Signal Modeling

Lecture Notes

Ahmet Ademoglu, *PhD*,
Bogazici University,
Institute of Biomedical Engineering

Some concepts and illustrations in this lecture are adapted from the textbook,
Statistical Digital Signal Processing and Modeling, Monson Hayes, *Wiley*.

- Signal as a sum of damped sinuisoids

$$x[n] \approx \sum_{k=1}^{p} \alpha_k \lambda_k^n \cos(n\omega_k)$$

- A rational system of polynomials

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

$$v[n] \longrightarrow \boxed{H(z) = \frac{B_q(z)}{A_p(z)}} \longrightarrow \hat{x}[n]$$

Least squares method

Determine $a_p[k]$ and $b_q[k]$ to minimize the squared sum of model error $e'[n] = x[n] - h[n]$, *i.e.*

$$\mathcal{E}_{LS} = \sum_{n=0}^{\infty} |e'[n]|^2$$

$h[n]$ and $x[n]$ are assumed to be 0 for $n < 0$.

Pade approximation

The signal is modeled as an impulse response of an LTI system:

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

In time domain

$$h[n] + \sum_{k=1}^{p} a_p[k]h[n-k] = b_q[n]$$

where $h[n] = 0$ for $n < 0$, $b_q[n] = 0$ for $n < 0$ and $n > q$.
Setting $h[n] = x[n]$ for $n = 0, 1, \ldots, p + q$ leads to

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

$$\begin{bmatrix} x[0] & 0 & \ldots & 0 \\ x[1] & x[0] & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ x[q] & x[q-1] & \ldots & x[q-p] \\ x[q+1] & x[q] & \ldots & x[q-p+1] \\ \vdots & \vdots & \vdots & \vdots \\ x[q+p] & x[q+p-1] & \ldots & x[q] \end{bmatrix} \begin{bmatrix} 1 \\ a_p[1] \\ a_p[2] \\ \vdots \\ a_p[p] \end{bmatrix} = \begin{bmatrix} b_q[0] \\ b_q[1] \\ \vdots \\ b_q[q] \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

The first step of the Pade approximation is to solve

$$\begin{bmatrix} x[q] & x[q-1] & \dots & x[q-p] \\ x[q+1] & x[q] & \dots & x[q-p+1] \\ \vdots & \vdots & \vdots & \vdots \\ x[q+p] & x[q+p-1] & \dots & x[q] \end{bmatrix} \begin{bmatrix} 1 \\ a_p[1] \\ a_p[2] \\ \vdots \\ a_p[p] \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

The second step is

$$\begin{bmatrix} x[q] & x[q-1] & \dots & x[q-p+1] \\ x[q+1] & x[q] & \dots & x[q-p+2] \\ \vdots & \vdots & \vdots & \vdots \\ x[q+p-1] & x[q+p-2] & \dots & x[q] \end{bmatrix} \begin{bmatrix} a_p[1] \\ a_p[2] \\ \vdots \\ a_p[p] \end{bmatrix} = - \begin{bmatrix} x[q+1] \\ x[q+2] \\ \vdots \\ x[q+p] \end{bmatrix}$$

In matrix notation $\mathbf{X}_q \bar{\mathbf{a}}_p = -\mathbf{x}_{q+1}$

1. If $\mathbf{X}_q$ is non-singular $\rightarrow$ unique solution $\bar{\mathbf{a}}_p = -\mathbf{X}_q^{-1} \mathbf{x}_{q+1}$
2. If $\mathbf{X}_q$ is singular $\rightarrow$ solution exists but it is not unique, $\bar{\mathbf{a}}_p$ with fewest nonzero elements is chosen
3. If $\mathbf{X}_q$ is singular $\rightarrow$ no solution exists, $a_p(0)$ is set to $a_p(0) = 0$ and $\mathbf{X}_q \bar{\mathbf{a}}_p = \mathbf{0}$ yields a solution

The numerator coefficients $b_q[k]$ are determined by

$$\begin{bmatrix} x[0] & 0 & \dots & 0 \\ x[1] & x[0] & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ x[q] & x[q-1] & \dots & x[q-p] \end{bmatrix} \begin{bmatrix} 1 \\ a_p[1] \\ a_p[2] \\ \vdots \\ a_p[p] \end{bmatrix} = \begin{bmatrix} b_q[0] \\ b_q[1] \\ \vdots \\ b_q[q] \end{bmatrix}$$

In matrix notation, $\mathbf{X}_0 \mathbf{a}_p = \mathbf{b}_q$

Example: Let $x[n] = [1, 4, 2, 1, 3]^T$ be approximated by $p = q = 2$.

Pade equations

$$\begin{bmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 2 & 4 & 1 \\ 1 & 2 & 4 \\ 3 & 1 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ a(1) \\ a(2) \end{bmatrix} = \begin{bmatrix} b(0) \\ b(1) \\ b(2) \\ 0 \\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} 2 & 4 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} a(1) \\ a(2) \end{bmatrix} = \begin{bmatrix} -1 \\ 3 \end{bmatrix}$$ No solution exist.

Setting $a[0] = 0$,

$$\begin{bmatrix} 2 & 4 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} a(1) \\ a(2) \end{bmatrix} = \mathbf{0} , \begin{bmatrix} a(1) \\ a(2) \end{bmatrix} = \begin{bmatrix} 2 \\ -1 \end{bmatrix}, \begin{bmatrix} b(0) \\ b(1) \\ b(2) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 2 & 4 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \\ 7 \end{bmatrix}$$

$$H(z) = \frac{2z^{-1} + 7z^{-2}}{2z^{-1} - z^{-2}} = 1 + \frac{8z^{-1}}{2z^{-1} - z^{-2}} = \rightarrow h[n] = \delta[n] + 8\frac{1}{2}^n u[n-1]$$

$$\hat{\mathbf{x}} = [1, 4, 2, 1, 0.5]^T$$

If $\mathbf{X}_q$ is singular and no solution exists, then it is not always possible to match all the signal values.

