Elec4621:
Advanced Digital Signal Processing
Chapter 7: Introduction to Linear Prediction

Dr. D. S. Taubman

April 18, 2011

1 Linear Prediction

1.1 Statistical Formulation

Let $X[n]$ denote the elements of a random process, having outcomes $x[n]$. The random process may represent sampled human speech, audio, or some other signal encountered in information communication. The goal of linear prediction is to find a set of weights, $a_1, a_2, \ldots, a_N$, such that

$$\bar{x}[n] = a_1 x[n-1] + a_2 x[n-2] + \cdots + a_N x[n-N]$$

is as close as possible to $x[n]$, for all n, and for all realizations of the underlying random process. We say that $\bar{x}[n]$ is a "prediction" for the actual value of $x[n]$ and we refer to the a_1 through a_N as the linear prediction (LP) coefficients of order N.

Evidently, linear prediction attempts to estimate a future outcome of the random process, based on the previous N observed outcomes. Linear prediction has many interesting applications, some of which we shall describe in Section 2. For the moment, however, we shall focus on the problem of finding the LP coefficients and understanding the nature of the prediction error,

$$e[n] = x[n] - \bar{x}[n]$$

The prediction error sequence is commonly called the "innovations" sequence, and its underlying random process, $E[n]$, is called the "innovations process." This terminology reflects the idea that an ideal predictor would extract all possible information from the samples which have already been seen, $x[n-1], x[n-2], \ldots$, so that the prediction error represents the information in $x[n]$ which is fundamentally new (innovative), which could not have been anticipated.

Since we are interested in finding a single set of coefficients which work for all n and all realizations of the underlying random process, we shall assume that $X[n]$ is stationary (actuall WSS is all we need), and look for the coefficients which minimize the expected (mean) squared prediction error,

$$E\left[\left(X[n]-\bar{X}[n]\right)^2\right]=E\left[\left(X[n]-\sum_{k=1}^{N}a_kX[n-k]\right)^2\right] \tag{1}$$

Differentiating with respect to a_p for each p, and setting each derivative equal to zero, we obtain the so-called "normal equations."

$$\begin{aligned}
0 &= \frac{\partial}{\partial a_p}E\left[\left(X[n]-\sum_{k=1}^{N}a_kX[n-k]\right)^2\right]\\
&= 2E\left[X[n-p]\left(X[n]-\sum_{k=1}^{N}a_kX[n-k]\right)\right]\\
&= 2E\left[X[n-p]X[n]\right]-2\sum_{k=1}^{N}E\left[a_kX[n-p]X[n-k]\right]\\
&= 2R_{XX}[p]-2\sum_{k=1}^{N}a_kR_{XX}[p-k], \qquad 1\le p\le N
\end{aligned} \tag{2}$$

Expressing all N equations in matrix form, and noting that $R_{XX}[m]=R_{XX}[-m]$, we have

$$\underbrace{\begin{pmatrix} R_{XX}[0] & R_{XX}[1] & \cdots & R_{XX}[N-1]\\ R_{XX}[1] & R_{XX}[0] & \cdots & R_{XX}[N-2]\\ \vdots & \vdots & \ddots & \vdots\\ R_{XX}[N-1] & R_{XX}[N-2] & \cdots & R_{XX}[0]\end{pmatrix}}_{R_{\mathbf{X}_{0:N}}}\underbrace{\begin{pmatrix}a_1\\a_2\\\vdots\\a_N\end{pmatrix}}_{\mathbf{a}}=\underbrace{\begin{pmatrix}R_{XX}[1]\\R_{XX}[2]\\\vdots\\R_{XX}[N]\end{pmatrix}}_{\mathbf{r}}$$

which may be solved for the unknown coefficients, by setting

$$\mathbf{a}=R_{\mathbf{X}_{0:N}}^{-1}\mathbf{r} \tag{3}$$

1.1.1 Some Observations

It is worth reinforcing the fact that in the above development, n, is only a place holder. Since the random process is stationary, it does not matter whether we are minimizing the expected squared prediction error at $n=0$, $n=1$, or any other value of n.

