Wiener filters

Table of Contents

Filters that minimize the output mean square error

Problem: using a linear estimator, we want to guess the value of random variable y from a set of observations of a related set of p random variables $x_1, x_2, \cdots, x_p$:

$$\hat{y} = \sum_{k=1}^{p} h_k x_k = \boldsymbol{h}^{\mathrm{T}} \boldsymbol{x}. \qquad (14.100)$$

This is basically a multivariate linear regression problem where $\boldsymbol{h}$ is the coefficients of the model. In the multivariate regression problem, we hypothesise that the response random variable Y can be computed as a linear combination of the cofficients of the model and the predictor variables $X_1, X_2, \cdots, X_p$

$$\widehat{y}_h(\boldsymbol{x}) = \boldsymbol{h}^T \boldsymbol{x}$$

The mean squared error loss function is:

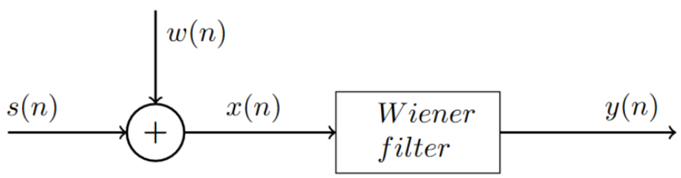
$$J_h = \frac{1}{2N} \sum_{i=1}^{N} \left(\widehat{y}_h(\boldsymbol{x}_i) - y_i\right)^2$$

We can minimise the loss function by taking the derivative, setting it zero and solving the equation.

Wiener filter

A special case of the multivariate linear regression model is the Wiener filter, which is used to estimate the original signal $s[n]$ given p observations of a noise distorted signal $x[n] = s[n] + w[n]$.

We assume that the noise process $w[n]$ is uncorrelated with the desired process that generated the original signal $s[n]$. We also assume that $x[n]$ and $w[n]$ are wide-sense stationary.



An p'order Wiener filter for estimating the original signal $s(n)$ is given by Eq. 14.112:

$$\hat{y}[n] = \sum_{k=1}^{p} h_k x[n+1-k] \qquad (14.112)$$

The normal equation for the Wiener filter is given by Eq. 14.113

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

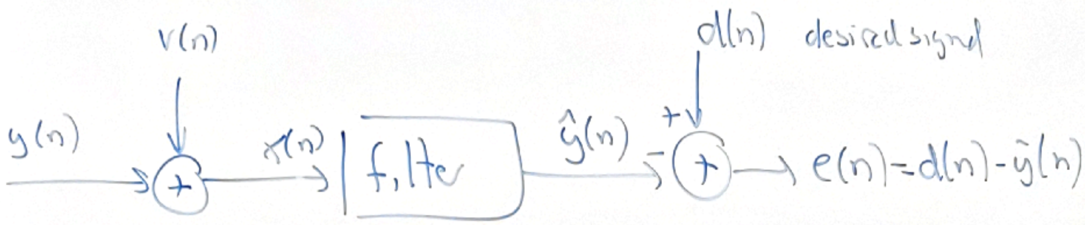
where $h_k = h_o[k-1]$

The minimum square error for a pth Wiener filter is given by Eq. 14.115:

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

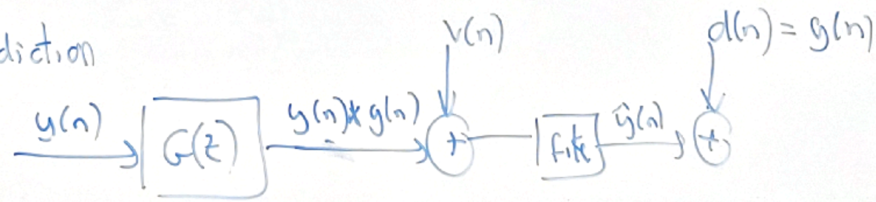
The optimum Wiener filter "passes" the input at bands with high SNR and "blocks" the input at bands with low SNR (see Eq. 14.120)

Applications of Wiener filtering



Desired signal:

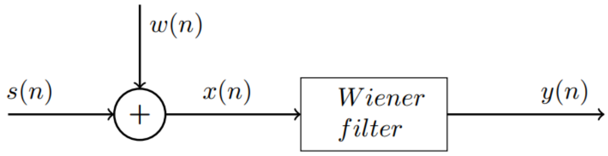
1. $x(n \pm m)$ forward/backward lin. pred
2. $y(n)$ filtering
3. $y(n \pm m)$ signal prediction
4. deconvolution



Exam 2015 Problem 3: Recover signal by a Wiener filter

```
clear variables;
```

A signal $s(n)$ is created as an AR(2) process. It is corrupted by uncorrelated, additive noise $w(n)$ as shown in the figure.



It is desired to recover the signal by a Wiener filter. The AR(2) process has two complex poles at $re^{\pm j\theta}$. For this kind of process the autocorrelation of the signal can be calculated analytically and is given by

$$r_s(l) = \frac{r^l\left(\sin((l+1)\theta) - r^2\sin((l-1)\theta)\right)}{((1-r^2)\sin\theta)(1-2r^2\cos 2\theta + r^4)}, \quad l \geq 0$$

For simplicity the first few values of the autocorrelation function for this specific process is given by

$\lvert l\rvert$	$r_s(l)$
0	7.89
1	5.55
2	0.67
3	-3.64
4	-5.18
5	-3.64

```
r_ss = [7.89, 5.55, 0.67, -3.64, -5.18, -3.64];
```

Less information is known about the noise. The autocorrelation for the first few lags are given by:

$\lvert l\rvert$	$r_w(l)$
0	5
1	3
2	1

For all other lags, the autocorrelation, $r_w(l)$ is assumed to be zero.

```
r_ww = [5, 3, 1, 0, 0, 0];
```

1) Design a 4 tap Wiener filter to recover *s*(*n*).

