Linear Predictors

Table of Contents

Forward Linear Predictors

Problem: we want predict the current signal value $\hat{x}[n]$ given a set of p previous samples of a wide-sense stationary process:

$$\hat{x}[n] = \sum_{k=1}^{p} h_k x[n-k] = \boldsymbol{h}^{\mathrm{T}} \boldsymbol{x}[n-1], \qquad (14.122)$$

This is called one-step *forward linear predictor.*

The normal equations for the optimum linear predictor are:

$$\boldsymbol{R}\boldsymbol{h} = \boldsymbol{r}, \qquad (14.125)$$

where $\boldsymbol{R}$, which is a symmetric Toeplitz matrix, and the vector $\boldsymbol{r}$ are defined by

$$\boldsymbol{R} \triangleq \begin{bmatrix} r[0] & r[1] & \dots & r[p-1] \\ r[1] & r[0] & \dots & r[p-2] \\ \vdots & \vdots & \ddots & \vdots \\ r[p-1] & r[p-2] & \dots & r[0] \end{bmatrix} \quad \text{and} \quad \boldsymbol{r} \triangleq \begin{bmatrix} r[1] \\ r[2] \\ \vdots \\ r[p] \end{bmatrix}. \qquad (14.126)$$

The minimum squared error is given by Eq. 14.127:

$$J_0 = r[0] - \boldsymbol{h}^{\mathrm{T}}\boldsymbol{r} = r[0] - \boldsymbol{r}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{r}. \qquad (14.127)$$

All-pole signal modelling

Suppose the input signal $x[n]$ is an AR(p) model i.e.:

$$x[n] = -\sum_{k=1}^{p} a_k x[n-k] + z[n] = -\boldsymbol{a}^{\mathrm{T}}\boldsymbol{x}[n-1] + z[n], \qquad (14.128)$$

where $z[n] \sim \mathrm{WN}(0, \sigma_z^2)$

The value that we want to predict $x[n]$ can written as:

$$x[n] = \sum_{k=1}^{p} h_k x[n-k] + e[n] = \boldsymbol{h}^{\mathrm{T}}\boldsymbol{x}[n-1] + e[n]. \qquad (14.131)$$

Setting $\boldsymbol{h} = -\boldsymbol{a}$, the prediction error can be expressed as:

$$e[n] = x[n] + \sum_{k=1}^{p} a_k x[n-k] = x[n] + \boldsymbol{a}^{\mathrm{T}}\boldsymbol{x}[n-1], \qquad (14.132)$$

This shows that the prediction error $e[n]$ is the output of a filter with the following system function:

$$A(z) = 1 + \sum_{k=1}^{p} a_k z^{-k}. \qquad (14.133)$$

This system $A(z)$ is known as the ***prediction error filter*** or the ***analysis filter***.

If the input $x[n]$ is an AR(p) process, the output of $A(z)$ is a white noise process $z[n]$

[✔] ADSI Problem 6.4: Linear interpolation, estimate missing samples

Sometimes it happens that a datapoint is missing from some signal acquistion due to sensor failure, transmission errors etc. Assume that we have a long stationary sequence $\{x[n]\}_{n=0}^{N-1}$ where the j'th sample is missing i.e.

$$\{x[n]\} = \{x[0],\ x[1],\ \ldots\ x[j-1]\ x[j+1],\ x[j+2],\ \ldots x[N-2],\ x[N-1]\}$$

We want estimate the missing datapoint as a linear combination of the two neighbouring samples

$$\hat{x}[j] = c_1 x[j-1] + c_2 x[j+1]$$

1. Use our standard mean square error approach to derive equations for c_1 and c_2 based on the autocorrelation $r_{xx}(l)$.

