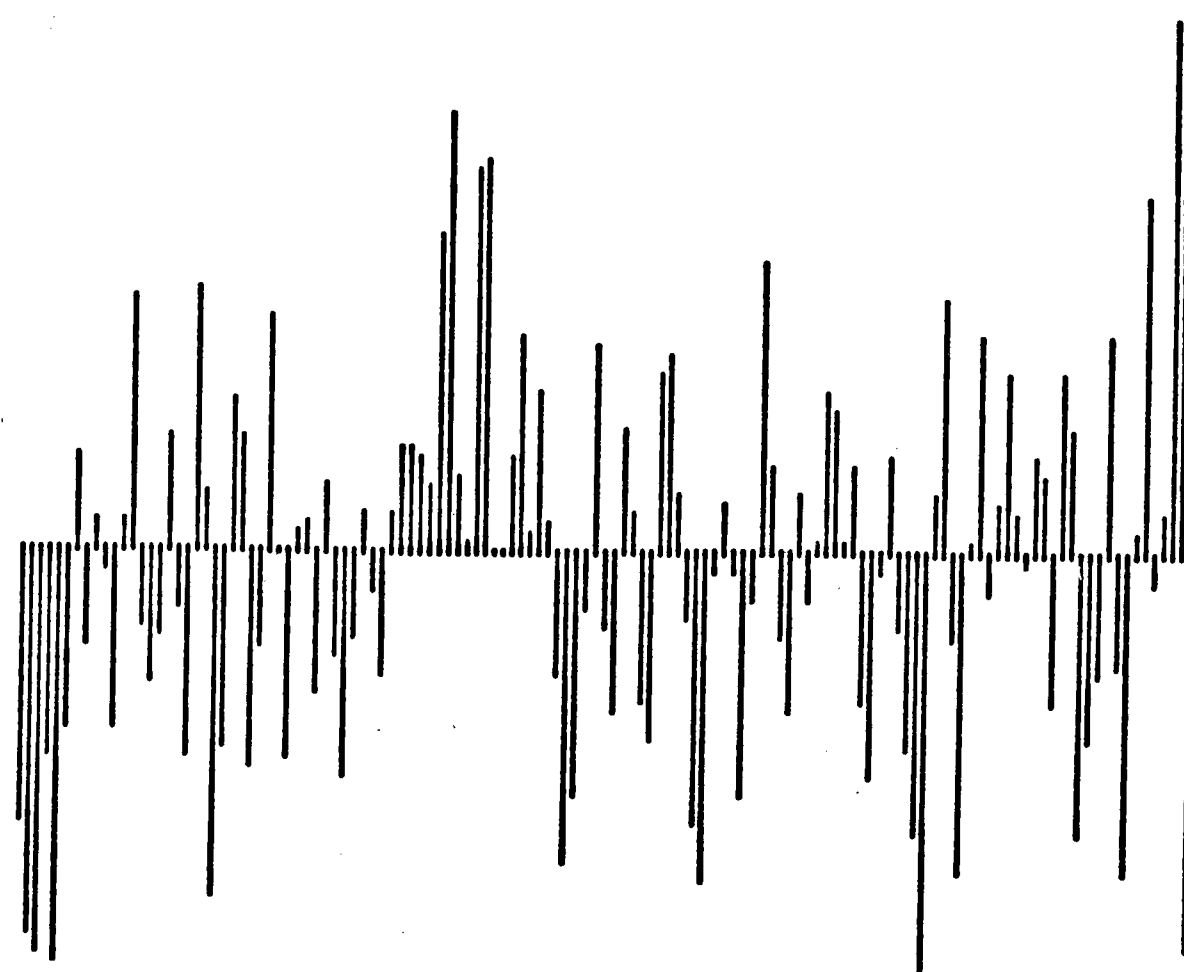
Zero-crossing rate = 45%



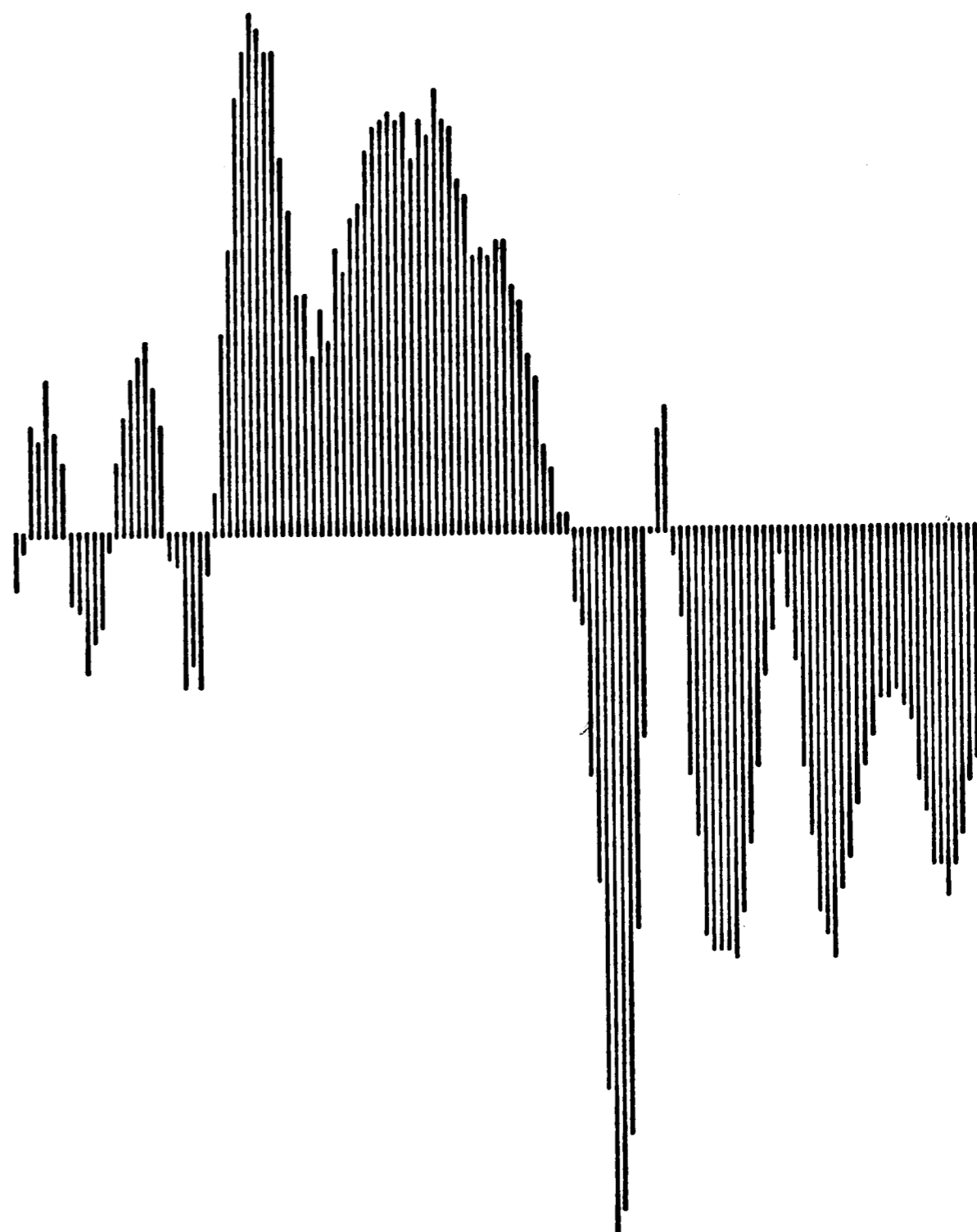
UNVOICED SAMPLE 6

Variance = 2.5×10^5

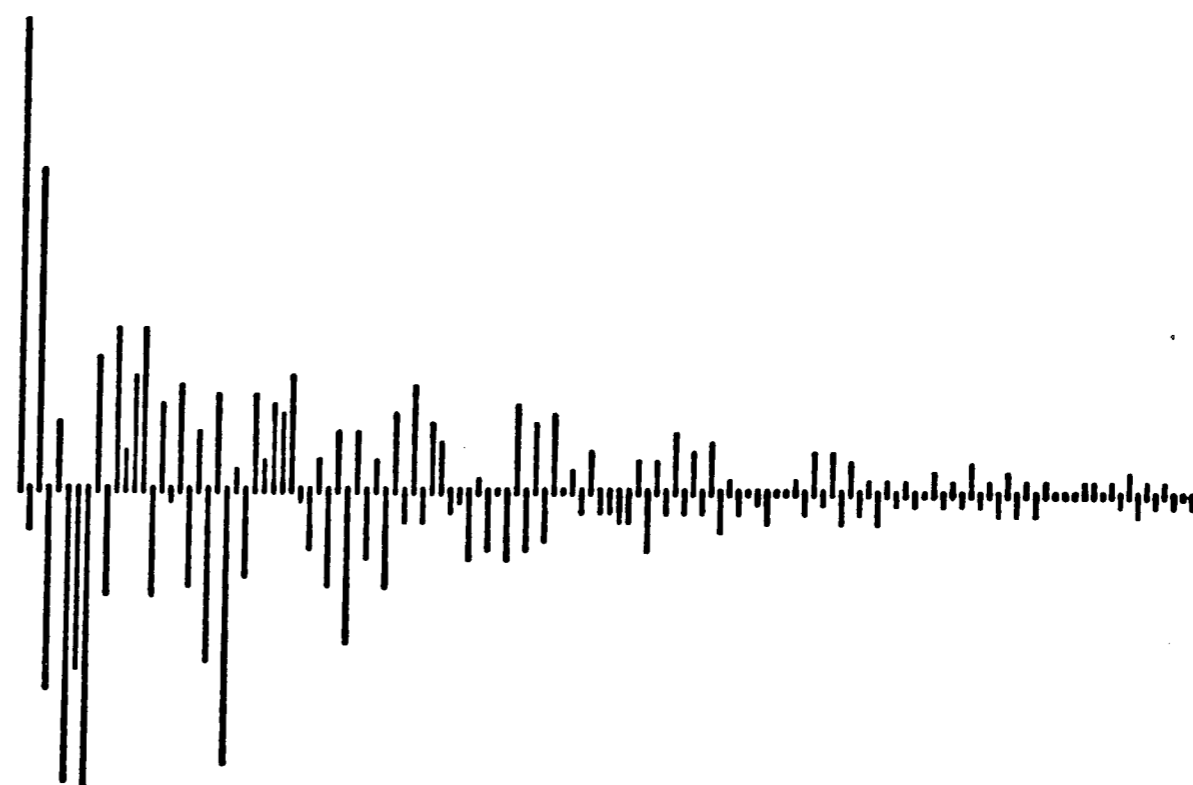
Zero-crossing rate = 44%



SYNTHETIC SPEECH (GLOTTAL)



SYNTHETIC SPEECH (IMPULSE)



Appendix C

Test One

VOICED SAMPLE ONE				
Data Points	Poles	Zeros	Iterations	Gain(dB)
128	12	0	6	12.577
128	11	1	6	13.096
128	10	2	6	13.619
128	9	3	6	13.931
128	8	4	6	14.211
128	7	5	6	14.563
128	6	6	6	14.924