The matrix, $R_{\mathbf{X}_{0:N}}$ in equation (3), is the auto-correlation matrix of the random vector, $\mathbf{X}_{0:N}$, whose elements are $X[0]$ through $X[N-1]$. We may call this the N^{th} order auto-correlation matrix for the stationary random process, $X[n]$. $R_{\mathbf{X}_{0:N}}$ has a great deal of structure:

- Obviously, it is symmetric, meaning that

$$R_{\mathbf{X}_{0:N}} = R^t_{\mathbf{X}_{0:N}}$$

- All elements on the main diagonal of $R_{\mathbf{X}_{0:N}}$ have a common value. Similarly, each sub-diagonal consists of identical elements. Such matrices are said to have "Toeplitz" structure.

- $R_{\mathbf{X}_{0:N}}$ has the property that for any vector, $\mathbf{y}$,

$$\begin{aligned} \mathbf{y}^t R_{\mathbf{X}_{0:N}} \mathbf{y} &= \mathbf{y}^t E\left[\mathbf{X}_{0:N}\mathbf{X}^t_{0:N}\right]\mathbf{y} \\ &= E\left[\left(\mathbf{y}^t\mathbf{X}_{0:N}\right)\left(\mathbf{y}^t\mathbf{X}_{0:N}\right)^t\right] \geq 0 \end{aligned}$$

since $\mathbf{y}^t\mathbf{X}_{0:N}$ is a scalar random variable and $E\left[\left(\mathbf{y}^t\mathbf{X}_{0:N}\right)\left(\mathbf{y}^t\mathbf{X}_{0:N}\right)^t\right]$ is its power, which cannot be negative. This property is known as "positive semi-definiteness."

A symmetric matrix has all real-valued eigenvalues and a set of mutually orthonormal eigenvectors. A symmetric, "positive definite" matrix, R, has $\mathbf{y}^t R\mathbf{y} > 0$ for all non-zero vectors, $\mathbf{y}$. Positive definite matrices are necessarily non-singular (i.e., invertible). We say that $R_{\mathbf{X}_{0:N}}$ is positive **semi**-definite, since we cannot be certain that there is not some non-zero $\mathbf{y}$ for which $\mathbf{y}^t R_{\mathbf{X}_{0:N}}\mathbf{y} = 0$. However, if this happens, then

$$\begin{aligned} 0 &= \mathbf{y}^t R_{\mathbf{X}_{0:N}}\mathbf{y} = \mathbf{E}\left[\left(\mathbf{y}^t\mathbf{X}_{0:N}\right)^2\right] \\ &= E\left[\left\{y_0 X[0] + \cdots + y_{N-1}X[N-1]\right\}^2\right] \\ &= y_{N-1}\cdot E\left[\left\{X[N-1] - \left(\frac{-y_0}{y_{N-1}}X[0] + \cdots + \frac{-y_{N-2}}{y_{N-1}}X[N-2]\right)\right\}^2\right] \\ &= y_{N-1}\cdot E\left[\left\{X[n] - \sum_{k=1}^{N-1} y'_k X[n-k]\right\}^2\right], \quad \text{for any } n \text{, where } y'_k = -\frac{y_{N-1-k}}{y_{N-1}} \end{aligned}$$

This means that $X[n]$ can be perfectly predicted (no innovations at all) using a linear predictor of order $N-1$, having coefficients y'_1 through y'_{N-1}. Of course, we do not expect perfect prediction to be possible. In any event, if it were possible using an order $N-1$ predictor, then we should not be attempting to find an order N predictor. In the above equation, we have assumed that the last element y_{N-1} of $\mathbf{y}$ is non-zero. More generally, we ca repeat the same argument for the last non-zero element of $\mathbf{y}$, say y_m, finding that $X[n]$ is a deterministic linear combination of the preceding m random variables, $X[n-m]$ through $X[n-1]$.

