Homework 10

Table of Contents

[✔] ADSI Problem 6.2: Autocorrelation expression for an AR(1) process (proof)

Consider an $\mathrm{AR}(1)$ process given by $\quad y(n) = -ay(n-1) + x(n)$

with $-1 < a < 1$ and $x(n)\ WN(0, \sigma_x^2)$.

1. Show that the autocorrelation of the $\mathrm{AR}(1)$ process is given by

$$r_{yy}(l) = \frac{\sigma_x^2}{1-a^2}(-a)^{|l|}$$

Hint: Use equation (13.138) and (13.140).

The autocorrelation of an AR(p) process is given by Eq. 13.138:

$$r_{yy}[\ell] = -\sum_{k=1}^{p} a_k r_{yy}[\ell - k] + \sigma_x^2 b_0 h[-\ell]. \text{ all } \ell \tag{13.138}$$

where $h[n]$ is the impulse response of an all-pole system.

Equation 13.138 is an general expression for all ℓ. However, since $h[\ell] = 0$ for negative values of ℓ then we know that $h[-\ell] = 0$. Therefore, Eq. 13.138 can be reduced to:

$$r_{yy}[\ell] = -\sum_{k=1}^{p} a_k r_{yy}[\ell - k], \ \ell > 0 \tag{13.140}$$

For an AR(1) process, Eq. 13.138 simplifies to:

$$r_{yy}[\ell] = -a_1 r_{yy}[\ell - 1] + \sigma_x^2 b_0 h[-\ell] \text{ for all } \ell$$

For an AR(1) process, Eq. 13.140 simplifies to:

$$r_{yy}[\ell] = -a_1 r_{yy}[\ell - 1] \text{ for } \ell > 0$$

Setting $\ell = 0$ in the first equation, we get:

$$r_{yy}[0] = -a_1 r_{yy}[-1] + \sigma_x^2 b_0 h[0]$$

The book says that $h[0] = b_0 = 1$ so we are left with:

$$r_{yy}[0] = -a_1 r_{yy}[-1] + \sigma_x^2$$

We can use the symmetry property of autocorrelation function i.e., $r_{yy}[-\ell] = r_{yy}[-\ell]$:

$$r_{yy}[0] = -a_1 r_{yy}[1] + \sigma_x^2$$

To find an expression for $r_{yy}[1]$, we set $\ell = 1$ in the second equation:

$$r_{yy}[1] = -a_1 r_{yy}[0]$$

We insert the second equation into the first equation:

$$r_{yy}[0] = -a_1(-a_1 r_{yy}[0]) + \sigma_x^2$$

$$r_{yy}[0] = a_1^2 r_{yy}[0] + \sigma_x^2$$

$$\sigma_x^2 = r_{yy}[0] - a_1^2 r_{yy}[0]$$

$$\sigma_x^2 = r_{yy}[0]\left(1 - a_1^2\right)$$

$$r_{yy}[0] = \frac{\sigma_x^2}{1 - a_1^2}$$

Now, we need to find an expression for $\ell > 0$. We can do this by using the second equation.

First, we set $\ell = 1$ in the second equation:

$$r_{yy}[1] = -a_1 r_{yy}[0] \Leftrightarrow -a_1 \frac{\sigma_x^2}{1 - a_1^2}$$

$$r_{yy}[2] = -a_1 r_{yy}[1] \Leftrightarrow -a_1 \left(-a_1 \frac{\sigma_x^2}{1 - a_1^2}\right) \Leftrightarrow (-a_1)^2 \frac{\sigma_x^2}{1 - a_1^2}$$

$$r_{yy}[3] = -a_1 r_{yy}[2] \Leftrightarrow -a_1 \left((-a_1)^2 \frac{\sigma_x^2}{1 - a_1^2}\right) \Leftrightarrow (-a_1)^3 \frac{\sigma_x^2}{1 - a_1^2}$$

This means that in general, we have:

$$r_{yy}[\ell] = (-a_1)^\ell \frac{\sigma_x^2}{1 - a_1^2}$$

If we apply the symmetric property of the autocorrelation, we get:

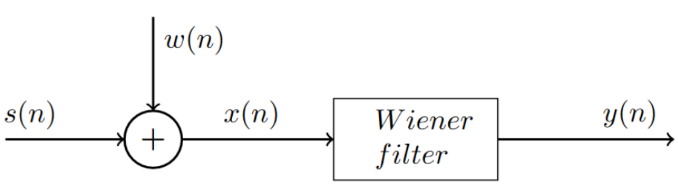
$$r_{yy}[\ell] = (-a_1)^{|\ell|} \frac{\sigma_x^2}{1 - a_1^2}$$

ADSI Problem 6.3: Wiener FIR Filtering, minimum square error

Consider a signal $x(n) = s(n) + w(n)$ where $s(n)$ is an AR(1) process that satisfies the difference equation

$$s(n) = 0.8s(n-1) + v(n)$$

where $\{v(n)\}$ is a white noise sequence with variance $\sigma_v^2 = 0.49$ and $\{w(n)\}$ is a white noise sequence with variance $\sigma_w^2 = 1$. The processes $\{v(n)\}$ and $\{w(n)\}$ are uncorrelated.




```
clear variables;
```

[✔] 1) Determine the autocorrelation sequence for a signal with two noise processes

Determine the autocorrelation sequences $\{r_s(l)\}$ and $\{r_x(l)\}$.

