SIGNAL MODELING (1)

INTRODUCTION
Signal modeling:
The central issues in signal modeling:
Modeling a signal as the output of a linear shift-invariant filter:
Techniques for determining model parameters:
Optimization Theory:

PART I. DETERMINISTIC SIGNAL MODELING
The signal model:

THE LEAST SQUARES (DIRECT) METHOD
The modeling error:
The error function to be minimized:
Determining the filter coefficients:

THE PADÉ APPROXIMATION
Modeling error:
Determining the filter coefficients:

THE PRONY’S METHOD
Alternative modeling error:
The square error to be minimized:
Determining the filter coefficients $a_p(k)$:
Minimum error:
Determining the filter coefficients $b_q(k)$:

THE SHANK’S METHOD

Some useful MATLAB functions
**SIGNAL MODELING**

**INTRODUCTION**

Signal modeling:

Signal modeling is concerned with the representation of signals in an efficient manner. It has been widely applied in speech and audio coding, imaging compression, data compression, and spectrum estimation.

**Example 1.** Consider a signal $x(n)$ consisting of 2000 data values, $x(0)$, $x(1)$, …, $x(1999)$. The plot of the signal $x(n)$ is shown in Fig. 1. Now we want to model this signal. What should we do?

![Fig. 1. (a) A given signal $x(n)$ to be modeled. (b) Difference between the given and modeled signals.](image)

**Step 1.** From our knowledge, the signal in Fig. 1 is sinusoidal. Therefore, we choose a parametric model of sinusoid to model the signal. A sinusoid model may be of the form $x(n) = A \sin(n \omega_0 + \phi)$ with parameters, $A$, $\omega_0$, and $\phi$.

**Step 2.** After having chosen a parametric model, we shall determine the parameters, $A$, $\omega_0$, and $\phi$, for the model. The parameters should be found based on a certain criterion (e.g., least squares method) so that the parameters provides the best approximation to the given signal. In the present case, we may use the direct observation method to estimate the parameters. From the figure we can estimate that $A=2$, $\phi = -\pi / 2 = -1.57$, and $\omega_0 = 2\pi / 200 = 0.0314$, which makes $x(n) = 2 \sin(0.0314n - 1.57)$. With the model parameters found, the model is established.

**Step 3.** After having established the model, we should evaluate the model and see how accurate the model will be, by looking into the error (difference) between the given signal and the model signal (see Fig. 1(b)).

Now consider to send the given signal $x(n)$ in Fig. 1(a) across a communication channel with two schemes: (1) directly send these 2000 signal values; or (2) only send four parameter values: the amplitude $A$, the phase $\phi$, the frequency $\omega_0$ and data length $N=2000$, and then reconstruct the signal using the model established. Obviously, the latter scheme of sending the signal is much more efficient than the formal one. With the established model, we can easily extrapolate the signal beyond $n=0$-1999, e.g., $x(2000)$, $x(2001)$, …, and interpolate if some values (e.g., $x(1400)$, …, $x(1500)$) are missing or severely distorted.

---

1 Fig. 1(a) originates from $x(n) = 2 \sin(2\pi n / 199.799 - \pi / 2)$
The central issues in signal modeling:

From the above discussion, we may summarize the following central issues in Signal Modeling:

1. Choosing an appropriate parametric model for modeling a given signal, deterministic or random;
2. Determining the model parameters that provide the best approximation to the given signal;
3. Evaluating the model performance by looking into an error between the model and the given signal.

Modeling a signal as the output of a linear shift-invariant filter:

![Diagram of LSI Filter]

**Fig. 2.** Modeling a signal \(x(n)\) as the response \(\hat{x}(n)\) of an LSI filter to an input \(v(n)\).

The model that is chosen and will be studied in this course is one that models a signal as the output of a causal linear shift-invariant filter that has a rational system function of the form

\[
H(z) = \frac{B_q(z)}{A_p(z)} = \frac{\sum_{k=0}^{q} b_q(k)z^{-k}}{1 + \sum_{k=1}^{p} a_p(k)z^{-k}} \tag{1}
\]

This is shown in Fig. 2. A causal LSI filter means that \(h(n)=0\) for \(n<0\). For such an LSI filter, the output \(\hat{x}(n)\) is related to the input \(v(n)\) by the linear constant coefficient difference equation,

\[
\hat{x}(n) + \sum_{k=1}^{p} a_p(k)\hat{x}(n-k) = \sum_{k=0}^{q} b_q(k)v(n-k) \tag{2}
\]