In view of the above arguments, there is no need to concern ourselves with matrices which are not **strictly positive definite**. Standard robust algorithms for inverting positive definite matrices always flag the anomalous "semi-definite" condition, in response to which we should simply reduce the order of the predictor and try again.

Positive definiteness might not mean a great deal to you, but it is an extremely useful property in linear algebra. Symmetric positive definite matrices have all eigenvalues strictly real and positive, with a mutually orthonormal set of eigenvectors. Simple, efficient and extremely robust algorithms exist (e.g., Cholesky Decomposition) for inverting symmetric positive definite (SPD) matrices. These algorithms can stably invert extremely large matrices, with minimal sensitivity to numerical errors.

1.1.2 The Orthogonality Principle

You would be right to wonder why we used the term "normal equations" to refer to the constaints expressed in equation (2), i.e.,

$$R_{XX}[p] = \sum_{k=1}^{N} a_k R_{XX}[p-k], \quad 1 \leq p \leq N$$

To understand this terminology, and the intuition which it brings, we start again from the second line in equation (2), repeated here as

$$E\left[X[n-p]\left(X[n] - \sum_{k=1}^{N} a_k X[n-k]\right)\right] = 0, \quad 1 \leq p \leq N$$

Now observe that the term in parentheses is the prediction error, $E[n]$, so that the optimal linear predictor is the one whose prediction error satisfies

$$E[E[n]X[n-p]] = 0, \quad 1 \leq p \leq N \tag{4}$$

Since the prediction error will always be a zero-mean random process, this is equivalent to saying that $E[n]$ must be uncorrelated with each of the available previous samples, $X[n-1]$ through $X[n-N]$. This makes perfect sense, since if $E[n]$ were correlated with $X[n]$, it would be possible to predict $E[n]$ using $X[n]$, thereby further reducing the prediction error energy. We summarize this intuition in the following very simple theorem.

Theorem 1 *If two random variables A and B are correlated, then it is possible to form a prediction, $\bar{A} = \alpha B$, such that $E\left[\left(A-\bar{A}\right)^2\right] < E\left[A^2\right]$.*

Proof. *Simply observe that $E\left[\left(A-\bar{A}\right)^2\right] = E\left[(A-\alpha B)^2\right]$ is minimized by differentiating by α and setting the result to 0, giving*

$$0 = E[B(A-\alpha B)] = E[AB] - \alpha E\left[B^2\right]$$

which yields non-zero α whenever $E[AB] \neq 0$. If the optimal α is non-zero, then $E\left[\left(A-\bar{A}\right)^2\right]$ must be smaller than $E\left[A^2\right]$. ■

It is possible to understand random variables as vectors in a linear vector space – one can check that they satisfy all the right mathematical properties. It is then not hard to show that correlation plays the role of an inner product in this vector space – one can check that it satisfies the bi-linearity and Cauchy-Schwarz properties required of an inner product. That is,

$$\langle A, B \rangle \triangleq E\left[AB\right]$$

It follows that power plays the role of squared length,

$$\|A\|^2 = \langle A, A \rangle = E\left[A^2\right]$$

With these connections in place, equation (4) tells us that the optimal LP coefficients should have the property that the prediction error, $E[n]$, is orthogonal (or "normal") to each of the predictor inputs, $X[n-1]$ through $X[n-N]$. This is known as the "orthogonality principle." This perspective allows us to leave the equations themselves behind and adopt a high level, geometric perspective of linear prediction. As we shall see, this perspective substantially eases the burden of understanding and remembering many results from statistical signal processing.