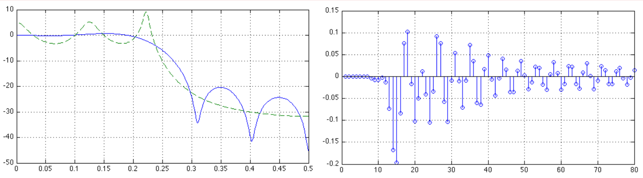
Lowpass filter design using Pade approximation

LP Filter response

$$I(e^{j\omega}) = \begin{cases} e^{-jn_d\omega} & |\omega| < \pi/2 \\ 0 & otherwise \end{cases} \longleftrightarrow i[n] = \frac{\sin[(n-n_d)\pi/2]}{(n-n_d)\pi}$$

1. Design 1: $p = 0,\ q = 10,\ p+q+1 = 11$ values of $i[n]$ are matched exactly. If $n_d = 5$ then $h[n]$ will match the maximum amount of energy in $i[n]$.
   $i[n] = [0.00637, 0., -0.1061, 0., 0.3183, 0.5, 0.3183, 0., -0.1061, 0., 0.0637]^T$.
   $h[n] = \begin{cases} i[n] & 0 \le n \le 11 \\ 0 & otherwise \end{cases}$ Rectangular window
2. Design2: $p = 5,\ q = 5,\ p+q+1 = 11$
   $\mathbf{a} = [1.0, -2.5256, 3.6744, -3.4853, 2.1307, -0.7034]^T$
   $\mathbf{b} = [0.0637, -0.1608, 0.1280, 0.0461, 0.0638, 0.0211]^T$

dashed : Pade Approximation   Error function



```
% MATLAB Script for plot
 w = [0:0.01:pi]';
 h = [0.00637, 0.,-0.1061,0., 0.3183,0.5, 0.3183, 0., -0.1061,0.,0.0637 ];
 a = [1.0  -2.5256  3.6744  -3.4853  2.1307  -0.7034];
 b = [0.0637, -0.1608, 0.1280, 0.0461, 0.0638, 0.0211];
 H1 = freqz(h,1,w);
 H2= freqz(b,a,w);
   subplot(1,2,1)
 plot(w/pi/2,20*log10(abs(H1)),'-',w/pi/2,20*log10(abs(H2)),'--');grid
 t = [0:79]';
 x=zeros(80,1);
 x(1) =1;
 x1 = filter(b,a,x);
 e = 0.5*sinc((t-5)/2);
 subplot(1,2,2)
 stem(e-x1);grid
```

Prony's Method

- A complex signal $x[n]$ is modeled as an impulse response of a LTI system $H(z) = \frac{B_q(z)}{A_p(z)}$.
- It is assumed that $x[n] = 0$ for $n < 0$ and $x[n]$ is known for all $n \geq 0$.
- The filter coefficients $a_p[k]$ and $b_q[n]$ are determined to make the impulse response $h[n]$ as close as possible to $x[n]$

$e'[n] = h[n] - x[n] \longleftrightarrow E'(z) = X(z) - H(z)$
$E'(z) = X(z) - \frac{B_q(z)}{A_p(z)} \longleftrightarrow E'(z)A_p(z) = X(z)A_p(z) - B_q(z) = E(z)$
In time domain,
$e[n] = a_p[n] * x[n] - b_q[n] = \hat{b}_q[n] - b_q[n]$
Since $b_q[n] = 0$ for $n > q$,

$$e[n] = \begin{cases} x[n] + \sum_{l=1}^{p} a_p[l]x[n-l] - b_q[n] & n = 0, 1, \ldots, q \\ x[n] + \sum_{l=1}^{p} a_p[l]x[n-l] & n > q \end{cases}$$

Instead of setting $e[n] = 0$ for $n = 0, 1, \ldots, p+q+1$ as in Pade approximation, Prony's method minimizes the squared error in the region where $b_q = 0$:

$$\mathcal{E}_{p,q} = \sum_{n=q+1}^{\infty} |e[n]|^2 = \sum_{n=q+1}^{\infty} |x[n] + \sum_{l=1}^{p} a_p[l]x[n-l]|^2$$

Complex Derivatives

$z = a + bj,\ a = \frac{1}{2}(z + z^*),\ b = \frac{1}{2j}(z - z^*)$

1. $\frac{dz}{dz} = 1 \longrightarrow \frac{d}{dz}z(a,b) = \frac{\partial z}{\partial a}\frac{da}{dz} + \frac{\partial z}{\partial b}\frac{db}{dz} = \frac{1}{2} + j\frac{1}{2j} = 1$
2. $\frac{dz}{dz^*} = 0$
3. $\frac{d|z|^2}{dz} = z^*$
4. $\frac{d|z|^2}{dz^*} = z$
5. $\frac{\partial \mathbf{a}^H\mathbf{z}}{\partial \mathbf{z}} = \mathbf{a}^*$
6. $\frac{\partial \mathbf{z}^H\mathbf{a}}{\partial \mathbf{z}^*} = \mathbf{0}$
7. $\frac{\partial \mathbf{a}^H\mathbf{z}}{\partial \mathbf{z}^*} = \mathbf{0}$
8. $\frac{\partial \mathbf{z}^H\mathbf{a}}{\partial \mathbf{z}^*} = \mathbf{a}$
9. $\frac{\partial \mathbf{z}^H\mathbf{a}\mathbf{z}}{\partial \mathbf{z}} = (\mathbf{A}\mathbf{z})^*$
10. $\frac{\partial \mathbf{z}^H\mathbf{a}\mathbf{z}}{\partial \mathbf{z}^*} = \mathbf{A}\mathbf{z}$

$$\frac{\partial \mathcal{E}_{p,q}}{\partial a_p^*[k]} = \sum_{n=q+1}^{\infty} \frac{\partial(e[n]e^*[n])}{\partial a_p^*[k]} = \sum_{n=q+1}^{\infty} e[n]\frac{\partial(e^*[n])}{\partial a_p^*[k]} = \sum_{n=q+1}^{\infty} e[n]x^*[n-k] = 0,\ \text{k=0,1,...,p}$$