128	5	7	6	15.196
128	4	8	6	15.392
128	3	9	6	15.445
128	2	10	6	15.552
128	1	11	6	16.642
128	0	12	6	13.904

VOICED SAMPLE TWO				
Data Points	Poles	Zeros	Iterations	Gain(dB)
128	12	0	6	11.058
128	11	1	6	11.402
128	10	2	6	11.628
128	9	3	6	11.858
128	8	4	6	12.133
128	7	5	6	12.283
128	6	6	6	12.413
128	5	7	6	12.576
128	4	8	6	12.749
128	3	9	6	12.952
128	2	10	6	13.004
128	1	11	6	13.331
128	0	12	6	11.667

VOICED SAMPLE THREE				
Data Points	Poles	Zeros	Iterations	Gain(dB)
128	12	0	6	11.421
128	11	1	6	12.035
128	10	2	6	12.517
128	9	3	6	12.883
128	8	4	6	13.172
128	7	5	6	13.353
128	6	6	6	13.523
128	5	7	6	13.692
128	4	8	6	13.818
128	3	9	6	13.913
128	2	10	6	13.940
128	1	11	6	14.542
128	0	12	6	12.772

VOICED SAMPLE FOUR				
Data Points	Poles	Zeros	Iterations	Gain(dB)
128	12	0	6	13.018
128	11	1	6	13.137
128	10	2	6	13.245
128	9	3	6	13.352
128	8	4	6	13.415
128	7	5	6	13.502
128	6	6	6	13.555
128	5	7	6	13.589
128	4	8	6	13.651
128	3	9	6	13.706
128	2	10	6	13.778
128	1	11	6	13.761
128	0	12	6	12.489