1.1.3 Whiteness of the Innovations Process

Suppose we were able to find an optimum linear predictor of infinite order. According to the orthogonality principle, the prediction residual, $E[n]$, is uncorrelated with (orthogonal to) any of the source random variables, $X[n-k]$, for $k < 0$. But

$$E[n-m] = X[n-m] - \sum_{k=1}^{\infty} a_k X[n-m-k]$$

is a linear combination of random variables which are all uncorrelated with $E[n]$, so $E[n]$ is uncorrelated with $E[n-m]$ for all m. That is, $E[n]$ has auto-correlation sequence

$$R_{EE}[m] = E\left[E[n]\,E[n-m]\right] = \begin{cases} \sigma_E^2 & m = 0 \\ 0 & \text{otherwise} \end{cases}$$

and power density spectrum

$$S_{EE}(\omega) = \sigma_E^2, \quad -\pi \le \omega \le \pi$$

From the above, we conclude that the innovations process is white, so long as the order of the linear predictor is unbounded. In practice, we have to work with finite order predictors. Nevertheless, most random processes exhibit rapidly decaying auto-correlation functions so that beyond a certain distance into the past, the outcomes of the random process have negligible predictive power. For this reason, as the order of the linear predictor is increased, the innovations process becomes increasingly white. For this reason, the causal filter with impulse response

$$h[n] = \begin{cases} 1 & n = 0 \\ -a_n & 1 \leq n \leq N \\ 0 & \text{otherwise} \end{cases}$$

is known as the "whitening filter." This is because applying this filter to the source process, $X[n]$, yields the innovations process,

$$E[n] = X[n] - \sum_{k=1}^{N} a_k X[n-k] = \sum_{k=0}^{N} h[k] X[n-k] = (h \star X)[n]$$

1.2 Deterministic Formulation

The result in equation (3) expresses the optimal LP coefficients in terms of the auto-correlation function, $R_{XX}[m]$, of the underlying random process. In practice, $R_{XX}[m]$ must be estimated for $0 \leq m \leq N$, based on available signal samples. Recall from the last chapter, that biased and unbiased estimators can be developed for the auto-correlation sequence. The biased estimator is generally to be preferred, due to the reduced variance of the estimates. In this case, we esimate

$$R_{XX}[m] \approx r_{xx}^{\text{biased}}[m] = \frac{1}{K} \sum_{n=m}^{K-1} x[n] x[n-m]$$

where $K > N$ is the number of samples which are available for forming the estimate. Usually K is much larger than N, the order of the linear predictor.

The strategy described above, in which the LP coefficients are estimated by first estimating $R_{XX}[m]$ from a block of K samples, $x[0]$ through $x[K-1]$, and then solving equation (3), is known as the "Yule-Walker Algorithm."

An alternative strategy for optimizing LP coefficients on the basis of a block of K observed samples, $x[n]$, is to find to LP coefficients which directly minimize the observed prediction error over these samples. Specifically, we wish to minimize

$$\frac{1}{K-N} \sum_{n=N}^{K-1} \left(x[n] - \sum_{k=1}^{N} a_k x[n-k] \right)^2 \tag{5}$$

We refer to this strategy as "direct optimization" or "deterministic optimization" of the LP parameters. The minimization objective is not directly connected with the statistics of the underlying random process, although if $X[n]$ really is stationary the deterministic objective will converge to the statistical objective in equation (1), as K becomes large.

Differentiating the deterministic objective in equation (5), with respect to each a_p, and setting the result equal to 0, yields the following equations

$$\underbrace{\left(\sum_{n=N}^{K-1} x[n-p]\, x[n]\right)}_{r_{p,0}} = \sum_{k=1}^{N} a_k \underbrace{\left(\sum_{n=N}^{K-1} x[n-p]\, x[n-k]\right)}_{r_{p,k}}, \quad 1 \leq p \leq N$$

which may be expressed in matrix form as

$$\underbrace{\begin{pmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,N} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ r_{N,1} & r_{N,2} & \cdots & r_{N,N} \end{pmatrix}}_{R} \underbrace{\begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{pmatrix}}_{\mathbf{a}} = \underbrace{\begin{pmatrix} r_{1,0} \\ r_{2,0} \\ \vdots \\ r_{N,0} \end{pmatrix}}_{\mathbf{r}} \tag{6}$$

Note that the matrix, R, has much less structure in this case. It is easy to see that $r_{p,k} = r_{k,p}$, so that $R = R^t$ is a symmetric matrix. However, it is not generally Toeplitz and not necessarily positive semi-definite. This complicates the process of inverting R.