If the input is a unit sample function, i.e., \(v(n) = \delta(n)\), then Eq. (2) becomes

\[
h(n) + \sum_{k=1}^{p} a_p(k)h(n-k) = b_q(n) \tag{3}
\]

and the signal is modeled as the unit sample response of the LSI filter, i.e.,

\[
\hat{x}(n) = h(n) \tag{4}
\]

which is deterministic. It should be noted that, since \(\delta(n)\) contains all possible frequency components over the frequency range from \(-\infty\) to \(\infty\), then \(\hat{x}(n)\) may be used to model a large variety of deterministic signals.

If the input \(v(n)\) is a white noise process with a unit variance, then the output \(\hat{x}(n)\) of the LSI filter will be a random process, and thus, can be used to model a random process.

To establish the model (the output of a causal LSI filter), we will investigate the filter coefficients, \(a_p(k)\) and \(b_q(k)\), along with a description of the input signal \(v(n)\). Depending on the poles and zeros in the system function \(H(z)\), we may have pole-zero model, all pole model, and all zero model for deterministic signals, and stochastic models (ARMA, AR, and MA models) for random signals.

Techniques for determining model parameters:

After the model (modeling a signal as the output of a causal LSI filter) is chosen, the next step is to determine the model parameters that provide the best approximation to the given signal. There are many
ways to define “the best” approximation, and using different definitions, we will have different solutions to the modeling problem along with different techniques for finding the model parameters.

The techniques for finding $a_p(k)$ and $b_q(k)$ for deterministic signals are the least squares method, the Padé approximation, the Prony’s method and the Shank’s method; and the techniques for finding $a_p(k)$ and $b_q(k)$ for random signals are modified Yule-Walker equations and extend Yule-Walker equations. Each method is established based on a certain “best” approximation, or a certain error function.

We start with developing the signal modeling techniques for deterministic signals, and then extend the techniques to the case of random signals.

**Optimization Theory:**
To find model parameters for the best approximation is equivalent to find model parameters for the minimal error between the given signal $x(n)$ and the signal model $\hat{x}(n)$, i.e., $e(n)= x(n) - \hat{x}(n)$. To minimize the error, we shall employ optimization theory.

**Theorem.** If $f(z, z^*)$ is a real-valued function of $z$ and $z^*$ and if $f(z, z^*)$ is analytic with respect to both $z$ and $z^*$, then the stationary points of $f(z, z^*)$ may be found by setting $\partial f(z, z^*)/\partial z = 0$ or $\partial f(z, z^*)/\partial z^* = 0$ and solving for $z$.

At the stationary points $f(x)$ are local and global minima!

**Example. The stationary point and the minima**
Consider that $f(z, z^*) = |e(z)|^2$ is a real-valued function of a complex number $z$, where $e(z) = x(n)z + b(n)$ is a linear function of $z$ and $x(n)$ and $b(n)$ are complex. The function $f(z) = |e(z)|^2$ can be written as $f(z, z^*) = e(z)e^*(z)$. The stationary points of $f(z)$ can be in two ways.

(i) If $f(z, z^*) = e(z)e^*(z)$ is treated as a function of $z$ with $z^*$ being constant, then

$$\frac{d}{dz}|e(z)|^2 = e^*(z)\frac{d}{dz}e(z) = [x^*(n)z^* + b^*(n)]x(n)$$

(ii) If $f(z, z^*) = e(z)e^*(z)$ is treated as a function of $z^*$ with $z$ being constant, then

$$\frac{d}{dz^*}|e(z)|^2 = e(z)\frac{d}{dz^*}e^*(z) = [x(n)z + b(n)]x^*(n)$$

Setting both derivatives in Eqs. (5) and (6) to be zero and solving for $z$, we will get the same solution, $z = -b(n)/x(n)$, at which (the stationary point) $f(x)$ is minimum since $f(z) = |e(z)|^2 \geq 0$. In this course, the second method, i.e., setting $d\|e(z)\|^2/\partial z^* = [x(n)z + b(n)]x^*(n) = 0$, will be used since it is more convenient due to the autocorrelation of $x(n)$ defined as $r_x(k) = E{x(n)x^*(n-k)}$.