VOICED SAMPLE FIVE				
Data Points	Poles	Zeros	Iterations	Gain(dB)
128	12	0	6	14.720
128	11	1	6	14.850\
128	10	2	6	14.907
128	9	3	6	14.965
128	8	4	6	15.197
128	7	5	6	15.568
128	6	6	6	15.958
128	5	7	6	16.539
128	4	8	6	17.406
128	3	9	6	17.588
128	2	10	6	16.458
128	1	11	6	17.033
128	0	12	6	13.489

VOICED SAMPLE SIX				
Data Points	Poles	Zeros	Iterations	Gain(dB)
128	12	0	6	9.9376
128	11	1	6	10.598
128	10	2	6	11.110
128	9	3	6	11.393
128	8	4	6	11.596
128	7	5	6	11.895
128	6	6	6	12.191
128	5	7	6	12.344
128	4	8	6	12.463
128	3	9	6	12.737
128	2	10	6	12.856
128	1	11	6	12.977
128	0	12	6	11.422

VOICED SAMPLE SEVEN				
Data Points	Poles	Zeros	Iterations	Gain(dB)
128	12	0	6	11.512
128	11	1	6	11.944
128	10	2	6	12.310
128	9	3	6	12.479
128	8	4	6	12.571
128	7	5	6	12.736
128	6	6	6	12.902
128	5	7	6	13.084
128	4	8	6	13.170
128	3	9	6	13.311
128	2	10	6	13.320
128	1	11	6	13.633
128	0	12	6	12.076

VOICED SAMPLE EIGHT				
Data Points	Poles	Zeros	Iterations	Gain(dB)
128	12	0	6	11.337
128	11	1	6	11.566
128	10	2	6	11.677
128	9	3	6	11.764
128	8	4	6	11.883
128	7	5	6	12.055
128	6	6	6	12.224
128	5	7	6	12.279
128	4	8	6	12.368
128	3	9	6	12.472
128	2	10	6	12.488
128	1	11	6	12.695
128	0	12	6	11.658

VOICED SAMPLE NINE				
Data Points	Poles	Zeros	Iterations	Gain(dB)
128	12	0	6	15.657
128	11	1	6	16.249
128	10	2	6	17.133
128	9	3	6	17.958
128	8	4	6	18.763
128	7	5	6	19.379
128	6	6	6	19.529
128	5	7	6	19.134
128	4	8	6	18.748
128	3	9	6	18.714
128	2	10	6	19.415
128	1	11	6	21.677
128	0	12	6	15.777

VOICED SAMPLE TEN				
Data Points	Poles	Zeros	Iterations	Gain(dB)
128	12	0	6	17.679
128	11	1	6	17.792
128	10	2	6	18.288
128	9	3	6	18.836
128	8	4	6	19.550
128	7	5	6	20.209
128	6	6	6	20.480
128	5	7	6	20.084
128	4	8	6	19.491
128	3	9	6	19.431
128	2	10	6	20.453
128	1	11	6	23.275
128	0	12	6	16.164

Durbin's Recursive Method on Voiced Samples 1-10

DURBIN'S RECURSIVE METHOD				
Data Points	Poles	Zeros	Sample No.	Gain(dB)
128	12	12	1	17.543
128	12	12	2	14.014
128	12	12	3	11.586
128	12	12	4	11.835
128	12	12	5	17.108
128	12	12	6	13.580
128	12	12	7	13.739
128	12	12	8	12.913
128	12	12	9	22.030
128	12	12	10	21.520

Test 2

UNVOICED SAMPLE ONE				
Data Points	Poles	Zeros	Iterations	Gain(dB)
128	12	0	6	4.631
128	11	1	6	4.574
128	10	2	6	4.597
128	9	3	6	4.675
128	8	4	6	4.672
128	7	5	6	4.711
128	6	6	6	4.749
128	5	7	6	4.809
128	4	8	6	4.838
128	3	9	6	4.969
128	2	10	6	4.325
128	1	11	6	3.970
128	0	12	6	4.030

UNVOICED SAMPLE TWO				
Data Points	Poles	Zeros	Iterations	Gain(dB)
128	12	0	6	3.773
128	11	1	6	3.803
128	10	2	6	3.866
128	9	3	6	3.899
128	8	4	6	3.651
128	7	5	6	3.860
128	6	6	6	3.843
128	5	7	6	3.888
128	4	8	6	3.724
128	3	9	6	3.634
128	2	10	6	3.669
128	1	11	6	3.507
128	0	12	6	3.109