2 Some Applications of Linear Prediction

2.1 Linear Predictive Coding (LPC)

The idea behind linear predictive coding is to quantize and encode the prediction residual sequence, $e[n]$, transmitting it to a decoder in place of the original source sequence, $x[n]$. Conceptually, if we send

$$e[n] = x[n] - \sum_{k=1}^{N} a_k x[n-k]$$

to the receiver, it may recover $x[n]$ from the recursive difference equations

$$x[n] = e[n] + \sum_{k=1}^{N} x[n-k]$$

where $x[n]$ is taken to be 0 for $n < 0$, with $x[0]$ the first information-bearing source sample.

In practice, both $x[n]$ and $e[n]$ are real-valued quantities which must be quantized into a finite number of discrete values for efficient transmission. For the present discussion, it is sufficient to consider a simple uniform quantizer, which approximates $e[n]$ by the nearest multiple of a quantization step size, Δ. Specifically, the quantized version of $e[n]$ is written

$$e'[n] = Q(e[n]) = \Delta \cdot \left\lfloor \frac{e[n]}{\Delta} + \frac{1}{2} \right\rfloor$$

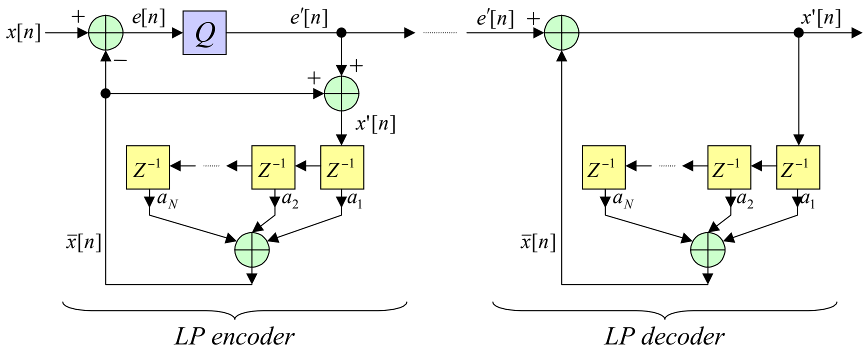

Figure 1: N^{th} *order linear predictive encoder and corresponding decoder, incorporating quantization of the prediction error sequence,* $e[n]$.

where $\left\lfloor u+\frac{1}{2}\right\rfloor$ rounds u to the nearest integer.

As a result of quantization, the decoder does not recover $x[n]$ exactly. Instead, it recovers an approximate representation of the source sequence, which we denote $x'[n]$. To avoid propagation of quantization errors, the LP encoder forms $e[n]$ by applying the LP coefficients to the approximate values, $x'[n-1]$ through $x'[n-N]$, which the decoder will be using. This process is summarized by the following equations:

$$e[n]=x[n]-\sum_{k=1}^{N}a_k x'[n-k] \tag{7}$$

and

$$x'[n]=e'[n]+\sum_{k=1}^{N}a_k x'[n-k] \tag{8}$$

Figure 1 gives a block diagram showing linear predictive encoding and decoding systems which implement these equations. Note that the encoder must be able to recover exactly the same values, $x'[n]$, which are available at the decoder, so that it can form $e[n]$ from the new value of $x[n]$ and the previous values of $x'[n]$.