The p-order Wiener filter for estimating the signal $s(n)$ is given by:

$$\hat{y}[n] = \sum_{k=1}^{p} h_k x[n+1-k] \qquad (14.112)$$

The optimum Wiener filter coefficients is given by the Wiener-Hopf equation:

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

where $\boldsymbol{R}_x$ is the autocorrelation matrix of the corrupted signal $s(n)$ and $\boldsymbol{g}$ is the cross-correlation between the desired signal $s(n)$ and the corrupted signal $x(n)$.

Designing a Wiener filter to recover a corrupted signal involves 3 steps:

1. Compute the autocorrelation sequence $r_x(\ell)$ and matrix R_x
2. Compute the cross-correlation $r_{sx}(\ell)$
3. Solve the Wiener-Hopf equation to find the optimum Wiener filter coefficients

Step 1: Compute the autocorrelation $r_x(\ell)$:

Since the signal $s(n)$ and the noise $w(n)$ are uncorrelated the autocorrelation function of $s(n)$ is just the sum of the individual autocorrelation functions:

$$rx(\ell) = E[x(n)x(n-\ell)]$$

$$= E[(s(n)+w(n))(s(n-\ell)+w(n-\ell))]$$

$$= E[s(n)s(n-\ell)+s(n)w(n-\ell)+w(n)s(n-\ell)+w(n)w(n-\ell)]$$

$$= r_s(\ell)+r_{sw}(\ell)+r_{sw}(\ell)+r_w(\ell)$$

$= r_s(\ell)+r_w(\ell)$ (assuming $s(n)$ and $w(n)$ are uncorrelated)

```
p = 4;
r_xx = r_ss + r_ww;
R_xx = toeplitz(r_xx(1:p))
```

```
R_xx = 4×4
   12.8900    8.5500    1.6700   -3.6400
    8.5500   12.8900    8.5500    1.6700
    1.6700    8.5500   12.8900    8.5500
   -3.6400    1.6700    8.5500   12.8900
```

Step 2: Compute the cross-correlation $r_{sx}(\ell)$:

Since the desired signal is $s(n)$, the cross-correlation between $s(n)$ and $s(n)$ simplifies to the autocorrelation of the signal $r_s(\ell)$:

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

$$= E[s(n)(s(n-\ell) + w(n-\ell))]$$

$$= E[s(n)s(n-\ell) + s(n)w(n-\ell)]$$

$$= r_s(\ell) + r_{sw}(\ell)$$

$= r_s(\ell)$ (assuming $s(n)$ and $w(n)$ are uncorrelated)

```
g = r_ss(1:p)'
```

```
g = 4×1
    7.8900
    5.5500
    0.6700
   -3.6400
```

Step 3: Compute the optimum filter coefficients:

```
h_opt = R_xx\g
```

```
h_opt = 4×1
    0.4841
    0.1026
    0.0477
   -0.1906
```

2) Calculate the minimum mean square error (MSE)

The minimum value of the mean square error $E[e^2(n)]E\left[(y(n) - \widehat{y}(n))^2\right]$ is given by

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

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

```
mse = 2.7757
```

3) Can the MSE be lowered using a longer Wiener filter?

Explain whether the minimum mean square error can be lowered by using a longer Wiener filter when we only have sparse knowledge of the noise.

The minimum mean square error can be reduced further by increasing the length of the Wiener filter. As the length of the fiter is increased the frequency resolution increases and the spectral shape of the filter becomes better suited for passing the signal and rejecting the noise.

This is also evident if the minimum mean square error is calculated. For a 5 tap filter, the minimum mean square error decreases to 2.6617:

```
p = 5;
r_xx = r_ss + r_ww;
R_xx = toeplitz(r_xx(1:p));
g = r_ss(1:p)';
h_opt = R_xx\g;
mse = r_ss(1) - h_opt'*g
```

```
mse = 2.6617
```

4) Calculate SNR before and after the Wiener filter

The signal to noise ratio of the input signal is given by

$$\text{SNR}_i = \frac{\text{power of signal}}{\text{power of noise}} = \frac{r_s(0)}{r_w(0)}$$

```
SNR_i = r_ss(1) / r_ww(1)
```

```
SNR_i = 1.5780
```

The output SNR of the Wiener filter is given by:

$$\text{SNR}_o = \frac{h_{\text{opt}}^T R_s h_{\text{opt}}}{h_{\text{opt}}^T R_w h_{\text{opt}}}$$

```
R_ss = toeplitz(r_ss(1:p));
R_ww = toeplitz(r_ww(1:p));
SNR_o = (h_opt'*R_ss*h_opt) / (h_opt'*R_ww*h_opt)
```

```
SNR_o = 2.2673
```

The filter increases the signal to noise ratio but only slightly. This is expected and is an excellent way to check your results.

5) What happens when the noise amplitude is twice as large?

Consider a second, identical Wiener filtering problem, except that the noise amplitude is twice as big. Discuss whether this second system will have the same minimum mean square error as the first system.

The expected outcome of an increase of the noise amplitude is an increase of the minimum mean square error as there is more noise to remove.

If the noise $w(n)$ doubles in amplitude, then the values of the autocorrelation function must quadruple:

$$r_{2w}(\ell) = E[2w(n)2w(n-\ell)] = 4E[w(n)w(n-\ell)] = 4r_w(\ell)$$

```
r_xx = r_ss + r_ww;
R_xx = toeplitz(r_xx(1:p));
g = r_ss(1:p)';
h_opt = R_xx\g;
mse = r_ss(1) - h_opt'*g
```

```
mse = 2.6617
```

```
r_xx = r_ss + r_ww.*4;
R_xx = toeplitz(r_xx(1:p));
g = r_ss(1:p)';
h_opt = R_xx\g;
mse = r_ss(1) - h_opt'*g
```

```
mse = 5.1096
```

We see that the minimum MSE increases when the noise amplitude is increased.