UNVOICED SAMPLE THREE				
Data Points	Poles	Zeros	Iterations	Gain(dB)
128	12	0	6	5.379
128	11	1	6	5.368
128	10	2	6	5.418
128	9	3	6	5.476
128	8	4	6	5.534
128	7	5	6	5.527
128	6	6	6	5.489
128	5	7	6	5.378
128	4	8	6	5.355
128	3	9	6	5.515
128	2	10	6	4.582
128	1	11	6	5.315
128	0	12	6	5.319

UNVOICED SAMPLE FOUR				
Data Points	Poles	Zeros	Iterations	Gain(dB)
128	12	0	6	8.636
128	11	1	6	8.858
128	10	2	6	9.000
128	9	3	6	9.157
128	8	4	6	9.234
128	7	5	6	9.247
128	6	6	6	9.345
128	5	7	6	9.344
128	4	8	6	9.456
128	3	9	6	9.584
128	2	10	6	9.551
128	1	11	6	9.465
128	0	12	6	9.139

UNVOICED SAMPLE FIVE				
Data Points	Poles	Zeros	Iterations	Gain(dB)
128	12	0	6	3.979
128	11	1	6	4.033
128	10	2	6	4.064
128	9	3	6	3.867
128	8	4	6	3.770
128	7	5	6	3.700
128	6	6	6	3.788
128	5	7	6	3.717
128	4	8	6	3.885
128	3	9	6	3.726
128	2	10	6	3.298
128	1	11	6	3.527
128	0	12	6	3.697

UNVOICED SAMPLE SIX				
Data Points	Poles	Zeros	Iterations	Gain(dB)
128	12	0	6	2.069
128	11	1	6	2.087
128	10	2	6	2.078
128	9	3	6	2.071
128	8	4	6	2.046
128	7	5	6	2.004
128	6	6	6	2.008
128	5	7	6	2.004
128	4	8	6	1.954
128	3	9	6	2.106
128	2	10	6	2.128
128	1	11	6	2.225
128	0	12	6	2.242

Test 3

SYNTHETIC SPEECH (GLOTTAL)				
Data Points	Poles	Zeros	Iterations	Gain(dB)
128	12	0	6	13.026
128	11	1	6	13.299
128	10	2	6	13.867
128	9	3	6	14.525
128	8	4	6	15.225
128	7	5	6	15.366
128	6	6	6	15.111
128	5	7	6	15.215
128	4	8	6	15.609
128	3	9	6	15.751
128	2	10	6	15.421
128	1	11	6	15.951
128	0	12	6	12.309

Test 4

SYNTHETIC SPEECH (IMPULSE)				
Data Points	Poles	Zeros	Iterations	Gain(dB)
128	12	0	6	6.538
128	11	1	6	6.587
128	10	2	6	6.306
128	9	3	6	3.315
128	8	4	6	3.176
128	7	5	6	3.306
128	6	6	6	3.267
128	5	7	6	3.382
128	4	8	6	3.938
128	3	9	6	3.625
128	2	10	6	3.865
128	1	11	6	3.510
128	0	12	6	3.538

Test 5

SYNTHETIC SPEECH (IMPULSE)				
Data Points	Poles	Zeros	Iterations	Gain(dB)
128	12	0	6	6.538
128	11	0	6	6.585
128	10	0	6	6.582
128	9	0	6	3.299
128	8	0	6	3.261
128	7	0	6	3.279
128	6	0	6	3.187
128	5	0	6	2.804
128	4	0	6	2.659
128	3	0	6	2.284
128	2	0	6	1.859
128	1	0	6	0.235
128	0	0	6	0.000

Test 6

VOICED SAMPLE ONE				
Data Points	Poles	Zeros	Iterations	Gain(dB)
128	6	0	6	13.628
128	5	1	6	14.311
128	4	2	6	14.620
128	3	3	6	14.574
128	2	4	6	14.667
128	1	5	6	16.045
128	0	6	6	11.899