$\sum_{n=q+1}^{\infty} e[n]x^*[n-k] = 0$ is called *orthogonality principle*

Substituting $e[n]$ into it

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

$$\sum_{l=1}^{p} a_p[l]\underbrace{\Big[\sum_{n=q+1}^{\infty} x[n-l]x^*[n-k]\Big]}_{r_x[k,l]} = -\sum_{n=q+1}^{\infty} x[n]x^*[n-k]\ ,\ \sum_{l=1}^{p} a_p[l]r_x[k,l] = -r_x[k,0]$$

$$\begin{bmatrix} r_x[1,1] & r_x[1,2] & \dots & r_x[1,p] \\ r_x[2,1] & r_x[2,2] & \dots & r_x[2,p] \\ \vdots & \vdots & \vdots & \vdots \\ r_x[p,1] & r_x[p,2] & \dots & r_x[p,p] \end{bmatrix} \begin{bmatrix} a_p[1] \\ a_p[2] \\ \vdots \\ a_p[p] \end{bmatrix} = -\begin{bmatrix} r_x[1,0] \\ r_x[2,0] \\ \vdots \\ r_x[p,0] \end{bmatrix},\ \mathbf{R}_x\bar{\mathbf{a}}_p = -\mathbf{r}_x$$

$$\underset{\text{min}}{\mathcal{E}_{p,q}} = \epsilon_{p,q} = \sum_{n=q+1}^{\infty} e[n]x^*[n] = r_x(0,0) + \sum_{k=1}^{p} a_p[k]r_x[0,k]$$

Augmented normal equations

$$\begin{bmatrix} r_x[0,0] & r_x[0,1] & r_x[0,2] & \dots & r_x[0,p] \\ r_x[1,0] & r_x[1,1] & r_x[1,2] & \dots & r_x[1,p] \\ r_x[2,0] & r_x[2,1] & r_x[2,2] & \dots & r_x[2,p] \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ r_x[p,0] & r_x[p,1] & r_x[p,2] & \dots & r_x[p,p] \end{bmatrix} \begin{bmatrix} 1 \\ a_p[1] \\ a_p[2] \\ \vdots \\ a_p[p] \end{bmatrix} = \epsilon_{p,q} \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

$$\mathbf{R}_x \mathbf{a}_p = \epsilon_{p,q} \mathbf{u}_1$$

Once the $a_p[k]$ are found $b_q[k]$ are determined by

$e[n] = a_p[n] * x[n] - b_q[n] = 0$, or $b_q[n] = x[n] + \sum_{k=1}^{p} a_p[k]x[n-k], \; n = 0, 1, \dots, q$

Low-pass filter design using Prony's method

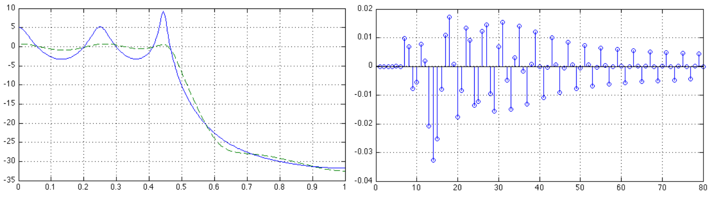
$$I(e^{j\omega}) = \begin{cases} e^{-jn_d\omega} & |\omega| < \pi/2 \\ 0 & otherwise \end{cases} \longleftrightarrow i[n] = \frac{\sin[(n-n_d)\pi/2]}{(n-n_d)\pi}$$

The filter will be designed with the orders $p = q = 5$ and the delay is $n_d = 5$.

$$\mathbf{a} = [1.0, -1.9167, 2.3923, -1.9769, 1.0537, -0.2970]^T$$
$$\mathbf{b} = [0.0637, -0.1220, 0.0462, 0.0775, 0.1316, 0.0807]^T$$

dashed : Prony Approximation
solid : Pade Approximation

Error function



The error of Prony is about 10-fold less than that of the Pade approximation.

MATLAB Scipt for Testing LP filter design using Prony's method

```
w = [0:0.01:pi]';
h = [0.00637,0,-0.1061,0.,0.3183,0.5,0.3183,0,-0.1061,0,0.0637];%sinc
a =[1.0, -1.9167, 2.3923, -1.9769, 1.0537, -0.2970]; %Prony
b =[0.0637, -0.1220, 0.0462, 0.0775, 0.1316, 0.0807]; %Prony
a0 = [1.0  -2.5256  3.6744  -3.4853  2.1307  -0.7034]; %Pade
b0 = [0.0637, -0.1608, 0.1280, 0.0461, 0.0638, 0.0211]; %Pade
H1 = freqz(h,1,w);
H2= freqz(b,a,w);
H0 =   freqz(b0,a0,w);
subplot(1,2,1);
plot(w/pi,20*log10(abs(H1)),w/pi,20*log10(abs(H0)),'-',w/pi,20*log10(abs(H2)
t = [0:79]';
x=zeros(80,1);
x(1) =1;
subplot(1,2,2)
x1 = filter(b,a,x);
e = 0.5*sinc((t-5)/2);
subplot(1,2,2)
stem(e-x1);grid
```

Shank's method

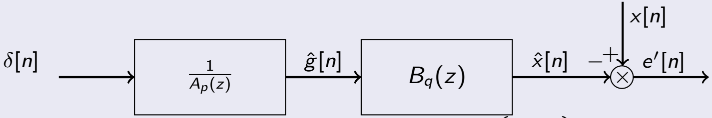
In Prony's method $b_q[n]$ was found by setting

$$e[n] = a_p[n] * x[n] - b_q[n] = 0 \text{ for } n = 0, 1, \ldots, q.$$

A better aproach is to perform least squares minimization over the model error

$$e'[n] = x[n] - \hat{x}[n]$$

over the entire data record.