**PART I. DETERMINISTIC SIGNAL MODELING**

The signal model:
The problem to be considered is to model a deterministic signal, $x(n)$, as the unit sample response of a LSI filter with a system function in Eq. (2), namely, $\hat{x}(n) = h(n)$. Thus, the input $v(n)$ is the unit sample $\delta(n)$. It is assumed that $x(n)=0$ for $n < 0$ and that the filter $h(n)$ is causal. There are different methods available for finding the filter's coefficients $a_p(k)$ and $b_q(k)$ when different error functions are used to get “the best” approximation of $x(n)$ from $\hat{x}(n)$.
THE LEAST SQUARES (DIRECT) METHOD

The modeling error:
In this method, the modeling error
\[ e'(n) = x(n) - h(n) \]  
(7) is used to find the filter coefficients, \( a_p(k) \) and \( b_q(k) \), that give best approximation of \( x(n) \).

The error function to be minimized:
The error measure that is to be minimized is the squared error
\[ \varepsilon_{LS} = \sum_{n=0}^{\infty} |e'(n)|^2 \]  
(8)
Note that \( e'(n) = 0 \) for \( n < 0 \) since \( x(n) \) and \( h(n) \) are both assumed to be zero for \( n < 0 \) (i.e., causal).

Determining the filter coefficients:
To find the filter coefficients \( a_p(k) \) and \( b_q(k) \) that give the minimized squared error, we apply the optimization theory, setting
\[ \frac{\partial \varepsilon_{LS}}{\partial a_p^*(k)} = 0; \ k = 1, 2, \ldots, p \]  
(9)
\[ \frac{\partial \varepsilon_{LS}}{\partial b_q^*(k)} = 0; \ k = 0, 1, \ldots, q \]  
(10)
Using Parseval’s theorem, the least square error may be written in the frequency domain as follows
\[ \varepsilon_{LS} = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left| E'(e^{j\omega}) \right|^2 d\omega \]  
(11)
where \( E'(e^{j\omega}) = X(e^{j\omega}) - H(e^{j\omega}) \) is the Fourier transform of \( e'(n) = x(n) - h(n) \).

Substituting Eq. (11) into Eq. (9), we have
\[ \frac{\partial \varepsilon_{LS}}{\partial a_p^*(k)} = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left[ X(e^{j\omega}) - \frac{B_q(e^{j\omega})}{A_p(e^{j\omega})} \right] \frac{B_q^*(e^{j\omega})}{A_q^*(e^{j\omega})} e^{j\omega} d\omega = 0, \ k = 1, 2, \ldots, p \]  
(12)
Similarly, we have
\[ \frac{\partial \varepsilon_{LS}}{\partial b_q^*(k)} = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left[ X(e^{j\omega}) - \frac{B_q(e^{j\omega})}{A_p(e^{j\omega})} \right] \frac{1}{A_q^*(e^{j\omega})} e^{j\omega} d\omega = 0, \ k = 0, 1, \ldots, q \]  
(13)
In Eq. (12) there are \( p \) equations, and in Eq. (13) there are \( q + 1 \) equations. Thus, the optimum set of model parameters, \( a_p(k) \) and \( b_q(k) \), are defined implicitly in terms of a set of \( p + q + 1 \) equations that are nonlinear because \( B_q(z)/A_p(z) \) and \( A_p(z)B_q(z) \) create nonlinear equations of \( a_p(k) \) and \( b_q(k) \), where
\[ A_p(z) = 1 + \sum_{k=1}^{q} a_p(k)z^{-k} \quad \text{and} \quad B_q(z) = \sum_{k=0}^{q} b_q(k)z^{-k} \]. Although iterative techniques such as the method of steepest descent, Newton’s method, or iterative pre-filtering could be used to solve these equations, the least
squares approach is not mathematically tractable and not amenable to real-time signal processing applications. It is for this reason that we shall find some indirect, simple methods of signal modeling. In these methods, the modeling problem is changed slightly so that the model parameters may be found more easily.