1) Use mean squared error to derive coefficients based on the ACRS

We want to estimate a missing sample $x(j)$ as a linar combination of two neighbouring samples:

$$\hat{x}(j) = c_1 x(j-1) + c_2 x(j+1)$$

To find the coefficients c_1 and c_2 using the mean squared error method, we need to take following steps:

1. Find an expression for the error: $e(n)$
2. Square the error quantity: $e^2(n)$
3. Take the expectation of squared error: $E[e^2(n)]$ to find the Mean Squared Error
4. Find the minimum by setting the partial derivatives of MSE to zero and solving the equation

Step 1: The error of the estimate is:

$$e(j) = x(j) - \hat{x}(j)$$

$$e(j) = x(j) - (c_1 x(j-1) + c_2 x(j+1))$$

$$e(j) = x(j) - c_1 x(j-1) - c_2 x(j+1)$$

Step 2: square the error

$$e^2(j) = (x(j) - c_1 x(j-1) - c_2 x(j+1))(x(j) - c_1 x(j-1) - c_2 x(j+1))$$

```
syms c1 c2 x_j x_jm1 x_jp1
expand((x_j - c1*x_jm1 - c2*x_jp1)^2)
```

ans = $c_1^2 x_{\mathrm{jm1}}^2 + 2\,c_1\,c_2\,x_{\mathrm{jm1}}\,x_{\mathrm{jp1}} - 2\,c_1\,x_j\,x_{\mathrm{jm1}} + c_2^2\,x_{\mathrm{jp1}}^2 - 2\,c_2\,x_j\,x_{\mathrm{jp1}} + x_j^2$

$$e^2(j) = c_1^2 x^2(j-1) + 2c_1c_2x(j-1)x(j+1) - 2c_1x(j)x(j-1) + c_2^2x^2(j+1) - 2c_2x(j)x(j+1) + x^2(j)$$

Step 3: Take the expectation of the squared error:

$$E[e^2(j)] = E\left[c_1^2 x^2(j-1) + 2c_1c_2x(j-1)x(j+1) - 2c_1x(j)x(j-1) + c_2^2x^2(j+1) - 2c_2x(j)x(j+1) + x^2(j)\right]$$

$$E[e^2(j)] = c_1^2 E[x^2(j-1)] + 2c_1c_2E[x(j-1)x(j+1)] - 2c_1E[x(j)x(j-l)]$$

$$+c_2^2E[x^2(j+1)] - 2c_2E[x(j)x(j+1)] + E[x^2(j)]$$

From the expression, we observe different autocorrelation values:

$$\begin{aligned} E\left[e^2[n]\right] = & r_x[0] - c_1 r_x[1] - c_2 r_x[1] \\ & - c_1 r_x[1] + c_1^2 r_x[0] + c_1 c_2 r_x[2] \\ & - c_2 r_x[1] + c_1 c_2 r_x[2] + c_2^2 r_x[0] \end{aligned}$$

Step 4: The minimum is found by taking the partial derivatives, setting it zero and solving the two equations:

$$\frac{\partial E\left[e^2[n]\right]}{\partial c_1} = 0, \quad \frac{\partial E\left[e^2[n]\right]}{\partial c_2} = 0,$$

The partial derivatives become:

$$\begin{aligned} & -2r_x[1] + 2c_1 r_x[0] + 2c_2 r_x[2] = 0 \\ & -2r_x[1] + 2c_1 r_x[2] + 2c_2 r_x[0] = 0 \end{aligned}$$

Which can be written as a matrix equation:

$$\left[\begin{array}{cc} r_x[0] & r_x[2] \\ r_x[2] & r_x[0] \end{array}\right]\left[\begin{array}{c} c_1 \\ c_2 \end{array}\right] = \left[\begin{array}{c} r_x[1] \\ r_x[1] \end{array}\right]$$

Working through the equations the solution is:

$$c_1 = c_2 = \frac{r_x[1](r_x[0] - r_x[2])}{r_x^2[0] - r_x^2[2]}$$

[✔] ADSI Problem 6.6: Levinson-Durbin and linear prediction

The autocorrelation function of an AR(2) process with two complex conjugated poles at $p = r_p e^{\pm j\omega_p}$ can be calculated analytically and is given by

$$r_{xx}(l) = \frac{r_p^l\left(\sin((l+1)\omega_p) - r_p^2\sin((l-1)\omega_p)\right)}{(1-r_p^2)\sin(\omega_p)(1-2r_p^2\cos(2\omega_p)+r_p^4)} \quad \text{for } l \geq 0$$

Assume that $r_p = 0.9$ and $\omega_p = \pi/16$.