$H(z)$ is a cascade of $B_q[z]$ and $A_p(z)$, $H(z) = B_q(z)\left\{\frac{1}{A_p(z)}\right\}$

Once $A_p(z)$ is determined, we can compute $g[n]$, the impulse response $1/A_p(z)$ by

$$g[n] = \delta[n] - \sum_{k=1}^{p} a_p[k]g[n-k] \text{ with } g[n] = 0 \text{ for } n < 0.$$

We then minimize $\mathcal{E}_S = \sum_{n=0}^{\infty} |e'[n]|^2$ where

$$e'[n] = x[n] - \hat{x}[n] = x[n] - \sum_{l=1}^{q} b_q[l]g[n-l]$$

$$\frac{\partial \mathcal{E}_S}{\partial b_q^*[k]} = \sum_{n=0}^{\infty} e'[n] \frac{\partial e'[n]^*}{\partial b_q^*[k]} = -\sum_{n=0}^{\infty} e'[n] g^*[n-k] = 0, \; k = 0, 1, \ldots, q$$

Substituting $e'[n]$ into the above equation

$$-\sum_{n=0}^{\infty} \left\{ x[n] - \sum_{l=0}^{q} b_q[l] g[n-l] \right\} g^*[n-k] \rightarrow$$

$$\sum_{l=0}^{q} b_q[l] \underbrace{\left[\sum_{n=0}^{\infty} g[n-l] g^*[n-k] \right]}_{r_g[k,l]} = \underbrace{\sum_{n=0}^{\infty} x[n] g^*[n-k]}_{r_{xg}[k]}, \; k = 0, 1, \ldots, q$$

$$\begin{bmatrix} r_g[0,0] & r_g[0,1] & r_g[0,2] & \ldots & r_g[0,q] \\ r_g[1,0] & r_g[1,1] & r_g[1,2] & \ldots & r_g[1,q] \\ r_g[2,0] & r_g[2,1] & r_g[2,2] & \ldots & r_g[2,q] \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ r_g[q,0] & r_g[q,1] & r_g[q,2] & \ldots & r_g[q,q] \end{bmatrix} \begin{bmatrix} b_q[0 \\ b_q[1] \\ b_q[2] \\ \vdots \\ b_q[q] \end{bmatrix} = \begin{bmatrix} r_{xg}[0] \\ r_{xg}[1] \\ r_{xg}[2] \\ \vdots \\ r_{xg}[q] \end{bmatrix},$$

On the other hand, $r_g[k+1, l+1] = \sum_{n=0}^{\infty} g[n-(l+1)] g^*[n-(k+1)]$

$$= \sum_{n=-1}^{\infty} g[n-l)] g^*[n-k)] + \underbrace{g[-1-l] g^*[-1-k]}_{0 \text{ for } k,l \geq 0} = r_g[k,l]$$

If $r_g[k+1, l+1] = r_g[k,l]$ then $r_g[k,l] = r_g[k-l]$ and the matrix equation becomes hermitian.

Hermitian matrix with Toeplitz structure

$$\begin{bmatrix} r_g[0] & r_g^*[1] & r_g^*[2] & \dots & r_g^*[q] \\ r_g[1] & r_g[0] & r_g^*[1] & \dots & r_g^*[q-1] \\ r_g[2] & r_g[1] & r_g[0] & \dots & r_g^*[q-2] \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ r_g[q] & r_g[q-1] & r_g[q-2] & \dots & r_g[0] \end{bmatrix} \begin{bmatrix} b_q[0 \\ b_q[1] \\ b_q[2] \\ \vdots \\ b_q[q] \end{bmatrix} = \begin{bmatrix} r_{xg}[0] \\ r_{xg}[1] \\ r_{xg}[2] \\ \vdots \\ r_{xg}[q] \end{bmatrix}$$

The minimum mean squared error $\mathcal{E}_S$ using the orthogonality is

$$\{\mathcal{E}_S\}_{\min} = \sum_{n=0}^{\infty} |e'[n]|^2 = \sum_{n=0}^{\infty} e'[n]x^*[n] = \sum_{n=0}^{\infty} |x[n]|^2 - \sum_{k=0}^{q} b_q[k]\Big[\sum_{n=0}^{\infty} g[n-k]x^*[n]\Big]$$

$$\{\mathcal{E}_S\}_{\min} = r_x[0] - \sum_{k=0}^{q} b_q[k]r_{gx}[-k] = r_x[0] - \sum_{k=0}^{q} b_q[k]r_{xg}^*[k]$$

All pole modeling using Prony's method

$$H(z) = \frac{b(0)}{1+\sum\limits_{k=1}^{p} a_p[k]z^{-k}}$$

$a_p[k]$ are found by minimizing $\mathcal{E}_{p,0} = \sum\limits_{n=1}^{\infty} |e[n]|^2$ where $e[n] = x[n] + \sum\limits_{k=1}^{p} a_p[k]x[n-k]$

$x[n] =$ for $n < 0$, $e[0] = x[0]$

so $\mathcal{E}_p = \sum\limits_{n=0}^{\infty} |e[n]|^2$ will be minimized instead of $\mathcal{E}_{p,0}$ *i.e.* since $\mathcal{E}_p$ is not effected by $a_p[k]$.

$\frac{\partial \mathcal{E}_p}{\partial a_p^*[k]} = 0$ leads to $\sum\limits_{l=1}^{p} a_p[l]r_x[k,l] = -r_x[k,0]$, $k = 1, \ldots, p$

$r_x[k,l] = \sum\limits_{n=0}^{\infty} x[n-l]x^*[n-k]$.