VOICED SAMPLE TWO				
Data Points	Poles	Zeros	Iterations	Gain(dB)
128	6	0	6	11.079
128	5	1	6	11.382
128	4	2	6	11.639
128	3	3	6	12.004
128	2	4	6	12.148
128	1	5	6	12.507
128	0	6	6	10.603

VOICED SAMPLE THREE				
Data Points	Poles	Zeros	Iterations	Gain(dB)
128	6	0	6	12.493
128	5	1	6	12.788
128	4	2	6	13.001
128	3	3	6	13.135
128	2	4	6	13.168
128	1	5	6	13.973
128	0	6	6	10.906

VOICED SAMPLE FOUR				
Data Points	Poles	Zeros	Iterations	Gain(dB)
128	6	0	6	13.122
128	5	1	6	13.216
128	4	2	6	13.300
128	3	3	6	13.369
128	2	4	6	13.523
128	1	5	6	13.799
128	0	6	6	10.995

VOICED SAMPLE FIVE				
Data Points	Poles	Zeros	Iterations	Gain(dB)
128	6	0	6	14.328
128	5	1	6	15.128
128	4	2	6	16.390
128	3	3	6	17.176
128	2	4	6	14.857
128	1	5	6	14.809
128	0	6	6	11.532

VOICED SAMPLE SIX				
Data Points	Poles	Zeros	Iterations	Gain(dB)
128	6	0	6	11.069
128	5	1	6	11.293
128	4	2	6	11.339
128	3	3	6	11.711
128	2	4	6	11.987
128	1	5	6	12.150
128	0	6	6	10.264

VOICED SAMPLE SEVEN				
Data Points	Poles	Zeros	Iterations	Gain(dB)
128	6	0	6	11.817
128	5	1	6	12.162
128	4	2	6	12.491
128	3	3	6	12.685
128	2	4	6	12.506
128	1	5	6	12.745
128	0	6	6	10.711

VOICED SAMPLE EIGHT				
Data Points	Poles	Zeros	Iterations	Gain(dB)
128	6	0	6	11.436
128	5	1	6	11.591
128	4	2	6	11.658
128	3	3	6	11.840
128	2	4	6	11.956
128	1	5	6	12.391
128	0	6	6	10.439

VOICED SAMPLE NINE				
Data Points	Poles	Zeros	Iterations	Gain(dB)
128	6	0	6	18.819
128	5	1	6	18.544
128	4	2	6	17.383
128	3	3	6	16.934
128	2	4	6	17.805
128	1	5	6	20.664
128	0	6	6	13.169

VOICED SAMPLE TEN				
Data Points	Poles	Zeros	Iterations	Gain(dB)
128	6	0	6	18.931
128	5	1	6	18.370
128	4	2	6	17.293
128	3	3	6	17.217
128	2	4	6	18.859
128	1	5	6	21.988
128	0	6	6	13.437

Test Seven

VOICED SAMPLE ONE				
Data Points	Poles	Zeros	Iterations	SNR(dB)
128	12	0	6	4.418
128	11	1	6	4.808
128	10	2	6	5.376
128	9	3	6	5.845
128	8	4	6	6.246
128	7	5	6	6.642
128	6	6	6	7.011
128	5	7	6	7.169
128	4	8	6	7.070
128	3	9	6	6.925
128	2	10	6	6.950
128	1	11	6	7.494
128	0	12	6	4.630

VOICED SAMPLE TWO				
Data Points	Poles	Zeros	Iterations	SNR(dB)
128	12	0	6	6.248
128	11	1	6	6.116
128	10	2	6	6.194
128	9	3	6	6.425
128	8	4	6	6.708
128	7	5	6	6.750
128	6	6	6	6.554
128	5	7	6	6.576
128	4	8	6	6.752
128	3	9	6	6.948
128	2	10	6	7.159
128	1	11	6	7.215
128	0	12	6	4.728

VOICED SAMPLE THREE				
Data Points	Poles	Zeros	Iterations	SNR(dB)
128	12	0	6	4.7176
128	11	1	6	6.3937
128	10	2	6	8.5692
128	9	3	6	10.914
128	8	4	6	12.293
128	7	5	6	12.267
128	6	6	6	11.132
128	5	7	6	10.665
128	4	8	6	10.825
128	3	9	6	10.709
128	2	10	6	10.638
128	1	11	6	10.761
128	0	12	6	6.0186