In Problem 6.2, we found that the autocorrelation function of an AR(1) is:

$$r_{\mathrm{yy}}^{\mathrm{AR}(1)}(\ell) = (-a_1)^{|\ell|}\frac{\sigma_x^2}{1-a_1^2}$$

For the AR(1) process, we are given $a_1 = -0.8$ and $\sigma_v^2 = 0.49$. So the autocorrelation sequence for $s(n)$ is:

$$r_{\mathrm{ss}}(\ell) = 0.8^{|\ell|}\frac{0.49}{1-(-0.8)^2} = 0.8^{|\ell|}\frac{0.49}{0.36}$$

Next, we need to find the autucorrelation sequence for $w(n)$. The autocorrelation of white Gaussian noise is $\sigma_w^2\delta(\ell)$ so since white noise has unit variance i.e. $\sigma_w^2 = 1$ we have:

$$r_{\mathrm{ww}}(\ell) = \delta(\ell)$$

Since the two noise processes are uncorrelated, the ACRS of $\{x(n)\}$ process is:

$$r_{\mathrm{xx}}(\ell) = 0.8^{|\ell|}\frac{0.49}{0.36} + \delta(\ell)$$

[�] 2) Design a Wiener filter of length M=2 to estimate an AR(1) process

Design a Wiener filter of length $M = 2$ to estimate $\{s(n)\}$.

The optimum Wiener filter to estimate a random process is given by Eq. 14.109:

$$\boldsymbol{h}_{\mathrm{o}} = \boldsymbol{R}_x^{-1}\boldsymbol{g}, \qquad (14.109)$$

where $\boldsymbol{R}_x$ is the correlation matrix of a random vector $\boldsymbol{x}$ and $\boldsymbol{g}$ is the cross-correlation vector between $\boldsymbol{x}$ and y (the signal that we want to recover which in this problem is $s[n]$).

Basically, to design a pth order Wiener filter, we have to solve following equation with respect to $\boldsymbol{h}$:

$$\begin{bmatrix} r_x[0] & r_x[1] & \dots & r_x[p-1] \\ r_x[1] & r_x[0] & \dots & r_x[p-2] \\ \vdots & \vdots & \ddots & \vdots \\ r_x[p-1] & r_x[p-2] & \dots & r_x[0] \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_p \end{bmatrix} = \begin{bmatrix} r_{yx}[0] \\ r_{yx}[1] \\ \vdots \\ r_{yx}[p-1] \end{bmatrix}, \qquad (14.113)$$

Computing the autocorrelation matrix is straightforward since we have computed the autocorrelation $r_{xx}(\ell)$ in part 1).

```
M = 2;
ell = 0:M-1;

r_ss = 0.8.^abs(ell) * (0.49/0.36);
r_ww = (ell == 0); % Simulate the delta function
r_xx = r_ss + r_ww;
R_xx = toeplitz(r_xx)
```

```
R_xx = 2×2
    2.3611    1.0889
    1.0889    2.3611
```

We have to come up with an expression for the cross-correlation $r_{sx}(\ell)$:

$$r_{sx}(\ell) = E[s(n)x(n-\ell)]$$

$$r_{sx}(\ell) = E[s(n)(s(n-\ell)+w(n-\ell))]$$

$$r_{sx}(\ell) = E[s(n)s(n-\ell)+s(n)w(n-\ell)]$$

$$r_{sx}(\ell) = E[s(n)s(n-\ell)]+E[s(n)w(n-\ell)]$$

Since the processes $s(n)$ and $w(n)$ are uncorrelated:

$$r_{sx}(\ell) = E[s(n)s(n-\ell)]+E[s(n)]\cdot E[w(n-\ell)]$$

� ***Is it okay to make this assumption?***

We assume that $w(n)$ is a zero-mean WGN, we know that $E[w(n)] = 0$, so we are left with

$$r_{sx}(\ell) = r_{ss}(\ell) + 0$$

From 1), we know the that:

$$r_{ss}(\ell) = 0.8^{|\ell|}\frac{0.49}{0.36}$$

```
g = r_ss';

h_opt = R_xx\g % Same as `inv(R_xx)*g` but better
```

```
h_opt = 2×1
    0.4621
    0.2481
```

[✔] 3) Determine the minimum mean square error for M=2

The minimum square error for an optimum pth Wiener (FIR) filter is given by Eq. 14.115:

$$J_o = r_y[0] - \boldsymbol{h}_o^T \boldsymbol{g} = r_y[0] - \sum_{k=0}^{p-1} h_o[k] r_{yx}[k]. \qquad (14.115)$$

```
mse = r_ss(1) - h_opt'*g
```

```
mse = 0.4621
```

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 square error to derive coefficients based on the ACRS

ADSI Problem 6.5: Levinson-Durbin by Hand

The aim of this problem is to get a finger-tip feeling of the flow of the Levinson-Durbin recursion. Assume that the following autocorrelation function values have been determined from an unknown random process $\{x(n)\}$

l	$r_{xx}(l)$
0	5
1	4
2	3
3	2
4	1

Work through the Levinson-Durbin recursion by hand and find the optimum linear predictors for m=1, 2 and 3 as well as the corresponding minimum mean square errors J_m's and reflection coefficients k_m's.

1) Work though Levinson-Durbin by hand to find the optimum linear predictors for m=1,2,3

2) Find the corresponding minimum mean square errors

3) Find the corresponding reflection coefficients

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

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

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

1) Plot the autocorrelation function.

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)?

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

ADSI Problem 6.7: Linear prediction

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.

1) Determine the autocorrelation function for *x*(*n*)

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 $\omega_0 = \frac{\pi}{3}$.

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.

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.

5) Calculate the prediction error

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?*

Functions