Notice that $n = 0$ instead of $n = 1$.

$$r_x[k+1, l+1] = \sum_{n=0}^{\infty} x[n-(l+1)]x^*[n-(k+1)]$$

$$= \sum_{n=-1}^{\infty} x[n-l)]x^*[n-k)] + \underbrace{x[-1-l]x^*[-1-k]}_{0 \text{ for } k,l \geq 0} = r_x[k,l]$$

$$r_x[k,l] = r_x[k-l] = \sum_{n=0}^{\infty} x[n-l]x^*[n-k]$$

All pole normal equations

$$\begin{bmatrix} r_x[0] & r_x^*[1] & r_x^*[2] & \dots & r_x^*[p-1] \\ r_x[1] & r_x[0] & r_x^*[1] & \dots & r_x^*[p-2] \\ r_x[2] & r_x[1] & r_x[0] & \dots & r_x^*[p-3] \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ r_x[p-1] & r_x[p-2] & r_x[p-3] & \dots & r_x[0] \end{bmatrix} \begin{bmatrix} a_p[0] \\ a_p[1] \\ a_p[2] \\ \vdots \\ a_p[p] \end{bmatrix} = - \begin{bmatrix} r_x[1] \\ r_x[2] \\ r_x[3] \\ \vdots \\ r_x[p] \end{bmatrix}$$

$$\{\mathcal{E}_p\}_{\min} = r_x[0] - \sum_{k=0}^{p} a_p[k] r_x^*[k] = r_x[0] - \sum_{k=0}^{p} a_p[k] r_x[k]$$

If $= b^2(0)$ is chosen such that $b(0) = \sqrt{\{\mathcal{E}_p\}_{\min}}$, then it can be shown that energy in $x[n]$ is equal to the energy in $\hat{x}[n] = h[n]$ *i.e.* $r_x[0] = r_h[0]$.

Example: Modeling the signal $x[n] = \delta[n] - \delta[n-1]$

The autocorrelation sequence is $r_x[k] = \begin{cases} 2 & k = 0 \\ -1 & k = \pm 1 \\ 0 & otherwise \end{cases}$

The normal equation is $r_x[0]a[1] = -r_x[1] \longrightarrow a[1] = -\frac{r_x[1]}{r_x[0]} = \frac{1}{2}$

The modeling error is $\{\mathcal{E}_p\}_{\min} = \epsilon = r_x[0] + a[1]r_x[1] = 1.5$

Energy matching constraint yields $b[0] = \sqrt{1.5} = 1.22$ and

$$H(z) = \frac{1.22}{1+0.5z^{-1}}$$

The normal equations for the second order model

$$\begin{bmatrix} r_x[0] & r_x[1] \\ r_x[1] & r_x[0] \end{bmatrix} \begin{bmatrix} a[1] \\ a[2] \end{bmatrix} = -\begin{bmatrix} r_x[1] \\ r_x[2 \end{bmatrix} \longrightarrow \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} a[1] \\ a[2] \end{bmatrix} = -\begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

The all pole filter parameters are $a[1] = 2/3$, $a[2] = 1/3$ and the modeling error is $\epsilon = r_x[0] + a[1]r_x[1] + a[2]r_x[2] = 2 - 2/3 = 4/3$ and $b[0] = \sqrt{4/3} = 1.15$

$$H(z) = \frac{1.15}{1+\frac{2}{3}z^{-1}+\frac{1}{3}z^{-2}}$$

FIR Least squares inverse filters

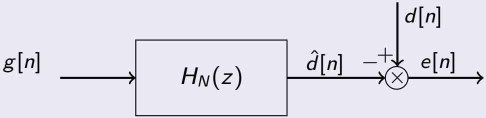
Given a filter $g[n]$ how can we determine its inverse filter $g\hat{[n]}$?

$$g[n] * g\hat{[n]} = \delta \text{ or } G(z)\hat{G}^{-1}(z) = 1$$

$$\hat{G}^{-1}(z) \approx H(z) \text{ so that } G(z)H(z) \approx 1$$

An FIR filter $h_N[n]$ such that $h_N[n] * g[n] = d[n] = \delta[n]$

$$e[n] = d[n] - h_N[n] * g[n] = d[n] - \sum_{l=0}^{N-1} h_N[l]g[n-l]$$



$$\mathcal{E}_N = \sum_{n=0}^{\infty} |e[n]|^2 = \sum_{n=0}^{\infty} |d[n] - \sum_{l=0}^{N-1} h_n[l]g[n-l]|^2 \rightarrow \text{Shank's method}$$

Optimum least squares filter

$$\sum_{l=0}^{N-1} h_n[l]r_g[k-l] = r_{dg}[k],\ k = 0, 1, \ldots, N-1$$

$$r_g[k-l] = \sum_{l=0}^{\infty} g[n-l]g^*[n-k] = r_g^*[l-k] \text{ and } r_{dg}[k] = \sum_{n=0}^{\infty} d[n]g^*[n-k]$$

The normal equations

$$\begin{bmatrix} r_g[0] & r_g^*[1] & r_g^*[2] & \cdots & r_g^*[p-1] \\ r_g[1] & r_g[0] & r_g^*[1] & \cdots & r_g^*[p-2] \\ r_g[2] & r_g[1] & r_g[0] & \cdots & r_g^*[p-3] \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ r_g[N-1] & r_g[N-2] & r_g[N-3] & \cdots & r_g[0] \end{bmatrix} \begin{bmatrix} h_N[0] \\ h_N[1] \\ h_N[2] \\ \vdots \\ h_N[p] \end{bmatrix} = \begin{bmatrix} g^*[0] \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

The minimum squared error is

$$\{\mathcal{E}_p\}_{\min} = r_d[0] - \sum_{k=0}^{N-1} h_n[k] r_{dg}^*[k]$$

Since $r_{dg}[k] = g^*[0]\delta[k]$

$$\{\mathcal{E}_p\}_{\min} = 1 - h_n[0]g[0]$$

Example: FIR least squares inverse filter for $g[n] = \delta[n] - \alpha\delta[n-1]$, $|\alpha| < 1$

$g[n] = \delta[n] - \alpha\delta[n-1] \rightarrow G(z) = 1 - \alpha z^{-1}$

$G^{-1}(z) = \frac{1}{1-\alpha z^{-1}} \rightarrow g^{-1}[n] = \alpha^n u[n]$ $\qquad r_g[k] = \begin{cases} 1+\alpha^2 & k = 0 \\ -\alpha & k \pm 1 \\ 0 & otherwise \end{cases}$