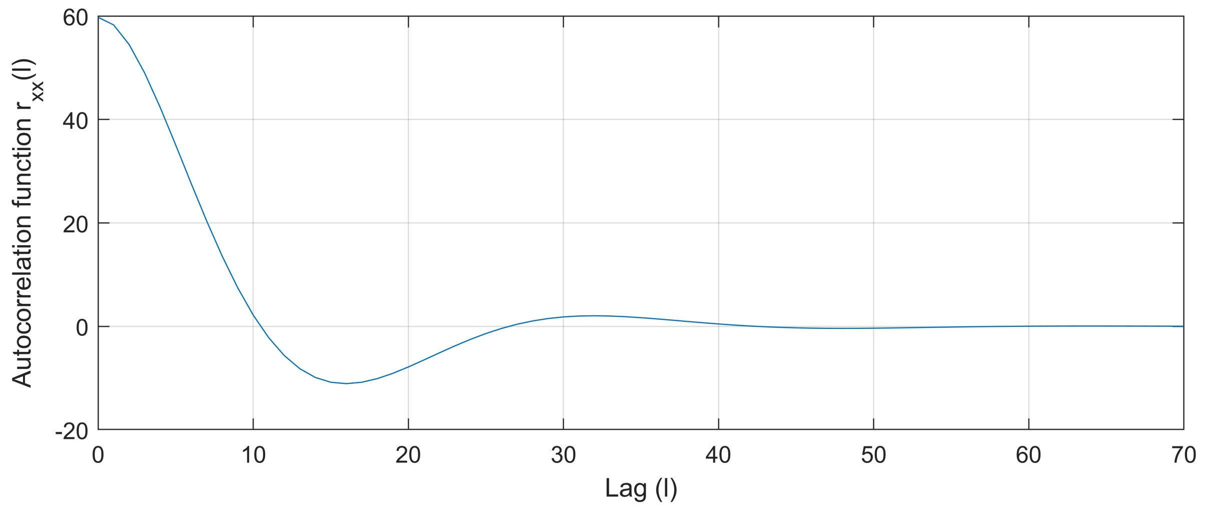
[✓] 1) Plot the autocorrelation function.

```
rp = 0.9;
wp = pi/16;

l = 0:70;

nominator = rp.^l.* (sin((l + 1)*wp) - rp^2*sin((l - 1)*wp));
denominator = (1 - rp^2) * sin(wp) * (1 - 2*rp^2*cos(2*wp) + rp^4);
r_xx = nominator / denominator;

figure('position', [0, 0, 800, 300])
plot(l, r_xx)
xlabel('Lag (l)')
ylabel('Autocorrelation function r_{xx}(l)')
grid on;
```



[✓] 2) Compute reflection coefficients for m'th order optimum linear predictors

Use the above autocorrelation function and the Levinson-Durbin recursion to calculate reflection coefficients and minimum mean square errors for m'th order optimum linear predictors for $m = 1$ to $m = 6$. Are the results in agreement with your anticipations and Eq. (14.149)?

First, we use the MATLAB function `levinson(r_xx, m)` to compute the reflection coefficients and the corresponding errors for the different m'th order linear predictors:

```
M = 6;
err = zeros(M, 1); % Array to store the errors

for m=1:M
    [a, e, k] = levinson(r_xx, m);
    % `a` is the coefficients of the AR(m) model
    % `e` is the prediction error of m'th order optimimum linear predictor
    % `k` is the reflection coefficients of the linear predictor
    err(m) = e;
    display(strcat('Coefficients for a', 32, num2str(m), '-th order predictor', 32))
    display(k)
end
```

```
Coefficients for a 1-th order predictor
k = -0.9754
Coefficients for a 2-th order predictor
k = 2×1
   -0.9754
    0.8100
Coefficients for a 3-th order predictor
k = 3×1
   -0.9754
    0.8100
   -0.0000
Coefficients for a 4-th order predictor
k = 4×1
   -0.9754
    0.8100
   -0.0000
   -0.0000
Coefficients for a 5-th order predictor
k = 5×1
   -0.9754
    0.8100
   -0.0000
   -0.0000
    0.0000
Coefficients for a 6-th order predictor
k = 6×1
   -0.9754
    0.8100
   -0.0000
   -0.0000
    0.0000
   -0.0000
```