Another example using following autocorrelation sequences:

l	$r_s(l)$	$r_w(l)$	$r_x(l)$
0	1	1	2
1	-0.4	0.8187	0.4187
2	0.2	0.7536	0.9536

```
p = 3;
r_ss = [1, -0.4, 0.2];
r_ww = [1, 0.8187, 0.7536];
r_xx = r_ss + r_ww;
R_xx = toeplitz(r_xx(1:p));
g = r_ss(1:p)';
h_opt = R_xx\g;
mse = r_ss(1) - h_opt'*g
```

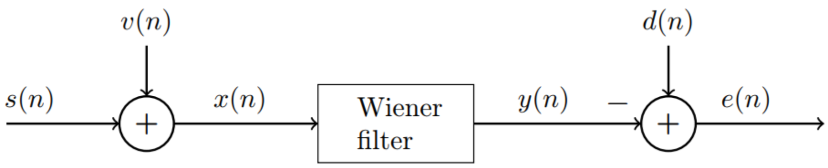
```
mse = 0.2777
```

```
r_xx = r_ss + r_ww.*4;
R_xx = toeplitz(r_xx(1:p));
g = r_ss(1:p)';
h_opt = R_xx\g;
mse = r_ss(1) - h_opt'*g
```

```
mse = 0.5081
```

Exam 2017 Problem 4: Recover signal using a Wiener filter

Consider the Wiener filtering problem shown below. A signal $s(n)$ is corrupted by additive, uncorrelated noise $v(n)$ giving the signal $x(n) = s(n) + v(n)$. It is desired to use a 2-tap Wiener filter to recover the signal i.e. $d(n) = s(n)$.



The noise $v(n)$ is generated in an MA(1) process as

$$v(n) = w(n) + w(n-1)$$

where $w(n)$ is a zero-mean gaussian white noise sequence with $\sigma_w^2 = 1$. The autocorrelation function of the signal is

$$r_s(l) = 4 \cdot 0.25^{|l|}.$$

```
clear variables;

M = 2;
ell = 0:M-1;
delta = @(l) ell == l;

r_ss = 4*(0.25).^(abs(ell));
```

1) Compute the autocorrelation function of the noise

1. Show that the autocorrelation function of the noise $v(n)$ is given by

$$r_v(l) = \delta(l-1) + 2\delta(l) + \delta(l+1).$$

In ADSI Problem 4.9, we found that the autocorrelation for an MA(1) process where the input signal is a white noise with unit variance can be described as:

$$r_v(\ell) = \left(b_0^2 + b_1^2\right)\delta(\ell) + b_0 b_1\left(\delta(\ell-1) + \delta(\ell+1)\right)$$

In this problem $b_0 = b_1 = 1$:

$$r_v(\ell) = \delta(\ell-1) + 2\delta(\ell) + \delta(\ell+1)$$

```
r_vv = delta(-1) + 2*delta(0) + delta(1)
```

2) Compute the optimum filter coefficients

2. Calculate the crosscorrelation vector $\mathbf{g}$, the autocorrelation matrix $\mathbf{R}_x$ and the optimum filter coefficients.

The p-order Wiener filter for estimating the signal $s(n)$ is given by:

$$\hat{y}[n] = \sum_{k=1}^{p} h_k x[n+1-k] \qquad (14.112)$$

The optimum Wiener filter coefficients is given by the Wiener-Hopf equation:

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

where $\boldsymbol{R}_x$ is the autocorrelation matrix of the corrupted signal $s(n)$ and $\boldsymbol{g}$ is the cross-correlation between the desired signal $s(n)$ and the corrupted signal $x(n)$.

Designing a Wiener filter to recover a corrupted signal involves 3 steps:

1. Compute the autocorrelation sequence $r_x(\ell)$ and matrix R_x
2. Compute the cross-correlation $r_{\mathrm{sx}}(\ell)$
3. Solve the Wiener-Hopf equation to find the optimum Wiener filter coefficients

Step 1: Compute the autocorrelation $r_x(\ell)$:

Since the signal $s(n)$ and the noise $w(n)$ are uncorrelated the autocorrelation function of $s(n)$ is just the sum of the individual autocorrelation functions:

$$r_x(\ell) = E[x(n)x(n-\ell)]$$

$$= E[(s(n)+v(n))(s(n-\ell)+v(n-\ell))]$$

$$= E[s(n)s(n-\ell)+s(n)v(n-\ell)+v(n)s(n-\ell)+v(n)v(n-\ell)]$$

$$= r_s(\ell)+r_{\mathrm{sv}}(\ell)+r_{\mathrm{sv}}(\ell)+r_v(\ell)$$

Since $s(n)$ and $v(n)$ are uncorrelated $r_{\mathrm{sv}}(\ell)=0$. We computed $r_v(\ell)=\delta(\ell-1)+2\delta(\ell)+\delta(\ell+1)$:

$$r_x(\ell) = r_s(\ell)+r_v(\ell)$$

```
r_xx = r_ss + r_vv;
R_xx = toeplitz(r_xx)
```

Step 2: Compute the cross-correlation $r_{\mathrm{sx}}(\ell)$:

Since the desired signal is $s(n)$, the cross-correlation between $s(n)$ and $s(n)$ simplifies to the autocorrelation of the signal $r_s(\ell)$.

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

$$= E[s(n)(s(n-\ell)+v(n-\ell))]$$

$$= E[s(n)s(n-\ell)]+E[s(n)v(n-\ell)]$$

$= r_s(\ell) + r_{sv}(\ell)$

$= r_s(\ell) + 0$ (since $s(n)$ and $v(n)$ are uncorrelated $r_{sv}(\ell) = 0$

```
g = r_ss'
```

Step 3: Compute the optimum filter coefficients:

```
h_opt = R_xx\g
```

3) Calculate SNR before and after the Wiener filter

The signal to noise ratio of the input signal is given by

$$\text{SNR}_i = \frac{(\text{value of signal at } n = n_0)^2}{\text{power of noise}} = \frac{s^2(n = n_0)}{r_v(0)} = \frac{r_s(0)}{r_v(0)}$$

We know that:

$$r_s(\ell) = 4 \cdot 0.25^{|\ell|} \quad \text{and} \quad r_v(\ell) = \delta(\ell - 1) + 2\delta(\ell) + \delta(\ell + 1)$$

This means:

$$r_s(0) = 4 \quad \text{and} \quad r_v(0) = 2$$

Thus:

$$\text{SNR}_i = \frac{r_s(0)}{r_v(0)} = \frac{4}{2} = 2$$

The output SNR of the Wiener filter is given by:

$$\text{SNR}_o = \frac{h_{\text{opt}}^T R_s h_{\text{opt}}}{h_{\text{opt}}^T R_v h_{\text{opt}}}$$

```
R_ss = toeplitz(r_ss);
R_vv = toeplitz(r_vv);
SNR_o = (h_opt'*R_ss*h_opt) / (h_opt'*R_vv*h_opt)
```

The filter increases the SNR but only slightly. This is expected and is an excellent way to check your results.