The normal equations for $N = 2$

$$\begin{bmatrix} 1+\alpha^2 & -\alpha \\ -\alpha & 1+\alpha^2 \end{bmatrix} \begin{bmatrix} h[0] \\ h[1] \end{bmatrix} = -\begin{bmatrix} 1 \\ 0 \end{bmatrix} \longrightarrow \begin{bmatrix} h[0] \\ h[1] \end{bmatrix} = -\begin{bmatrix} \frac{1+\alpha^2}{1+\alpha^2+\alpha^4} \\ \frac{\alpha}{1+\alpha^2+\alpha^4} \end{bmatrix}$$

$$H(z) = \frac{1+\alpha^2}{1+\alpha^2+\alpha^4} + \frac{\alpha}{1+\alpha^2+\alpha^4} z^{-1}$$

The normal equations for length N

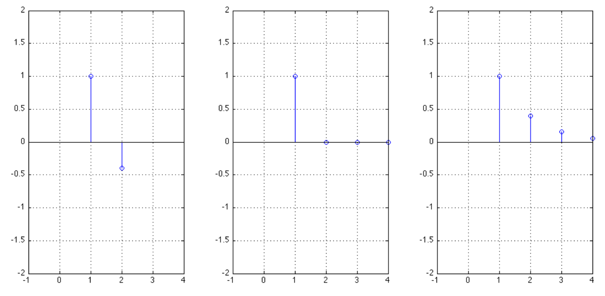
$$\begin{bmatrix} 1+\alpha^2 & -\alpha & 0 & \dots & 0 \\ -\alpha & 1+\alpha^2 & -\alpha & \dots & 0 \\ 0 & -\alpha & 1+\alpha^2 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & -\alpha & \dots & 1+\alpha^2 \end{bmatrix} \begin{bmatrix} h_N[0] \\ h_N[1] \\ h_N[2] \\ \vdots \\ h_N[N-1] \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Solving the recursive equation $\alpha h_N[n-1] + (1+\alpha^2)h_N[n] - \alpha h_N[n+1] = 0$ we get $h_n[n] = c_1\alpha^n + c_2\alpha^{-n}$. The boundary conditions are $(1+\alpha^2)h_N[0] - \alpha h_N[1] = 1$ and $-\alpha h_N[N-2] + (1+\alpha^2)h_N[N-1] = 0$ which yield $c_1 = \alpha^{-(N-1)}/(\alpha^{-(N-1)} - \alpha^{N+3})$ and $c_1 = -\alpha^{N+1}/(\alpha^{-(N-1)} - \alpha^{N+3})$

$$h_N[n] = \begin{cases} (\alpha^{n-N} - \alpha^{N-n})/(\alpha^{-N} - \alpha^{N+2}) & 0 \le n \le N-1 \\ 0 & else \end{cases}$$

MATLAB script for Testing FIR LS Inverse Filter

```
a=0.4;
g = [ 1 -a];
N= 4;
n=[0:N-1]';
h = (a.^(n-N) - a.^(N-n) )./ (a.^(-N) - a.^(N+2));
subplot(1,3,1);stem(g);grid;axis([ -1 4 -2 2] )
subplot(1,3,2);stem(conv(h,g));grid;axis([ -1 4 -2 2] )
subplot(1,3,3);stem(h);grid;axis([ -1 4 -2 2] )
```



Autoregressive Moving Average (ARMA) Models

$$v(n) \longrightarrow \boxed{H(z) = \frac{B_q(z)}{A_p(z)}} \longrightarrow \hat{x}(n)$$

$H(z) = \frac{B_q(z)}{A_p(z)} = \frac{\sum_{k=0}^{q} b_q(k) z^{-k}}{1+\sum_{k=1}^{p} a_p(k) z^{-k}}$ and $v(n)$ is a unit variance white noise.

$$\xi_{MS} = E\left\{|x(n) - \hat{x}(n)|^2\right\}$$

$$H(z)A_p(z) = B_q(z) \longrightarrow h(n) + \sum_{k=1}^{p} a_p(k)h(n-k) = b_q(n)$$

$$x[n] + \sum_{l=1}^{p} a_p(l)x(n-l) = \sum_{l=0}^{q} b_q(l)v(n-l)$$

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

$$r_x(k) + \sum_{l=1}^{p} a_p(l)r_x(k-l) = \sum_{l=0}^{q} b_q(l)E\left\{v(n-l)x^*(k-l)\right\}$$

$$x(n) = h(n) * v(n)$$

$$E\left\{v(n-l)x^*(n-k)\right\} = E\left\{\sum_{m=-\infty}^{\infty} v(n-l)v^*(m)\right\} h^*(n-k-m)$$

$$\sum_{m=-\infty}^{\infty} E\left\{v(n-l)v^*(m)\right\} h^*(n-k-m) = \sigma_v^2 h^*(l-k)$$

$$r_x(k) + \sum_{l=1}^{p} a_p(l) r_x(k-l) = \sigma_v^2 \sum_{l=0}^{q} b_q(l) h^*(l-k)$$

$$c_q(k) = \sum_{l=0}^{q-k} b_q(l+k) h^*(l) = \sum_{l=0}^{q} b_q(l) h^*(l-k)$$

Modified Yule-Walker Equations

$$r_x(k) + \sum_{l=1}^{p} a_p(l) r_x(k-l) = \begin{cases} \sigma_2^2 c_q(k) & k = 0, 1, \ldots, q \\ 0 & k > q \end{cases}$$

$$
\left[\begin{array}{cccc}
r_x(0) & r_x(-1) & \cdots & r_x(-p) \\
r_x(1) & r_x(0) & \cdots & r_x(-p+1) \\
\vdots & \vdots & \vdots & \vdots \\
r_x(q) & r_x(q-1) & \cdots & r_x(q-p) \\
\hline
r_x(q+1) & r_x(q) & \cdots & r_x(q-p+1) \\
\vdots & \vdots & \vdots & \vdots \\
r_x(q+p) & r_x(q+p-1) & \cdots & r_x(q)
\end{array}\right]
\left[\begin{array}{c}
1 \\ a_p[1] \\ a_p[2] \\ \vdots \\ a_p[p]
\end{array}\right]
= \sigma_v^2
\left[\begin{array}{c}
c_q(0) \\ c_q(1) \\ \vdots \\ c_q(q) \\ \hline 0 \\ \vdots \\ 0
\end{array}\right]
$$