To understand why linear prediction is beneficial for coding the sequence, $x[n]$, we begin by combining equations (7) and (8) to get

$$\begin{aligned} x[n]-x'[n] &= \left(e[n]+\sum_{k=1}^{N}a_k x'[n-k]\right)-\left(e'[n]+\sum_{k=1}^{N}a_k x'[n-k]\right) \\ &= e[n]-e'[n] \end{aligned}$$

This tells us that the error in $x[n]$ is identical to the quantization error assocatiated with applying the quantizer, Q, to $e[n]$. If Δ is sufficiently small,

$E[n]-E'[n]$ and hence $X[n]-X'[n]$ are distributed uniformly over the interval $\left(-\frac{1}{2},\frac{1}{2}\right)$, so that the quantization error power is

$$E\left[\left(X[n]-X'[n]\right)^2\right]=E\left[\left(E[n]-E'[n]\right)^2\right]=\frac{\Delta^2}{12}$$

In other words, it does not matter whether we apply the quantizer directly to $x[n]$ or to the prediction error sequence, $e[n]$; in either case, the sequence recovered by the decoder will differ from the original by an amount whose variance is $\Delta^2/12$.

On the other hand, the number of bits required to code the quantized values, $e'[n]$, varies with the log of the source variance. In particular, for Gaussian sources, the average (expected) number of bits required to code each output of a uniform scalar quantizer with small step size Δ, when applied directly to $x[n]$ is given by

$$\frac{1}{2}\log_2\left(2\pi e\sigma_X^2\right)-\log_2\Delta$$

while the number of bits required to code the output of the same quantizer, when applied to $e[n]$ is

$$\frac{1}{2}\log_2\left(2\pi e\sigma_E^2\right)-\log_2\Delta$$

Thus, for the same amount of error between $x[n]$ and $x'[n]$, the use of a linear predictor provides a saving of

$$\frac{1}{2}\log_2\frac{\sigma_X^2}{\sigma_E^2}\quad\text{bits per sample}$$

The ratio, σ_X^2/σ_E^2 is known as the "prediction coding gain." It is maximized by selecting the LP coefficients in such a way as to minimize σ_E^2. Under the assumption that Δ is small, we can ignore the difference between $x[n]$ and $x'[n]$ in equation (7), so that σ_E^2 is the variance of the innovations process,

$$X[n]-\sum_{k=1}^{N}a_kX[n-k]$$

In practical applications, prediction gains between 10 : 1 and 100 : 1 are not uncommon.

2.2 AR Power Spectrum Estimation

Recall that for sufficiently large prediction orders, N, the innovations process, $E[n]$, is white. That is, $S_{EE}(\omega)=\sigma_E^2$. But $E[n]$ is obtained by filtering $X[n]$ with the whitening filter, $h[n]$, given by

$$h[n]=\begin{cases}1 & n=0\\ -a_n & 1\leq n\leq N\\ 0 & \text{otherwise}\end{cases}$$

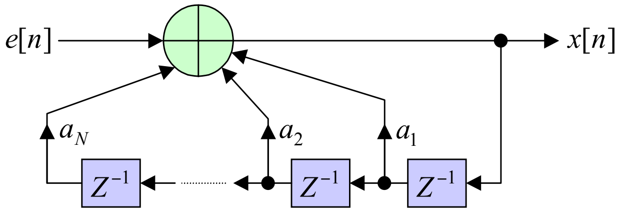

Figure 2: *Auto-regressive power spectrum model. The* $e[n]$ *are outcomes of an IID (white) noise process,* $E[n]$*, so the generated* $x[n]$ *are outcomes of a random process,* $X[n]$*, with power spectrum* $S_{XX}(\omega) = \sigma_E^2 / \left|1 - \sum_{k=1}^{N} a_k e^{-j\omega k}\right|^2$.

It follows that

$$\begin{aligned} \sigma_E^2 &= S_{EE}(\omega) = S_{XX}(\omega) \cdot \left|\hat{h}(\omega)\right|^2 \qquad (9) \\ \implies & S_{XX}(\omega) = \frac{\sigma_E^2}{\left|\hat{h}(\omega)\right|^2} = \frac{\sigma_E^2}{\left|1 - \sum_{k=1}^{N} a_k e^{-j\omega k}\right|^2} \end{aligned}$$

Equation (9) suggests a new method for power spectrum estimation. Starting with limited observations of a signal, $x[n]$, over the range $0 \le n < K$, we first estimate the auto-correlation sequence, $R_X[m]$, for $|m| \le N$ where N is usually much smaller than K. We then solve equation (3) to find the N^{th} order linear prediction coefficients, a_1 through a_N, and hence estimate the power spectrum, $S_{XX}(\omega)$.