It is important to note that the reflection coefficients $\{k_3, k_4, k_5, k_6\}$ are zero. This is to be expected as the reflection coefficients is dependent on the AR(q) model. Since we have an AR(2) model, the corresponding reflection coefficients become zero.

Let us consider the error of each of the predictors. Eq. 14.149 states that the prediction error can be calculated based on the previous error and current reflection coefficient:

$$J_{m+1} = J_m + \beta_{m+1} k_{m+1} = (1 - k_{m+1}^2) J_m. \qquad (14.149)$$

where $J_0 = r_{xx}(0)$ and $\beta_1 = r_{xx}(1)$

The squared error should be 1 once the order of the predictor matches or exceeds the AR(q) model.

```
J0 = r_xx(1)
```

```
J0 = 59.7579
```

```
J1 = (1 - k(1)^2) * J0
```

```
J1 = 2.9078
```

```
J2 = (1 - k(2)^2) * J1
```

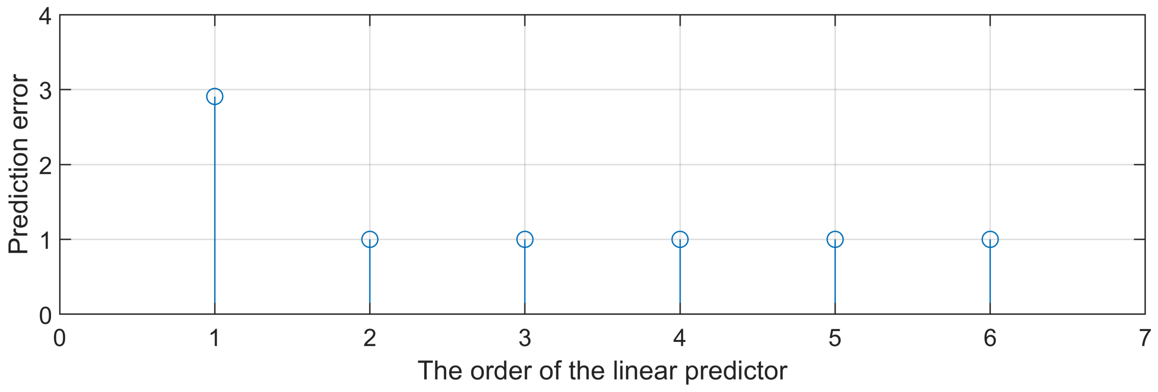
```
J2 = 1.0000
```

```
J3 = (1 - k(3)^2) * J2
```

```
J3 = 1.0000
```

Let us plot the order of the linear predictor versus the prediction error for an AR(2) model.

```
figure('position', [0, 0, 700, 200])
stem(err)
xlabel('The order of the linear predictor')
ylabel('Prediction error')
xlim([0, M+1])
ylim([0, 4])
grid on;
```



[?] ADSI Problem 6.7: Linear prediction of a sine signal

This problem addresses linear prediction on a simple harmonic signal where the results can be compared with our intuitive understanding.