**THE PADÉ APPROXIMATION**

The model error:
The Padé method sets \( h(n) = x(n) \) for \( n = 0, 1, \ldots, p+q \), and thus, the modeling error becomes

\[
e'(n) = x(n) - h(n) = \begin{cases} 0, & \text{for } n = 0, 1, 2, \ldots, p+q \\ \text{unknown,} & \text{otherwise} \end{cases}
\]

Determining the filter coefficients:
Unlike the least squares solution, the Padé approximation only requires solving a set of linear equations to find the coefficients \( a_p(k) \) and \( b_q(k) \). Specifically, the Padé method sets \( h(n) = x(n) \) for \( n = 0, 1, \ldots, p+q \) in Eq. (3). This leads to the following set of \( p + q + 1 \) linear equations in \( p + q + 1 \) unknowns,

\[
x(n) + \sum_{k=0}^{p} a_p(k) x(n-k) = \begin{cases} b_q(n), & n = 0, 1, \ldots, q \\ 0, & n = q+1, \ldots, q+p \end{cases}
\]

Note that \( x(n) = 0 \) for \( n < 0 \) because \( h(n) \) is causal and \( x(n) = h(n) \). In matrix form these equations become

\[
\begin{bmatrix}
  x(0) & 0 & \cdots & 0 \\
  x(1) & x(0) & \cdots & 0 \\
  x(2) & x(1) & \cdots & 0 \\
  \vdots & \vdots & \ddots & \vdots \\
  x(q) & x(q-1) & \cdots & x(q-p) \\
  x(q+1) & x(q) & \cdots & x(q-p+1) \\
  \vdots & \vdots & \ddots & \vdots \\
  x(q+p) & x(q+p-1) & \cdots & x(q) \\
\end{bmatrix}
\begin{bmatrix}
  a_p(1) \\
  a_p(2) \\
  \vdots \\
  a_p(p) \\
\end{bmatrix}
= \begin{bmatrix}
  b_q(0) \\
  b_q(1) \\
  b_q(2) \\
  \vdots \\
  b_q(q) \\
  0 \\
  \vdots \\
  0 \\
\end{bmatrix}
\]

In the Padé approximation method, two steps are used to solve for the coefficients \( a_p(k) \) and \( b_q(k) \). The first step is to solve for the \( p \) coefficients \( a_p(k) \) using the last \( p \) equations in the lower part of Eq. (15), i.e.,

\[
\begin{bmatrix}
  x(q+1) & x(q) & \cdots & x(q-p+1) \\
  x(q+2) & x(q+1) & \cdots & x(q-p) \\
  \vdots & \vdots & \ddots & \vdots \\
  x(q+p) & x(q+p-1) & \cdots & x(q) \\
\end{bmatrix}
\begin{bmatrix}
  a_p(1) \\
  a_p(2) \\
  \vdots \\
  a_p(p) \\
\end{bmatrix}
= \begin{bmatrix}
  0 \\
  0 \\
  \vdots \\
  0 \\
\end{bmatrix}
\]

Rearranging Eq. (16) we have the following set of \( p \) equations with the \( p \) coefficients \( a_p(k) \) \( (k = 1, 2, \ldots, p) \)

\[
\begin{bmatrix}
  x(q) & x(q-1) & \cdots & x(q-p+1) \\
  x(q+1) & x(q) & \cdots & x(q-p+2) \\
  \vdots & \vdots & \ddots & \vdots \\
  x(q+p-1) & x(q+p-2) & \cdots & x(q) \\
\end{bmatrix}
\begin{bmatrix}
  a_p(1) \\
  a_p(2) \\
  \vdots \\
  a_p(p) \\
\end{bmatrix}
= \begin{bmatrix}
  x(q+1) \\
  x(q+2) \\
  \vdots \\
  x(q+p) \\
\end{bmatrix}
\]

Eq. (17) may be expressed in a compact form as

\[
X_q \bar{\alpha}_p = -x_{q+1}
\]
where
\[ \mathbf{a}_p = [a_p(1), \ a_p(2), \ \ldots, \ a_p(p)]^T, \]
\[ \mathbf{x}_{q+1} = [x(q+1), \ x(q+2), \ \ldots, \ x(q+p)]^T, \]
\[ \mathbf{X}_q = \begin{bmatrix} x(q) & x(q-1) & \ldots & x(q-p+1) \\ x(q+1) & x(q) & \ldots & x(q-p+2) \\ \vdots & \vdots & \ddots & \vdots \\ x(q+p-1) & x(q+p-2) & \ldots & x(q) \end{bmatrix}. \]

is a \( p \times p \) non-symmetric Toeplitz matrix.

Depending on whether or not the matrix \( \mathbf{X}_q \) is invertible, there are three cases of interest:

(i) \( \mathbf{X}_q \) is nonsingular. In this case, the coefficients \( a_p(k) \) are uniquely determined by Eq. (17).

(ii) \( \mathbf{X}_q \) is singular and a solution \( \mathbf{a}_p \) to Eq. (18) exists. The solution is not unique, i.e., \( \mathbf{a}_p = \mathbf{a}_p + \mathbf{z} \) is also a solution.