Exam 2018 Problem 4: Recover signal using a Wiener filter

A system given by $H(z) = 4 + z^{-1}$ is excited by unit variance white Gaussian noise $w(n)$ to give the signal $s(n)$.

```
clear variables;
```

1) Determine the autocorrelation function, $r_s(l)$

The output signal of the system is given as:

$$s(n) = 4w(n) + w(n-1)$$

The autocorrelation of this signal is:

$$r_s(\ell) = E[s(n)s(n-\ell)]$$

$$= E[(4x(n) + x(n-1))(4x(n-\ell) + x(n-\ell-1))]$$

$$= E[4x(n)4x(n-\ell)] + E[4x(n)x(n-\ell-1)] + E[x(n-1)4x(n-\ell)] + E[x(n-1)x(n-\ell-1)]$$

$$= 16E[x(n)x(n-\ell)] + 4E[x(n)x(n-\ell-1)] + 4E[x(n-1)x(n-\ell)] + E[x(n-1)x(n-\ell-1)]$$

```
syms n l w(n)
expand((4*w(n) + w(n-1)) * (4*w(n-l) + w(n-1-l)))
```

ans = $16\, w(n-l)\, w(n) + 4\, w(n-l-1)\, w(n) + 4\, w(n-l)\, w(n-1) + w(n-l-1)\, w(n-1)$

$$= 16r_w(\ell) + 4r_w(\ell-1) + 4r_w(\ell+1) + r_w(\ell)$$

Since $r_w(\ell-1) = r_w(\ell+1)$

$$r_s(\ell) = 17r_w(\ell) + 8r_w(\ell-1)$$

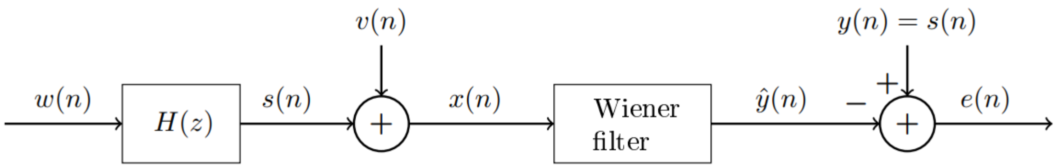
Since the autocorrelation of unit variance white noise is $r_{xx}(\ell) = \sigma_x^2\delta(\ell) = \delta(\ell)$:

$$r_s(\ell) = 17\delta(\ell) + 8\delta(\ell-1)$$

```
r_ss = [17, 8, 0];
```

2) Solve the Wiener-Hopf Equation

As shown in the block diagram below, the signal is corrupted by additive white Gaussian noise with $\sigma_v^2 = 3$ giving $x(n) = s(n) + v(n)$. It is desired to recover the signal with a 3-tap Wiener filter.



2. Determine the 3×3 autocorrelation matrix R_x and the 3×1 crosscorrelation vector $\mathbf{g} = E[\mathbf{x}(n)y(n)]$ and use these to solve the Wiener-Hopf equation.

We know that $x(n) = s(n) + v(n)$ and $y(n) = s(n) = 4w(n) + w(n-1)$

Assuming $w(n)$ and $v(n)$ are uncorrelated, the autocorrelation for $x(n)$:

$$r_x(\ell) = E[x(n)x(n-\ell)]$$

$$= E[(s(n) + v(n))(s(n-\ell) + v(n-\ell))]$$

$$= E[s(n)s(n-\ell) + s(n)v(n-\ell) + v(n)s(n-\ell) + v(n)v(n-\ell)]$$

$$= E[s(n)s(n-\ell)] + E[s(n)v(n-\ell)] + E[v(n)s(n-\ell)] + E[v(n)v(n-\ell)]$$

$$= r_s(\ell) + 2r_{sv}(\ell) + r_v(\ell)$$

$= r_s(\ell) + r_v(\ell)$ (because $w(n)$ and $v(n)$ are uncorrelated so $r_{sv}(\ell) = 0$)

$= 17\delta(\ell) + 8\delta(\ell - 1) + 3\delta(\ell)$ -- $r_v(\ell) = 3\delta(\ell)$ since $\sigma_v^2 = 3$

$$= 20\delta(\ell) + 8\delta(\ell - 1)$$

```
r_xx = [20, 8, 0];
R_xx = toeplitz(r_xx);
```

We know that $x(n) = s(n) + v(n)$ and $y(n) = s(n) = 4w(n) + w(n-1)$.

Compute the cross-correlation vector:

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

$$= E[(s(n) + v(n))(4w(n-\ell) + w(n-\ell-1))]$$

$$= 4E[s(n)w(n-\ell)] + E[s(n)w(n-\ell-1)] + 4E[v(n)w(n-\ell)] + E[v(n)w(n-\ell-1)]$$

$$= 4r_{sw}(\ell) + r_{sw}(\ell - 1) + 4r_{vw}(\ell) + r_{vw}(\ell - 1)$$

We assume that $w(n)$ and $v(n)$ are uncorrelated so $r_{vw}(\ell) = 0$:

$$= 4r_{sw}(\ell) + r_{sw}(\ell - 1) + 0 + 0$$

According to Eq. 13.100, the cross-correlation $r_{sw}(\ell) = \sigma_w^2 h(\ell) = h(\ell)$ since $\sigma_w^2 = 1$. We know that the impulse response $h(\ell) = [4, 1]$:

$$= 4h(\ell) + h(\ell - 1)$$

```
g = [4, 1, 0]';
```

The third order Wiener filter for estimating the signal $y(n)$ is given by:

$$\hat{y}(n) = h_1 x(n) + h_2 x(n-1) + h_3 x(n-2)$$

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

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

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

```
h_opt = R_xx\g
```

```
h_opt = 3×1
    0.2176
   -0.0441
    0.0176
```

The optimal Wiener filter is given by:

$$H(z) = 0.2471 - 0.1176z^{-1} + 0.0471z^{-2}$$

3) Discuss whether 3 taps is an optimum choice for this problem?