VOICED SAMPLE FOUR				
Data Points	Poles	Zeros	Iterations	SNR(dB)
128	12	0	6	1.686
128	11	1	6	1.185
128	10	2	6	0.185
128	9	3	6	- 1.047
128	8	4	6	- 2.399
128	7	5	6	- 2.748
128	6	6	6	- 3.893
128	5	7	6	- 4.160
128	4	8	6	- 2.192
128	3	9	6	- 0.591
128	2	10	6	2.004
128	1	11	6	5.253
128	0	12	6	2.584

VOICED SAMPLE FIVE				
Data Points	Poles	Zeros	Iterations	SNR(dB)
128	12	0	6	9.018
128	11	1	6	7.201
128	10	2	6	6.340
128	9	3	6	5.944
128	8	4	6	5.734
128	7	5	6	5.534
128	6	6	6	5.145
128	5	7	6	5.005
128	4	8	6	5.085
128	3	9	6	3.779
128	2	10	6	2.967
128	1	11	6	2.929
128	0	12	6	3.191

VOICED SAMPLE SIX				
Data Points	Poles	Zeros	Iterations	SNR(dB)
128	12	0	6	5.707
128	11	1	6	6.395
128	10	2	6	6.854
128	9	3	6	6.836
128	8	4	6	6.509
128	7	5	6	6.184
128	6	6	6	5.441
128	5	7	6	4.845
128	4	8	6	4.549
128	3	9	6	4.388
128	2	10	6	4.240
128	1	11	6	4.286
128	0	12	6	3.493

VOICED SAMPLE SEVEN				
Data Points	Poles	Zeros	Iterations	SNR(dB)
128	12	0	6	7.693
128	11	1	6	8.144
128	10	2	6	8.580
128	9	3	6	8.552
128	8	4	6	8.309
128	7	5	6	8.450
128	6	6	6	8.911
128	5	7	6	9.124
128	4	8	6	8.394
128	3	9	6	7.476
128	2	10	6	7.079
128	1	11	6	7.308
128	0	12	6	5.549

VOICED SAMPLE EIGHT				
Data Points	Poles	Zeros	Iterations	SNR(dB)
128	12	0	6	8.809
128	11	1	6	8.729
128	10	2	6	7.959
128	9	3	6	7.190
128	8	4	6	6.614
128	7	5	6	6.137
128	6	6	6	5.472
128	5	7	6	4.766
128	4	8	6	4.298
128	3	9	6	4.134
128	2	10	6	3.941
128	1	11	6	4.247
128	0	12	6	4.198

VOICED SAMPLE NINE				
Data Points	Poles	Zeros	Iterations	SNR(dB)
128	12	0	6	3.848
128	11	1	6	3.829
128	10	2	6	4.288
128	9	3	6	5.200
128	8	4	6	6.986
128	7	5	6	9.134
128	6	6	6	8.134
128	5	7	6	6.921
128	4	8	6	6.331
128	3	9	6	6.163
128	2	10	6	6.257
128	1	11	6	7.069
128	0	12	6	3.957

VOICED SAMPLE TEN				
Data Points	Poles	Zeros	Iterations	SNR(dB)
128	12	0	6	5.295
128	11	1	6	6.984
128	10	2	6	7.100
128	9	3	6	6.952
128	8	4	6	6.750
128	7	5	6	6.484
128	6	6	6	6.074
128	5	7	6	5.587
128	4	8	6	5.213
128	3	9	6	4.963
128	2	10	6	4.846
128	1	11	6	5.518
128	0	12	6	3.482

Test Eight

SYNTHETIC SPEECH (GLOTTAL)				
Data Points	Poles	Zeros	Iterations	SNR(dB)
128	12	0	6	- 1.909
128	11	1	6	- 1.330
128	10	2	6	- 0.372
128	9	3	6	0.674
128	8	4	6	2.318
128	7	5	6	3.933
128	6	6	6	4.318
128	5	7	6	4.267
128	4	8	6	4.352
128	3	9	6	4.262
128	2	10	6	3.850
128	1	11	6	4.344
128	0	12	6	2.896

Vita

Christopher Gosnell was born in Durban, South Africa on 25 July 1963 to Dr. Jeremy and Mrs. Marilyn Gosnell, the fourth of six children. He attended high school at Hilton College in South Africa and received the B.Sc (Electrical Engineering) degree with 1st class honours from the University of Cape Town in December 1985. He presently resides in Big Bend, Swaziland.