(iii) \( \mathbf{X}_q \) is singular and no solution to Eq. (18) exists. In this case, we must set \( a_p(0) = 0 \), and thus Eq. (18) becomes
\[ \mathbf{X}_q \mathbf{a}_p = 0. \]

After obtaining the coefficients \( a_p(k) \), the second step is to solve for the numerator coefficients \( b_q(k) \) using the first \((q+1)\) equations in Eq. (15)
\[ \begin{bmatrix} x(0) & 0 & 0 & \ldots & 0 \\ x(1) & x(0) & 0 & \ldots & 0 \\ x(2) & x(1) & x(0) & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x(q) & x(q-1) & x(q-2) & \ldots & x(q-p) \end{bmatrix} \begin{bmatrix} 1 \\ a_p(1) \\ a_p(2) \\ \vdots \\ a_p(p) \end{bmatrix} = \begin{bmatrix} b_q(0) \\ b_q(1) \\ b_q(2) \\ \vdots \\ b_q(q) \end{bmatrix}, \]

or
\[ \mathbf{X}_q \mathbf{a}_p = \mathbf{b}_q \] (24)

where
\[ \mathbf{a}_p = [1, \ a_p(1), \ a_p(2), \ \ldots, \ a_p(p)]^T \]
\[ \mathbf{b}_q = [b_q(0), \ b_q(1), \ \ldots, \ b_q(q)]^T. \]

Equivalently, coefficients \( b_q(k) \) can be found using Eq. (14), i.e.,
\[ b_q(n) = x(n) + \sum_{k=1}^{p} a_p(k)x(n-k) \]
(27)

Note that the number of \( x(n) \) used is \( p+q+1 \) that is just the coefficients number, and that \( b_q(0) = x(0) \).

**Example 2. Padé Approximation** (Example 4.3.2 on p. 139 in textbook)

**Example 3. Filter Design Using Padé Approximation** (Example 4.3.4 on p. 142 in textbook)
Special notes for Padé Approximation:

(i) The Padé Approximation is not an optimum method.

(ii) The model that is formed from the Padé Approximation will always produce an exact fit to the data over the interval \([0, p+q]\), i.e., \(x(0), x(1), \ldots, x(p+q)\), provided that \(X_q\) is nonsingular. However, since the data outside the interval \([0, p+q]\) is never considered in the modeling process, there is no guarantee on how accurate the model will be for \(n > p+q\). In some cases, the model fits the data very well, while in others it may fit poor.

(iii) The Padé Approximation will give the correct model parameters for those cases in which \(x(n)\) has a rational \(z\)-transform and the model order is chosen to be large enough.

(iv) Since the Padé Approximation forces the model to match the signal only over a limited range of values, the model that is generated is not guaranteed to be stable.

(v) A given signal can be modeled using different LSI filters, e.g., pole-zero, all pole, and all zero models.

- **Denominator coefficients** \(a_p(n)\) are found by

\[
\begin{bmatrix}
    x(q) & x(q-1) & \cdots & x(q-p+1) \\
    x(q+1) & x(q) & \cdots & x(q-p+2) \\
    \vdots & \vdots & \ddots & \vdots \\
    x(q+p-1) & x(q+p-2) & \cdots & x(q)
\end{bmatrix} \begin{bmatrix}
    a_p(1) \\
    a_p(2) \\
    \vdots \\
    a_p(p)
\end{bmatrix} = \begin{bmatrix}
    x(q+1) \\
    x(q+2) \\
    \vdots \\
    x(q+p)
\end{bmatrix}
\]

- **Numerator coefficients** \(b_q(n)\) are determined by

\[
b_q(n) = x(n) + \sum_{k=1}^{p} a_p(k)x(n-k)
\]

Note that \(b_q(0) = x(0)\).

**THE PRONY’S METHOD**

**Alternative modeling error:**

In principle, the filter coefficients \(a_p(k)\) and \(b_q(k)\) should be determined by minimizing the modeling error

\[
e'(n) = x(n) - h(n)
\]

which in the frequency domain is of the form

\[
E'(z) = X(z) - \frac{B_q(z)}{A_p(z)}
\]

However, minimization of error \(e'(n)\) by setting

\[
\frac{\partial}{\partial a_p^*(k)} \left( \sum_{n=0}^{\infty} |e'(n)|^2 \right) = 0 \quad \text{and} \quad \frac{\partial}{\partial b_q^*(k)} \left( \sum_{n=0}^{\infty} |e'(n)|^2 \right) = 0
\]

usually yields nonlinear equations that are mathematically intractable like in the least squares method. Thus, it loses its practical applicability.