$$
c_q(k) = b_q(k) * h^*(-k) = \sum_{l=0}^{q-k} b_q(l+k) h^*(l)
$$

1. The AR coefficients are determined using

$$
\left[\begin{array}{cccc}
r_x(q) & r_x(q-1) & \cdots & r_x(q-p+1) \\
r_x(q+1) & r_x(q) & \cdots & r_x(q-p+2) \\
\vdots & \vdots & \vdots & \vdots \\
r_x(q+p-1) & r_x(q+p-2) & \cdots & r_x(q)
\end{array}\right]
\left[\begin{array}{c}
a_p[1] \\ a_p[2] \\ \vdots \\ a_p[p]
\end{array}\right]
= -
\left[\begin{array}{c}
r_x(q+1) \\ r_x(q+2) \\ \vdots \\ r_x(q+p)
\end{array}\right]
$$

2. $$
\left[\begin{array}{cccc}
r_x(0) & r_x^*(1) & \cdots & r_x^*(p) \\
r_x(1) & r_x(0) & \cdots & r_x^*(p-1) \\
\vdots & \vdots & \vdots & \vdots \\
r_x(q) & r_x(q+1) & \cdots & r_x(0)
\end{array}\right]
\left[\begin{array}{c}
1 \\ a_p[1] \\ a_p[2] \\ \vdots \\ a_p[p]
\end{array}\right]
= -
\left[\begin{array}{c}
c_q(0) \\ c_q(1) \\ \vdots \\ c_q(q)
\end{array}\right]
$$

$$\text{Positive part of } [C_q(z)]_+ = \sum_{k=0}^{\infty} c_q(k) z^{-k}$$

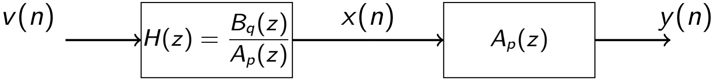
$$\text{Negative part of } [C_q(z)]_- = \sum_{k=-\infty}^{-1} c_q(k) z^{-k} = \sum_{k=1}^{\infty} c_q(-k) z$$

$$C_q(z) = B_q(z) H^*(1/z^*) = B_q(z) B_q^*(1/z^*)/A_p^*(1/z^*)$$

$$P_y(z) = C_q(z) A_p^*(1/z^*) = [C_q(z)]_+ A_p^*(1/z^*) + [C_q(z)]_- A_p^*(1/z^*)$$

$$\text{Causal part of } P_y(z) \longrightarrow [P_y(z)]_+ = \Big[[C_q(z)]_+ A_p^*(1/z^*) \Big]_+$$

$$P_x(z) = \frac{B_q(z) B_q^*(1/z^*)}{A_p(z) A_p^*(1/z^*)} \qquad P_y(z) = B_q(z) B_q^*(1/z^*)$$



ARMA(1,1) $r_x(0) = 26$; $r_x(1) = 7$; $r_x(2) = 7/2$

$$\begin{bmatrix} r_x(0) & r_x(1) \\ r_x(1) & r_x(0) \\ r_x(2) & r_x(1) \end{bmatrix} \begin{bmatrix} 1 \\ a_1(p) \end{bmatrix} = \begin{bmatrix} c_1(0) \\ c_1(1) \\ 0 \end{bmatrix} \quad a_1(1) = -1/2$$

$$\begin{bmatrix} r_x(0) & r_x(1) \\ r_x(1) & r_x(0) \\ r_x(2) & r_x(1) \end{bmatrix} \begin{bmatrix} 1 \\ a_1(1) \end{bmatrix} = \begin{bmatrix} c_1(0) \\ c_1(1) \\ 0 \end{bmatrix}, \begin{bmatrix} 26 & 7 \\ 7 & 26 \end{bmatrix} \begin{bmatrix} 1 \\ -1/2 \end{bmatrix} = \begin{bmatrix} \frac{45}{2} \\ -6 \end{bmatrix}$$

$[C_1(z)]_+ = 45/2 - 6z^{-1}$ and $A_1^*(1/z^*) = 1 - 0.5z = -45/4z + 51/2 - 6z^{-1}$

$[P_y(z)]_+ = \left[[C_1(z)]_+ A_1^*(1/z^*)\right]_+ = \frac{51}{2} - 6z^{-1}$

Using symmetry of $P_y(z)$

$[C_1(z)]_+ A_1^*(1/z^*) = B_1(z)B_1^*(1/z^*) = -6z + \frac{51}{2} - 6z^{-1} = 24(1 - \frac{1}{4}z^{-1})(1 - \frac{1}{4}z)$

$$H(z) = 2\sqrt{6}\frac{1-0.25z^{-1}}{1-0.5z^{-1}}$$

Autoregressive (AR) Models

$$H(z) = \frac{b(0)}{1+\sum\limits_{k=1}^{p} a_p(k) z^{-k}}$$

Yule-Walker Equations for AR process

$$r_x(k) + \sum_{l=1}^{p} a_p(l) r_x(k-l) = |b(0)|^2 \delta(k) \qquad k \geq 0$$

$$\begin{bmatrix} r_x(0) & r_x(-1) & \dots & r_x(-p+1) \\ r_x(1) & r_x(0) & \dots & r_x(-p+2) \\ \vdots & \vdots & \vdots & \vdots \\ r_x(p-1) & r_x(p-2) & \dots & r_x(0) \end{bmatrix} \begin{bmatrix} a_p[1] \\ a_p[2] \\ \vdots \\ a_p[p] \end{bmatrix} = - \begin{bmatrix} r_x(1) \\ r_x(2) \\ \vdots \\ r_x(p) \end{bmatrix}$$

$$|b(0)|^2 = r_x(0) + \sum_{k=1}^{p} a_p(k) r_x(k)$$

For a given sample of $x(n)$, $\hat{r}_x(k) = \frac{1}{N} \sum\limits_{n=0}^{N-1} x(n) x(n-k)$