The spectral estimation technique described above is known as auto-regressive (AR) spectral estimation. The reason for this is that finite order predictors are able to perfectly estimate the power spectrum of an auto-regressive random process. An N^{th} order auto-regressive random process is one whose outcomes can be described as

$$x[n] = \sum_{k=1}^{N} a_k x[n-k] + e[n]$$

where $e[n]$ is the realization of a white random noise source, $E[n]$, with variance σ_E^2. The AR model is depicted in Figure 2. Evidently, an N^{th} order predictor finds the coefficients, a_1 through a_N of the AR model, since these are the coefficients which satisfy the orthogonality principle

$$E\left[\left(X[n] - \sum_{k=1}^{N} a_k X[n-k]\right) X[n-p]\right] = 0, \quad 1 \le p \le N$$

In practice, the LP coefficients must be estimated from a finite number of outcomes, $x[n]$, from the random process, so we do not have access to the exact statistics, and hence cannot be expected to find the AR model parameters exactly. Nevertheless, if the order of the predictor is well matched to the order of the underlying AR random process, the estimated power spectrum is generally much more accurate than that obtained using any of the methods previously considered for power spectrum estimation.

Various strategies have been developed to estimate the model order, N, from the properties of the innovations variance, σ_E^2. The basic idea is that σ_E^2 should decrease rapidly as N is increased, but once the model order is reached, σ_E^2 should be approximately constant, influenced only by uncertainties in our estimates of the auto-correlation function. One may think of finding N by looking for the "knee" point in a plot of σ_E^2 against N.

Even if the underlying random process is not auto-regressive, AR spectrum estimation provides a robust technique for estimating power spectra from a limited set of observations, $x[n]$. In this context, the model order, N, provides a means of trading the frequency "resolution" of the estimate, against variance in the estimate due to limited observations. We have already seen the need for this type of trade-off in our study of the Periodogram and the Bartlett and Blackman-Tukey power spectrum estimation strategies.

2.3 Speech Synthesis

Let us assume that our linear predictor has sufficient order, N, that the innovations process, $E[n]$, is approximately white. This means that the power spectrum of the random process is well modeled by the AR spectral estimate in equation (9). Moreover, a random process with this same power spectrum may be synthesized in the manner suggested by Figure 2, driving a white noise generator into an all-pole filter, having transfer function

$$G(z) = \frac{1}{1 - \sum_{k=1}^{N} a_k z^{-k}}$$

$G(z)$ is known as the LP synthesis filter.

2.3.1 Stability and Spectral Factorization

The astute reader will be wondering right now whether we have any guarantee that $G(z)$ is stable. It can be shown that whenever the LP coefficients are derived by solving equation (3) with a valid correlation matrix, $R_{\mathbf{X}_{0:N}}$, the whitening filter,

$$H(z) = 1 - \sum_{k=1}^{N} a_k z^{-k}$$

will have minimum phase. As a result, $G(z)$ is guaranteed to be stable. A valid correlation matrix, $R_{\mathbf{X}_{0:N}}$, is one whose entries are drawn from a valid auto-correlation function, $R_{XX}[m]$, which is any function having the property that $R_{XX}[m] = R_{XX}[-m]$ and $R_{XX}[0] \geq R_{XX}[m]$, $\forall m$.