We can check whether the minimum mean square error can be lowered by using a longer Wiener filter.

Typically, increasing the length of the Wiener filter lowers the MSE which means that the filter becomes better

The minimum value of the mean square error $E[e^2(n)]E\left[(y(n) - \hat{y}(n))^2\right]$ is given by

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

Let us compute it for a 3-tap Wiener filter:

```
mse_tap3 = r_ss(1) - h_opt'*g
```

```
mse_tap3 = 16.1735
```

Let us compute it for a 4-tap Wiener filter:

```
R_xx4 = toeplitz([r_xx, 0]);
g4 = [g', 0]';
```

```
h_opt4 = R_xx4\g4;
mse_tap4 = r_ss(1) - h_opt4'*g4
```

```
mse_tap4 = 16.1723
```

Let us compute it for a 5-tap Wiener filter:

```
R_xx5 = toeplitz([r_xx, 0, 0]);
g5 = [g', 0, 0]';
h_opt5 = R_xx5\g5;
mse_tap5 = r_ss(1) - h_opt5'*g5
```

```
mse_tap5 = 16.1720
```

We observe that the minimum MSE can be lowered further by increasing the filter length. However, the decrease in minimum MSE is relatively small. If we want a faster filter, we would stick with 3-tap or even a 2-tap.

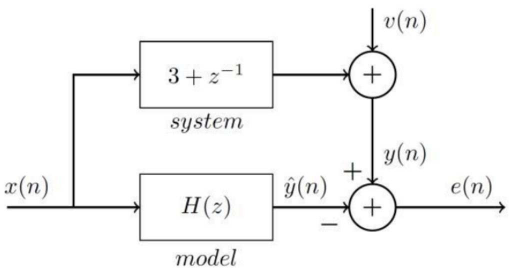
Let us compute it for a 2-tap Wiener filter:

```
R_xx2 = toeplitz(r_xx(1:2));
g2 = g(1:2);
h_opt2 = R_xx2\g2;
mse_tap2 = r_ss(1) - h_opt2'*g2
```

```
mse_tap2 = 16.1786
```

Exam 2018 Problem 1: Model a system using a Wiener filter

Consider the following setup, where we aim to model the system $3+z^{-1}$ using a Wiener filter. The input signal $x(n)$ is unit variance white noise. The output of the system is corrupted by additive noise $v(n)$, which is also white with $\sigma_v^2 = 0.2$. It can be assumed that $x(n)$ and $v(n)$ are uncorrelated.



$x(n) \sim \mathrm{WN}(0,1)$, $v(n) \sim \mathrm{WN}(0,0.2)$ and $x(n)$ and $v(n)$ are uncorrelated

```
clear variables;
```

1) Calcuate the autocorrelation matrix and cross-correction vector

1. Calculate the 2×2 autocorrelation matrix $\mathbf{R}_x$ and the 2×1 cross-correlation vector $\mathbf{g}$.

Since $x(n) \sim \mathrm{WN}(0, 1)$, its autocorrelation function is:

$$r_{\mathrm{xx}}(\ell) = \sigma_x^2 \delta(\ell) = \delta(\ell)$$

The autocorrelation matrix $\mathbf{R}_x$ is therefore

```
M = 2;
ell = 0:M-1;
r_xx = [1, 0];
R_xx = toeplitz(r_xx)
```

```
R_xx = 2×2
     1     0
     0     1
```

From the drawing, we know that the output of the system is:

$$y(n) = 3x(n) + x(n-1) + v(n)$$

Now, we can compute the cross-correlation vector g:

$$r_{\mathrm{xy}}(\ell) = E[x(n)y(n-\ell)]$$

$$= E[x(n)(3x(n-\ell) + x(n-\ell-1) + v(n-\ell))]$$

$$= E[3x(n)x(n-\ell) + x(n)x(n-\ell-1) + x(n)v(n-\ell)]$$

$$= 3E[x(n)x(n-\ell)] + E[x(n)x(n-\ell-1)] + E[x(n)v(n-\ell)]$$

$$= 3r_{\mathrm{xx}}(\ell) + r_{\mathrm{xx}}(\ell-1) + E[x(n)v(n-\ell)]$$

If we assume that $x(n)$ and $v(n)$ are uncorrelated then:

$$= 3r_{\mathrm{xx}}(\ell) + r_{\mathrm{xx}}(\ell-1)$$

Since $r_{\mathrm{xx}}(\ell) = \delta(\ell)$ then:

$$r_{\mathrm{xy}}(\ell) = 3\delta(\ell) + \delta(\ell-1)$$

Thus, the cross-correlation vector g is:

$$\boldsymbol{g} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$$

2) Calculate the expected energy

2. Calculate $E[y^2(n)]$.

From the drawing, we know that the output of the system is:

$$y(n) = 3x(n) + x(n-1) + v(n)$$

Let us first expand $y^2(n)$:

```
syms n x(n) v(n)
expand((3*x(n) + x(n-1) + v(n))^2)
```

ans = $x(n-1)^2 + 2\,x(n-1)\,v(n) + 6\,x(n-1)\,x(n) + v(n)^2 + 6\,v(n)\,x(n) + 9\,x(n)^2$