Moving Average (MA) Models

$$x(n) = \sum_{k=0}^{q} b_q(k) v(n-k)$$

Yule-Walker Equations for MA process

$$r_x(k) = b_q(k) * b_q^*(-k) = \sum_{l=0}^{q-|k|} b_q(l+|k|) b_q^*(l)$$

$$P_x(z) = \sum_{k=-q}^{q} r_x(k) z^{-k} = B_q(z) B_q^*(1/z^*) = |b(0)|^2 \prod_{k=1}^{q} (1-\beta_k z^{-1}) \prod_{k=1}^{q} (1-\beta_k^* z)$$

Using Spectral Factorization

$$P_x(z) = \sigma_0^2 Q(z) Q^*(1/z^*) = \sigma_0^2 \prod_{k=1}^{q} (1-\alpha_k z^{-1}) \prod_{k=1}^{q} (1-\alpha_k^* z)$$

where $Q(z)$ is a minimum phase, monic polynomial and $\sigma_0 = b(0)$

$$H(z) = \sigma_0 \sum_{k=0}^{q} q(k) z^{-k}$$

MA process with $r_x(k) = 17\delta(k) + 4[\delta(k-1) + 4\delta(k+1)]$

$$P_x(z) = 17 + 4z^{-1} + 4z = [z^{-2} + 17/4z^{-1} + 1]4z = (z^{-1} + 1/4)(z^{-1} + 4)4z$$
$$= (z^{-1} + 4)(4 + z) = \sigma_0^2 B(z) B^*(1/z^*)$$
$$\sigma_0 = 1 \text{ and } H(z) = B(z) = 4 + z^{-1}$$

Durbin's Method

An MA process is $B_q(z) = \sum_{k=0}^{q} b_q(k) z^{-k}$ so that $x(n) = \sum_{k=0}^{q} b_q(k) w(n-k)$ where $w(n)$ is a white noise.

$$B_q(z) \approx \frac{1}{A_p(z)} = \frac{1}{a_p(0) + \sum_{k=1}^{p} a_p(k) z^{-k}}$$

For example, if $B_1(z) = b(0) - b(1)z^{-1}$ where $|b(0)| > |b(1)|$ then

$$\frac{1}{B_0(z)} = \frac{1}{b(0) - b(1)z^{-1}} = \frac{1}{b(0)} \sum_{k=0}^{\infty} \beta^k z^{-k} \text{ where } \beta = \frac{b(1)}{b(0)}.$$

This means that $B_1(z)$ can be approximated by the expansion

$$B_1(z) \approx \frac{b(0)}{1 + \beta z^{-1} + \ldots + \beta z^{-p}}$$

Once a high order all pole model for $x(n)$ is found, the MA coefficients can be computed from all pole coeffcients $a_p(k)$.

Since $A_p(z) \approx \frac{1}{B_q(z)} = \frac{1}{b_q(0) + \sum_{k=1}^{q} b_q(k) z^{-k}}$, an all pole model with order q for $a_p(k)$ will yield $b_q(k)$.

1. Given $x(n)$ for $n = 0, 1, \ldots, N-1$, find a p^{th} order AR model for $x(n)$ with order $> 4q$ and normalize the coefficients so that the gain term is unity.
2. Using the AR coefficients in Step 1 as input data, fit a q^{th} order AR model which will yield the MA coefficients after normalized by gain term.

$X(z) = 1 - \beta z^{-1}$ and $|\beta| < 1$

The normal equation for an AR(1) model $\frac{b(0)}{1+a_1(1)z^{-1}}$ for a sequence of length N is

$$\left[\; r_x(0) \;\right] \left[\; a_1[1] \;\right] = - \left[\; r_x(1) \;\right] \longrightarrow a(1) = -r_x(1)/r_x(0)$$

$$r_x(k) = \sum_{n=k}^{N} x(n)x^*(n-k) \qquad \sum_{n=k}^{N} \beta^n(\beta^*)^{n-k} = \beta^k \frac{1-|\beta|^{2(N-k+1)}}{1-|\beta|^2}$$

$$r_x(1) = \beta \frac{1-|\beta|^{2N}}{1-|\beta|^2} \text{ and } r_x(0) = \frac{1-|\beta|^{2N+2}}{1-|\beta|^2} \longrightarrow a(1) = -\beta \frac{1-|\beta|^{2N}}{1-|\beta|^{2N+2}}$$

and $|b(0|^2 = r_x(0) + a_1(1)r_x(1) = \frac{1-|\beta|^{4N+2}}{1-|\beta|^{2(N+1)}}$

$$A_p(z) \approx \frac{1}{X(z)} = \frac{1}{1-\beta z^{-1}} = \sum_{k=0}^{\infty} \beta^k z^{-k} \longrightarrow a_p(k) = \beta^k = x(k)$$

Then all pole model for sequence $x(n)$ for $p >> 1$ is

$$X(z) = \frac{1}{A_p(z)} = \frac{1}{1+\beta z^{-1}+\ldots+\beta^p z^{-p}}$$

A 1st order AR model for the sequence $a_p(k) = \beta^k$ with length p is

$$H(z) = \frac{b(0)}{1+a(1)z^{-1}} \longrightarrow a_1(1) = -r_x(1)/r_x(0)$$

$a(1) = -\beta \frac{1-|\beta|^{2p}}{1-|\beta|^{2p+2}}$ and $\lim_{p\to\infty} a(1) = -\beta$ and $|b(0)|^2 = r_x(0) + a(1)r_x^*(1) = \frac{1-|\beta|^{4p+2}}{1-|\beta|^{2p+2}}$

and $\lim_{p\to\infty} b(0) = 1$

$$B_1(z) = 1 - \beta z^{-1}$$