If multiplying \(A_p(z)\) on both sides of Eq. (29), then we have

\[
E(z) = A_p(z)E'(z) = A_p(z)X(z) - B_q(z)
\]

which in the time domain may be written as
\[ e(n) = \begin{cases} x(n) + \sum_{k=1}^{p} a_p(k)x(n-k) - b_q(n); & 0 \leq n \leq q \\ x(n) + \sum_{k=1}^{p} a_p(k)x(n-k); & n > q \end{cases} \]  

(31)

which is an alternative modeling error and is linear in the filter coefficients, \( a_p(k) \) and \( b_q(k) \). Thus, minimization of the error \( e(n) \) will give linear solution to the coefficients desired.

**The square error to be minimized:**

In the Prony’s method the error to be minimized for determining the coefficients \( a_p(k) \) is the square error

\[
\varepsilon_{p,q} = \sum_{n=q+1}^{\infty} [e(n)]^2 = \sum_{n=q+1}^{\infty} [x(n) + \sum_{k=1}^{p} a_p(k)x(n-k)]^2 ,
\]

(32)

which only depends on \( a_p(k) \) but not on \( b_q(k) \).

**Determining the filter coefficients \( a_p(k) \):**

To minimize the square error, we may set

\[
\frac{\partial \varepsilon_{p,q}}{\partial a^*_p(k)} = 0 , \quad \text{that is}
\]

\[
\frac{\partial \varepsilon_{p,q}}{\partial a^*_p(k)} = \frac{\partial}{\partial a^*_p(k)} \left[ \sum_{n=q+1}^{\infty} e(n)^2 \right] = \frac{\partial}{\partial a^*_p(k)} \left[ \sum_{n=q+1}^{\infty} e(n)e^*(n) \right] = \sum_{n=q+1}^{\infty} e(n) \frac{\partial e^*(n)}{\partial a^*_p(k)}
\]

\[
= \sum_{n=q+1}^{\infty} e(n) \frac{\partial}{\partial a^*_p(k)} \left[ x^*(n) + \sum_{k=1}^{p} a^*_p(k)x^*(n-k) \right] = \sum_{n=q+1}^{\infty} e(n)x^*(n-k) = 0 \quad k=1, 2, \ldots, p
\]

(33)

In the above equation, we see

\[
\sum_{n=q+1}^{\infty} e(n)x^*(n-k) = 0 \quad k=1, 2, \ldots, p
\]

(34)

which is known as the orthogonality principle, which follows from \( r_{ei}(k) = \sum_{n=q+1}^{\infty} e(n)x^*(n-k) = 0 \). This principle states that for the optimum linear predictor the estimation error will be orthogonal to the data \( x \). It is fundamental in mean-square estimation problems. Inserting \( e(n) \) in Eq. (31) into Eq. (34), we have

\[
\sum_{n=q+1}^{\infty} \left\{ x(n) + \sum_{l=1}^{p} a_p(l)x(n-l) \right\}x^*(n-k) = 0
\]

(35)

or equivalently,

\[
\sum_{l=1}^{p} a_p(l) \left\{ \sum_{n=q+1}^{\infty} x(n-l)x^*(n-k) \right\} = -\sum_{n=q+1}^{\infty} x(n)x^*(n-k)
\]

(36)

by defining

\[
r_{x}(k,l) = \sum_{n=q+1}^{\infty} x(n-l)x^*(n-k)
\]

(37)

which is a conjugate symmetry function, \( r_{x}(k,l) = r_{x}^*(l,k) \). Eq. (36) becomes

\[
\sum_{l=1}^{p} a_p(l)r_{x}(k,l) = -r_{x}(k,0) \quad k=1, 2, \ldots, p
\]