Now, compute the expectation of each term:

$$E[x^2(n-1)] = E[x(n)x(n)] = r_{xx}(0) = 1 \cdot \delta(0) = 1 \cdot 1 = 1$$

$E[2x(n-1)v(n)] = 2E[x(n-1)v(n)] = 0$ (because $x(n)$ and $v(n)$ are uncorrelated)

$$E[6x(n-1)x(n)] = 6r_{xx}(1) = 6\delta(1) = 6 \cdot 0 = 0$$

$$E[v^2(n)] = r_{vv}(0) = 0.2 \cdot \delta(0) = 0.2 \cdot 1 = 0.2$$

$E[6v(n)x(n)] = 6r_{xv}(0) = 6 \cdot 0 = 0$ (because $x(n)$ and $v(n)$ are uncorrelated)

$$E[9x(n)x(n)] = 9r_{xx}(0) = 9\delta(0) = 9 \cdot 1 = 9$$

The expected energy is:

$$E[y^2(n)] = 1 + 0.2 + 9 = 10.2$$

3) Solve for H(z) and comment on the influence of v(n)

3. Solve for $H(z)$ and comment on the result, in particular what is the influence of $v(n)$.

The p-order Wiener filter for estimating the signal $x(n)$ is given by:

$$\hat{y}[n] = \sum_{k=1}^{p} h_k x[n+1-k] \qquad (14.112)$$

The optimum Wiener filter coefficients is given by the Wiener-Hopf equation:

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

where $\boldsymbol{R}_x$ is the autocorrelation matrix of the input signal $x(n)$ and $\boldsymbol{g}$ is the cross-correlation between the input signal $x(n)$ and the desired signal $y(n)$.

```
g = [3, 1]';
R_xx = [[1, 0];
        [0, 1]];
h_opt = R_xx\g
```

```
h_opt = 2×1
     3
     1
```

The optimal Wiener filter is:

$$H(z) = 3 + z^{-1}$$

This is exactly the system that we are trying to model.

Exam 2018 Problem 4: Recover signal using a Wiener filter

A system given by $H(z) = 4 + z^{-1}$ is excited by unit variance white Gaussian noise $w(n)$ to give the signal $s(n)$.

```
clear variables;
```

[✔] 1) Determine the autocorrelation function, $r_s(l)$

The output signal of the system is given as:

$$s(n) = 4w(n) + w(n-1)$$

The autocorrelation of this signal is:

$$r_s(\ell) = E[s(n)s(n-\ell)]$$

$$= E[(4x(n) + x(n-1))(4x(n-\ell) + x(n-\ell-1))]$$

$$= E[4x(n)4x(n-\ell)] + E[4x(n)x(n-\ell-1)] + E[x(n-1)4x(n-\ell)] + E[x(n-1)x(n-\ell-1)]$$

$$= 16E[x(n)x(n-\ell)] + 4E[x(n)x(n-\ell-1)] + 4E[x(n-1)x(n-\ell)] + E[x(n-1)x(n-\ell-1)]$$

```
syms n l w(n)
expand((4*w(n) + w(n-1)) * (4*w(n-l) + w(n-1-l)))
```

ans = $16\, w(n-l)\, w(n) + 4\, w(n-l-1)\, w(n) + 4\, w(n-l)\, w(n-1) + w(n-l-1)\, w(n-1)$

$$= 16r_w(\ell) + 4r_w(\ell-1) + 4r_w(\ell+1) + r_w(\ell)$$

Since $r_w(\ell-1) = r_w(\ell+1)$

$$r_s(\ell) = 17r_w(\ell) + 8r_w(\ell-1)$$

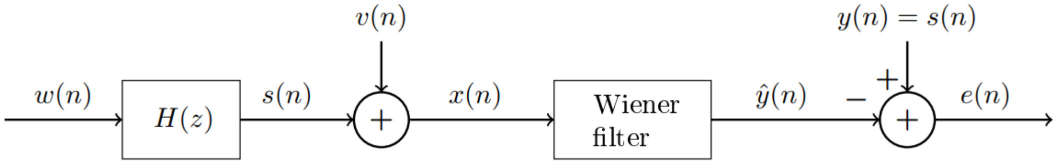
Since the autocorrelation of unit variance white noise is $r_{xx}(\ell) = \sigma_x^2\delta(\ell) = \delta(\ell)$:

$$r_s(\ell) = 17\delta(\ell) + 8\delta(\ell - 1)$$

```
r_ss = [17, 8, 0];
```

[✔] 2) Solve the Wiener-Hopf Equation

As shown in the block diagram below, the signal is corrupted by additive white Gaussian noise with $\sigma_v^2 = 3$ giving $x(n) = s(n) + v(n)$. It is desired to recover the signal with a 3-tap Wiener filter.



2. Determine the 3×3 autocorrelation matrix R_x and the 3×1 crosscorrelation vector $\mathbf{g} = E[\mathbf{x}(n)y(n)]$ and use these to solve the Wiener-Hopf equation.

We know that $x(n) = s(n) + v(n)$ and $y(n) = s(n) = 4w(n) + w(n - 1)$

Assuming $w(n)$ and $v(n)$ are uncorrelated, the autocorrelation for $x(n)$:

$$r_x(\ell) = E[x(n)x(n - \ell)]$$

$$= E[(s(n) + v(n))(s(n - \ell) + v(n - \ell))]$$

$$= E[s(n)s(n - \ell) + s(n)v(n - \ell) + v(n)s(n - \ell) + v(n)v(n - \ell)]$$

$$= E[s(n)s(n - \ell)] + E[s(n)v(n - \ell)] + E[v(n)s(n - \ell)] + E[v(n)v(n - \ell)]$$

$$= r_s(\ell) + 2r_{sv}(\ell) + r_v(\ell)$$

$= r_s(\ell) + r_v(\ell)$ (because $w(n)$ and $v(n)$ are uncorrelated so $r_{sv}(\ell) = 0$)

$= 17\delta(\ell) + 8\delta(\ell - 1) + 3\delta(\ell)$ -- $r_v(\ell) = 3\delta(\ell)$ since $\sigma_v^2 = 3$

$$= 20\delta(\ell) + 8\delta(\ell - 1)$$

We know that $x(n) = s(n) + v(n)$ and $y(n) = s(n) = 4w(n) + w(n - 1)$.