Let a discrete time signal be given by

$$x(n) = \sqrt{2}\sin(\omega_0 n + \phi)$$

Where the phase ϕ is uniformly distributed between 0 and 2π.

```
clear variables;
```

[✓] 1) Determine the autocorrelation function for *x*(*n*)

The autocorrelation function of a real **sine signal** $z(n) = A\sin(\omega n + \phi)$ where A and ω are real constants and $\phi \sim U(0, 2\pi)$ is:

$$r_{zz}(\ell) = -\frac{A^2}{2}\cos(\omega\ell)$$

As $A = \sqrt{2}$, we have:

$$r_{xx}(\ell) = -\frac{(\sqrt{2})^2}{2}\cos(\omega_0\ell)$$

$$r_{xx}(\ell) = -\cos(\omega_0\ell)$$

[?] 2) Determine the 2nd order forward linear prediction filter

Write down the normal equation for the forward linear prediction filter and determine the filter coefficients for a 2nd order filter. For mathematical convinience we set $\omega_0 = \frac{\pi}{3}$.

The forward linear predictor can predict "future" sample $x(n)$ given the previous p samples $x(n-1), x(n-2), \cdots, x(n-p)$. Such an predictor can be modelled as a FIR filter:

$$\hat{x}[n] = \sum_{k=1}^{p} h_k x[n-k] = \boldsymbol{h}^{\mathrm{T}}\boldsymbol{x}[n-1], \qquad (14.122)$$

where $\boldsymbol{h} \triangleq [h_1\ h_2\ \ldots\ h_p]^{\mathrm{T}}$ and $\boldsymbol{x}[n-1] \triangleq [x[n-1]\ x[n-2]\ \ldots\ x[n-p]]^{\mathrm{T}}$

The normal equation for the optimum p'th order linear predictor is given by Eq. 14.125:

$$\boldsymbol{Rh} = \boldsymbol{r}, \qquad (14.125)$$

The normal equation for the second order filter is:

$$\begin{bmatrix} r_{xx}(0) & r_{xx}(1) \\ r_{xx}(1) & r_{xx}(0) \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} = \begin{bmatrix} r_{xx}(1) \\ r_{xx}(2) \end{bmatrix}$$

```
M = 2;
w0 = pi/3;
ell = 1:M+1;

r_xx = -cos(w0*ell);
R = toeplitz(r_xx(1:M));
r = r_xx(2:M+1)';
h_opt = R\r
```

```
h_opt = 2×1
10^15 ×
   -2.2365
```

```
    -2.2365
```

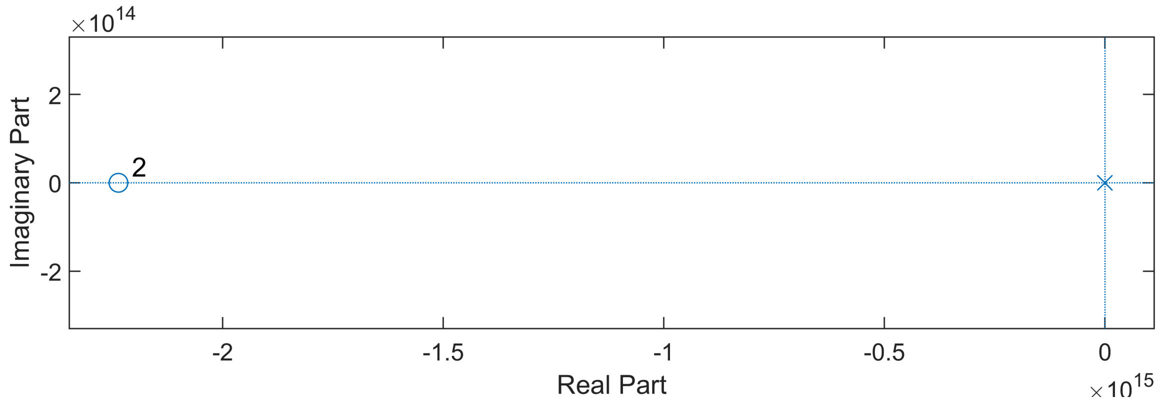
[?] The coefficients seem too large.