(38)
Eq. (38) is referred to as the *Prony normal equations* that are a set of $p$ linear equations with the $p$ unknowns, the coefficients $a_p(k)$. In the matrix form, the normal equations are

$$
\begin{bmatrix}
  r_s(1,1) & r_s(1,2) & \ldots & r_s(1,p) \\
  r_s(2,1) & r_s(2,2) & \ldots & r_s(2,p) \\
  \vdots & \vdots & \ddots & \vdots \\
  r_s(p,1) & r_s(p,2) & \ldots & r_s(p,p)
\end{bmatrix}
\begin{bmatrix}
  a_p(1) \\
  a_p(2) \\
  \vdots \\
  a_p(p)
\end{bmatrix}
= -
\begin{bmatrix}
  r_x(1,0) \\
  r_x(2,0) \\
  \vdots \\
  r_x(p,0)
\end{bmatrix}
$$

(39)

or

$$\mathbf{R}_s \mathbf{\bar{a}}_p = -\mathbf{r}_x$$

(40)

where $\mathbf{R}_s$ is a $p \times p$ autocorrelation matrix with elements defined in Eq. (37),

$$\mathbf{R}_s =
\begin{bmatrix}
  r_x(1,1) & r_x(1,2) & \ldots & r_x(1,p) \\
  r_x(2,1) & r_x(2,2) & \ldots & r_x(2,p) \\
  \vdots & \vdots & \ddots & \vdots \\
  r_x(p,1) & r_x(p,2) & \ldots & r_x(p,p)
\end{bmatrix}
$$

(41)

$$\mathbf{\bar{a}}_p = [a_p(1), \ a_p(2), \ \cdots, \ a_p(p)]^T,$$  \hspace{1cm}  (42)

$$\mathbf{r}_x = [r_x(1,0), \ r_x(2,0), \ \cdots, \ r_x(p,0)]^T,$$  \hspace{1cm}  (43)

Note that $r_x(k,l)$ are the autocorrelation of the deterministic signal $x(n)$ and obtained using an *infinite* set of data $x(n)$ for $n = q+1, \ldots, \infty$ (see Eq. (37)).

The coefficients $a_p(k)$ have been found by minimizing the error $\varepsilon_{p,q}$ in Eq. (32). Now we shall see the minimum error.

**Minimum error:**

From Eq. (32), it follows that

$$\varepsilon_{p,q} = \sum_{n=q+1}^{\infty} e(n)^2 = \sum_{n=q+1}^{\infty} e(n)e^*(n) = \sum_{n=q+1}^{\infty} e(n) \left[ x(n) + \sum_{k=1}^{p} a_p(k)x(n-k) \right]^*$$

$$= \sum_{n=q+1}^{\infty} e(n)x^*(n) + \sum_{n=q+1}^{\infty} e(n) \left( \sum_{k=1}^{p} a_p(k)x(n-k) \right)^*$$

$$= \sum_{n=q+1}^{\infty} e(n)x^*(n) + \sum_{k=1}^{p} a_p^*(k) \sum_{n=q+1}^{\infty} e(n)x^*(n-k)$$

Because of the orthogonal principle, $\sum_{n=q+1}^{\infty} e(n)x^*(n-k) = 0$, for $k = 1, 2, \ldots, p$ (but not for $k=0$), we have

$$\varepsilon_{p,q} = \sum_{n=q+1}^{\infty} e(n)x^*(n) = \sum_{n=q+1}^{\infty} \left[ x(n) + \sum_{k=1}^{p} a_p(k)x(n-k) \right] x^*(n)$$

$$= \sum_{n=q+1}^{\infty} \left[ x(n)x^*(n) + \sum_{k=1}^{p} a_p(k)x(n-k)x^*(n) \right]$$

(44)

Using the autocorrelation defined in Eq. (37), the *minimum error* may be

$$\varepsilon_{p,q} = r_x(0,0) + \sum_{k=1}^{p} a_p(k)r_x(0,k)$$

(45)
Determining the filter coefficients, $b_q(k)$:

Once the coefficients $a_p(k)$ have been found, the second step is to determine the $q+1$ numerator coefficients. Using the same way as in the Padé method, the numerator coefficients $b_q(n)$ are found by

$$b_q(n) = x(n) + \sum_{k=1}^{p} a_p(k)x(n-k), \quad (46)$$

or by the matrix form

$$
\begin{bmatrix}
  x(0) & 0 & 0 & \cdots & 0 \\
  x(1) & x(0) & 0 & \cdots & 0 \\
  x(2) & x(1) & x(0) & \cdots & 0 \\
  \vdots & \vdots & \vdots & \ddots & \vdots \\
  x(q) & x(q-1) & x(q-2) & \cdots & x(q-p)
\end{bmatrix}
\begin{bmatrix}
  1 \\
  a_p(1) \\
  a_p(2) \\
  \vdots \\
  a_p(p)
\end{bmatrix}
= \begin{bmatrix}
  b_q(0) \\
  b_q(1) \\
  b_q(2) \\
  \vdots \\
  b_q(q)
\end{bmatrix}, \quad (47)
$$