Compute the cross-correlation vector:

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

$$= E[(s(n) + v(n))(4w(n - \ell) + w(n - \ell - 1))]$$

$$= 4E[s(n)w(n-\ell)] + E[s(n)w(n-\ell-1)] + 4E[v(n)w(n-\ell)] + E[v(n)w(n-\ell-1)]$$

$$= 4r_{\text{sw}}(\ell) + r_{\text{sw}}(\ell-1) + 4r_{\text{vw}}(\ell) + r_{\text{vw}}(\ell-1)$$

We assume that $w(n)$ and $v(n)$ are uncorrelated so $r_{\text{vw}}(\ell) = 0$:

$$= 4r_{\text{sw}}(\ell) + r_{\text{sw}}(\ell-1) + 0 + 0$$

According to Eq. 13.100, the cross-correlation $r_{\text{sw}}(\ell) = \sigma_w^2 h(\ell) = h(\ell)$ since $\sigma_w^2 = 1$. We know that the impulse response $h(\ell) = [4, 1]$:

$$= 4h(\ell) + h(\ell-1)$$

The third order Wiener filter for estimating the signal $y(n)$ is given by:

$$\hat{y}(n) = h_1 x(n) + h_2 x(n-1) + h_3 x(n-2)$$

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

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

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

```
r_xx = [20, 8, 0];
R_xx = toeplitz(r_xx)
```

```
R_xx = 3×3
    20     8     0
     8    20     8
     0     8    20
```

```
g = [4, 1, 0]'
```

```
g = 3×1
     4
     1
     0
```

```
h_opt = R_xx\g
```

```
h_opt = 3×1
    0.2176
   -0.0441
    0.0176
```

The optimal Wiener filter is given by:

$$H(z) = 0.2471 - 0.1176z^{-1} + 0.0471z^{-2}$$

[✔] 3) Discuss whether 3 taps is an optimum choice for this problem?

We can check whether the minimum mean square error can be lowered by using a longer Wiener filter.

Typically, increasing the length of the Wiener filter lowers the MSE which means that the filter becomes better

The minimum value of the mean square error $E[e^2(n)]E\left[(y(n)-\hat{y}(n))^2\right]$ is given by

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

Let us compute it for a 3-tap Wiener filter:

```
mse_tap3 = r_ss(1) - h_opt'*g
```

```
mse_tap3 = 16.1735
```

Let us compute it for a 4-tap Wiener filter:

```
R_xx4 = toeplitz([r_xx, 0]);
g4 = [g', 0]';
h_opt4 = R_xx4\g4;
mse_tap4 = r_ss(1) - h_opt4'*g4
```

```
mse_tap4 = 16.1723
```

Let us compute it for a 5-tap Wiener filter:

```
R_xx5 = toeplitz([r_xx, 0, 0]);
g5 = [g', 0, 0]';
h_opt5 = R_xx5\g5;
mse_tap5 = r_ss(1) - h_opt5'*g5
```

```
mse_tap5 = 16.1720
```

We observe that the minimum MSE can be lowered further by increasing the filter length. However, the decrease in minimum MSE is relatively small. If we want a faster filter, we would stick with 3-tap or even a 2-tap.

Let us compute it for a 2-tap Wiener filter:

```
R_xx2 = toeplitz(r_xx(1:2));
g2 = g(1:2);
h_opt2 = R_xx2\g2;
mse_tap2 = r_ss(1) - h_opt2'*g2
```

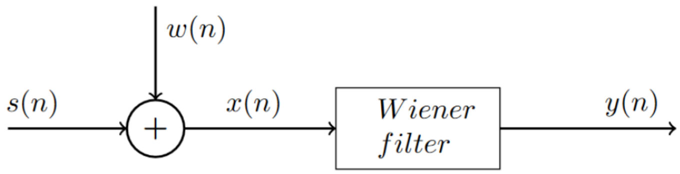
```
mse_tap2 = 16.1786
```

ADSI Problem 6.3: Recover AR(1) signal using Wiener filter

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:

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

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:

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

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:

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

As the two white noise processes are uncorrelated, the signal *s*(*n*) and *w*(*n*) are also uncorrelated. The autocorrelation of the noisy signal is therefore just the sum of the individual autocorrelations:

$$r_{\text{xx}}(\ell) = r_{\text{ss}}(\ell) + r_{\text{ww}}(\ell)$$

Therefore, the ACRS of $\{x(n)\}$ process is:

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

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 $\{s(n)\}$.

An M'order Wiener filter for estimating the original signal $s(n)$ is given by Eq. 14112:

$$\hat{y}[n] = \sum_{k=1}^{p} h_k x[n+1-k] \qquad (14.112)$$

Note: *In this problem, we must use $s(n)$ instead of $y(n)$ which is used in the book.*

The second order Wiener filter for estimating the original signal $s(n)$ is given by:

$$\hat{s}(n) = h_1 x(n) + h_2 x(n-1)$$

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

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

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 $\boldsymbol{h}$:

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

Eq. 14.113 is the normal equation for the Wiener filter.