[»] 3) Find the system function for the filter and locate the zeros

Find the system function $H(z)$ for the filter and locate the zeros.

$$H(z) = h_1 z^{-1} + h_2 z^{-2}$$

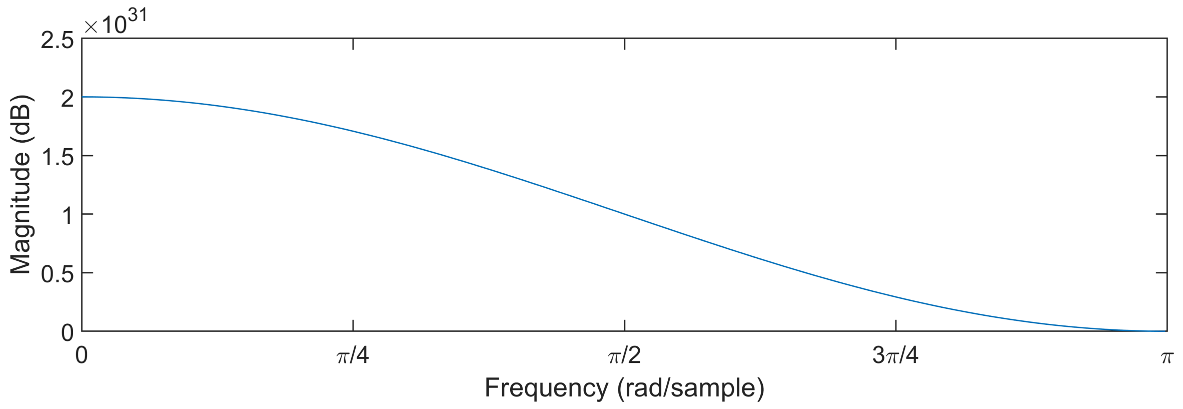
```
zplane(h_opt, 1)
```



[»] 4) Determine the frequency response, plot it and comment on the result

Determine the frequency response $H(\omega)$. Plot it and comment on the result.

```
[H, w] = freqz(h_opt, 1);
plot(w, H.*conj(H));
set(gca,'XTick', 0:pi/4:pi)
set(gca,'XTickLabel', {'0','\pi/4','\pi/2','3\pi/4','\pi'})
xlabel('Frequency (rad/sample)')
ylabel('Magnitude (dB)')
xlim([0, pi]);
```



[»] 5) Calculate the prediction error

The minimum square error for an optimum linear predictor is given by Eq. 14.127:

$$J_0 = r[0] - \boldsymbol{h}^{\mathrm{T}}\boldsymbol{r} = r[0] - \boldsymbol{r}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{r}. \qquad (14.127)$$

```
r_xx(1)
```

```
ans = -0.5000
```

```
mse = r_xx(1) - h_opt'*r
```

```
mse = 3.3547e+15
```

[?] ADSI Problem 6.8: Autocorrelation function and linear prediction

Assume that for a given sequence of data $\{x(n)\}$ *the autocorrelation function has been calculated and used to solve the normal equations so that the optimum p'th order linear predictor was found. Now, an amplifier is placed in the signal chain so that the signal is* $\{c \cdot x(n)\}$*. How does the autocorrelation function and the linear predictor change?*

In this problem, the autocorrelation function $r_{xx}(\ell)$ is already computed for signal $x(n)$.

We want to find the autocorrelation function for the amplified signal $y(n) = c \cdot x(n)$:

$$r_{yy}(\ell) = E[y(n)y(n-\ell)]$$

$$r_{yy}(\ell) = E[c\,x(n)\,c\,x(n-\ell)]$$

$$r_{yy}(\ell) = c^2 E[x(n)x(n-\ell)]$$

$$r_{yy}(\ell) = c^2\, r_{xx}(\ell)$$

So, the autocorrelation sequence of the amplified signal is multiplied by c^2

The normal equation for the optimum linear predictor is given by Eq. 14.125:

$$\boldsymbol{Rh} = \boldsymbol{r}, \qquad (14.125)$$

Even when the autocorrelation sequence is multiplied by some factor, the impulse response of the linear predictor $\boldsymbol{h}$ will remain the same. Therefore, the linear predictor will still work the same way.

[?] Are there other explanations?