Example 4. Prony’s Method (Example 4.4.1 on p. 149 in textbook)

Example 5. Filter Design Using Prony’s Method (Example 4.4.2 on p. 152 in textbook)

Special notes for Prony’s Method:

(i) The Prony’s method is an optimum method because the optimization theory is used to minimize the squared modeling error. The orthogonal principle (Eq. (34)) is the result of the optimization, and it makes the modeling error $\varepsilon_{p,q}$ be minimum.

(ii) The Prony’s method uses an infinite set of signal values, i.e., $x(n)$ for all $n \geq 0$, to find the autocorrelation $r_s(k,l)$ defined in Eq. (37).

(iii) The Prony’s method is usually more accurate than the Padé Approximation.

- **Denominator coefficients** $a_p(n)$ are found by

$$
\begin{bmatrix}
  r_s(1,1) & r_s(1,2) & \cdots & r_s(1,p) \\
  r_s(2,1) & r_s(2,2) & \cdots & r_s(2,p) \\
  \vdots & \vdots & \ddots & \vdots \\
  r_s(p,1) & r_s(p,2) & \cdots & r_s(p,p)
\end{bmatrix}
\begin{bmatrix}
  a_p(1) \\
  a_p(2) \\
  \vdots \\
  a_p(p)
\end{bmatrix}
= \begin{bmatrix}
  r_s(1,0) \\
  r_s(2,0) \\
  \vdots \\
  r_s(p,0)
\end{bmatrix}, \quad k=1, 2, \ldots, p
$$

$$r_s(k,l) = \sum_{n=q+1}^{\infty} x(n-l)x^*(n-k);$$

- **Numerator coefficients** $b_q(n)$ are determined by

$$b_q(n) = x(n) + \sum_{k=1}^{p} a_p(k)x(n-k)$$

Note that $b_q(0) = x(0)$.

The minimum error is $\varepsilon_{p,q} = r_s(0,0) + \sum_{k=1}^{p} a_p(k)r_s(0,k)$.
The alternative form of the Prony’s method

Representing

$$X_q = \begin{bmatrix}
  x(q) & x(q-1) & \ldots & x(q-p+1) \\
  x(q+1) & x(q) & \ldots & x(q-p+2) \\
  x(q+2) & x(q+1) & \ldots & x(q-p+3) \\
  \vdots & \vdots & \ldots & \vdots 
\end{bmatrix}$$

(48)

and

$$x_{q+1} = \begin{bmatrix} x(q+1), x(q+2), x(q+3), \ldots \end{bmatrix}^T$$

(49)

we have

$$R_x = X_q^H X_q$$

(50)

and

$$r_x = X_q^H x_{q+1}$$

(51)

The Prony normal equations in Eq. (40) become

$$X_q^H X_q \vec{a}_p = -X_q^H x_{q+1}$$

(52)

which is the alternative form of the Prony normal equations.

THE SHANK’S METHOD (refer to pp. 154-160 in the textbook)

In the Prony’s method, \( a_p(k) \) are determined in an optimum way that the squares error \( \varepsilon_{p,q} = \sum_{n=q+1}^{\infty} |e(n)|^2 = \sum_{n=q+1}^{\infty} |x(n) + \sum_{k=1}^{p} a_p(k)x(n-k)|^2 \) is minimized, but \( b_q(k) \) are not. In the Shank’s method, \( b_q(k) \) are found in an optimum way that the squares error \( \varepsilon_S = \sum_{n=0}^{\infty} |e'(n)|^2 = \sum_{n=0}^{\infty} |x(n) - \hat{x}(n)|^2 \) is minimized assuming that \( a_p(k) \) is fixed.

After \( a_p(k) \) are determined with the Prony’s method, using the Shank’s method to find the optimum coefficients \( b_q(k) \) may lead to a further improvement of the model’s accuracy. Note that in this case both \( a_p(k) \) and \( b_q(k) \) are determined in optimum ways, but these optimum ways are different from the one in the least squares method.

Some useful MATLAB functions

```matlab
>> [B, A] = PRONY(H, NB, NA) % finds a filter with numerator order NB, denominator order NA, and having the impulse response in vector H.
```