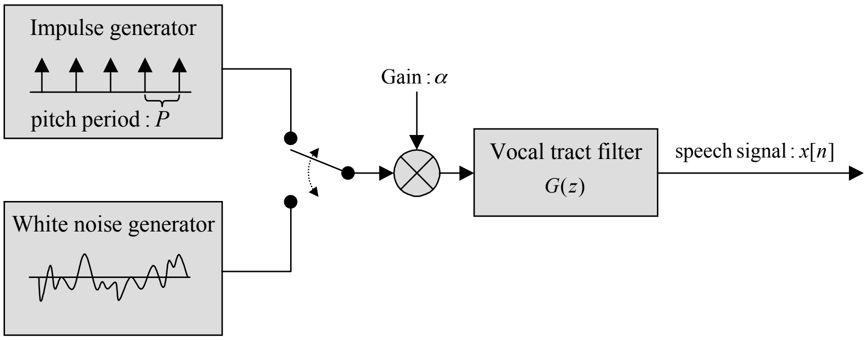

Figure 3: *Crude speech production model.*

In practice, we may derive the LP coefficients using some other means which does not guarantee minimum phase for $H(z)$. Equation (6), for example, does not necessarily lead to minimum phase whitening and hence stability of $G(z)$. In such cases, a stable synthesis filter, $G(z)$, must be derived by spectral factorization.

To understand the spectral factorization technique, observe that $S_{XX}(\omega)$, can be expressed in the Z-transform domain as

$$S_{XX}(z) = \frac{\sigma_E^2}{H(z)\, H(z^{-1})}$$

One can easily verify that, $S_{XX}(\omega) = S_{XX}(z)|_{z=e^{j\omega}}$. For this reason, the AR spectral estimation technique is said to yield an "all-pole" spectrum, and it is often referred to as "all-pole spectral estimation". Applying white noise with variance σ_E^2 to any synthesis filter $G(z)$, yields a random process with power spectrum $\sigma_E^2 G(z)\, G\left(z^{-1}\right)$. It follows that $G(z)$ should be constructed by taking all of the poles of $S_{XX}(z)$ which lie inside the unit circle. Equivalently, we first create a minimum phase version, $H_m(z)$, of $H(z)$ by reciprocating any zeros which lie outside the unit circle, and then set $G(z) = 1/H_m(z)$.

2.3.2 Speech Production Model

A crude but widely used model of speech production is that shown in Figure 3.Two types of activity are commonly identified:

Voiced speech: Voicing periods may be associated with the pronunciation of vowels. During these periods, the vocal chords periodically interrupt the flow of air from the lungs creating an excitation source which may be crudely modeled by a roughly periodic impulse train. The period, P, determines the pitch of the speech.

Unvoiced speech: Unvoiced periods may be associated with the pronunciation of consonants. During these periods, the vocal chords do not resonate. Instead, a portion of the vocal tract is restricted and air is forced through the restriction with sufficient pressure to induce turbulence. The turbulent air flow may be modeled crudely by a white noise source.

The remainder of the vocal tract (and other physical organs) act as a filter which shapes the spectral characteristics of the raw excitation sources described above. This filter is often called a "formant" filter, since it forms (or shapes) the harmonics produced by the resonating vocal chords (these are called "formant frequencies"). An all-pole model for the formant filter is somewhat justifiable.

2.3.3 Simple Vocoder

In view of the above discussion, it is possible to think of synthesizing speech using the all-pole synthesis filter, $G(z)$, obtained by LP analysis, together with a switched excitation source, exactly as shown in the above figure. A number of speech compressors work in this way. Rather than encoding the speech itself, we encode the LP parameters which essentially model the vocal tract. The LP parameters are updated periodically (typically every 20 milliseconds), along with the gain parameter, α, the voiced/unvoiced switch position and, if voiced, the pitch period, P.

Speech coders which work in this way are called "vocoders", to suggest the fact that they are encoding properties of the vocal tract, rather than the speech waveform itself. The synthesized waveform might bear no resemblance whatsoever to the original. However, so long as the power spectrum of the synthesized speech closely approximates that of the original, the human hearer will perceive the spoken message correctly. This is because the human ear essentially acts like a short term spectrum analyser.