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 $r_{sx}(\ell)$:

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

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

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

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

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

We already have an expression for $r_{ss}(\ell)$. Since the processes $s(n)$ and $w(n)$ are uncorrelated, we know that the cross-correlation between the two processes is $r_{sw}(\ell) = 0$ 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
```

The second order Wiener filter for estimating the original signal $s(n)$ is therefore:

$$\hat{s}(n) = 0.4621x(n) + 0.2481x(n-1)$$

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:

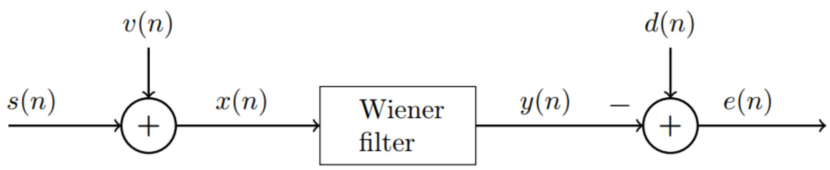
$$J_o = r_y[0] - \boldsymbol{h}_o^T\boldsymbol{g} = r_y[0] - \sum_{k=0}^{p-1} h_o[k]r_{yx}[k]. \qquad (14.115)$$

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

```
mse = 0.4621
```

Exam 2017 Problem 4: Recover signal using a Wiener filter

Consider the Wiener filtering problem shown below. A signal $s(n)$ is corrupted by additive, uncorrelated noise $v(n)$ giving the signal $x(n) = s(n) + v(n)$. It is desired to use a 2-tap Wiener filter to recover the signal i.e. $d(n) = s(n)$.



The noise $v(n)$ is generated in an MA(1) process as

$$v(n) = w(n) + w(n-1)$$

where $w(n)$ is a zero-mean gaussian white noise sequence with $\sigma_w^2 = 1$. The autocorrelation function of the signal is

$$r_s(l) = 4 \cdot 0.25^{|l|}.$$

```
clear variables;

M = 2;
ell = 0:M-1;
delta = @(l) ell == l;

r_ss = 4*(0.25).^(abs(ell));
```

1) Compute the autocorrelation function of the noise

1. Show that the autocorrelation function of the noise $v(n)$ is given by

$$r_v(l) = \delta(l-1) + 2\delta(l) + \delta(l+1).$$

In ADSI Problem 4.9, we found that the autocorrelation for an MA(1) process where the input signal is a white noise with unit variance can be described as:

$$r_v(\ell) = \left(b_0^2 + b_1^2\right)\delta(\ell) + b_0 b_1 \left(\delta(\ell-1) + \delta(\ell+1)\right)$$

In this problem $b_0 = b_1 = 1$:

$$r_v(\ell) = \delta(\ell-1) + 2\delta(\ell) + \delta(\ell+1)$$

```
r_vv = delta(-1) + 2*delta(0) + delta(1)
```

```
r_vv = 1×2
     2     1
```

2) Compute the optimum filter coefficients

2. Calculate the crosscorrelation vector $\mathbf{g}$, the autocorrelation matrix $\mathbf{R}_x$ and the optimum filter coefficients.

The p-order Wiener filter for estimating the signal $s(n)$ is given by:

$$\hat{y}[n] = \sum_{k=1}^{p} h_k x[n+1-k] \qquad (14.112)$$

The optimum Wiener filter coefficients is given by the Wiener-Hopf equation:

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

where $\boldsymbol{R}_x$ is the autocorrelation matrix of the corrupted signal $s(n)$ and $\boldsymbol{g}$ is the cross-correlation between the desired signal $s(n)$ and the corrupted signal $x(n)$.

Designing a Wiener filter to recover a corrupted signal involves 3 steps:

1. Compute the autocorrelation sequence $r_x(\ell)$ and matrix R_x
2. Compute the cross-correlation $r_{sx}(\ell)$
3. Solve the Wiener-Hopf equation to find the optimum Wiener filter coefficients

Step 1: Compute the autocorrelation $r_x(\ell)$:

Since the signal $s(n)$ and the noise $w(n)$ are uncorrelated the autocorrelation function of $s(n)$ is just the sum of the individual autocorrelation functions:

$$\begin{aligned} r_x(\ell) &= E[x(n)x(n-\ell)] \\ &= E[(s(n)+v(n))(s(n-\ell)+v(n-\ell))] \\ &= E[s(n)s(n-\ell)+s(n)v(n-\ell)+v(n)s(n-\ell)+v(n)v(n-\ell)] \\ &= r_s(\ell)+r_{sv}(\ell)+r_{sv}(\ell)+r_v(\ell) \end{aligned}$$

Since $s(n)$ and $v(n)$ are uncorrelated $r_{sv}(\ell) = 0$. We computed $r_v(\ell) = \delta(\ell-1)+2\delta(\ell)+\delta(\ell+1)$:

$$r_x(\ell) = r_s(\ell) + r_v(\ell)$$

```
r_xx = r_ss + r_vv;
R_xx = toeplitz(r_xx)
```

```
R_xx = 2×2
     6     2
     2     6
```

Step 2: Compute the cross-correlation $r_{sx}(\ell)$:

Since the desired signal is $s(n)$, the cross-correlation between $s(n)$ and $s(n)$ simplifies to the autocorrelation of the signal $r_s(\ell)$.

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

$$= E[s(n)(s(n-\ell) + v(n-\ell))]$$

$$= E[s(n)s(n-\ell)] + E[s(n)v(n-\ell)]$$

$$= r_s(\ell) + r_{\mathrm{sv}}(\ell)$$

$$= r_s(\ell) + 0 \quad \text{(since } s(n) \text{ and } v(n) \text{ are uncorrelated } r_{\mathrm{sv}}(\ell) = 0$$

```
g = r_ss'
```

```
g = 2×1
     4
     1
```

Step 3: Compute the optimum filter coefficients:

```
h_opt = R_xx\g
```

```
h_opt = 2×1
    0.6875
   -0.0625
```

3) Calculate SNR before and after the Wiener filter

The signal to noise ratio of the input signal is given by

$$\mathrm{SNR}_i = \frac{(\text{value of signal at } n = n_0)^2}{\text{power of noise}} = \frac{s^2(n = n_0)}{r_v(0)} = \frac{r_s(0)}{r_v(0)}$$

We know that:

$$r_s(\ell) = 4 \cdot 0.25^{|\ell|} \quad \text{and} \quad r_v(\ell) = \delta(\ell - 1) + 2\delta(\ell) + \delta(\ell + 1)$$

This means:

$$r_s(0) = 4 \quad \text{and} \quad r_v(0) = 2$$

Thus:

$$\mathrm{SNR}_i = \frac{r_s(0)}{r_v(0)} = \frac{4}{2} = 2$$

The output SNR of the Wiener filter is given by:

$$\mathrm{SNR}_o = \frac{h_{\mathrm{opt}}^T R_s h_{\mathrm{opt}}}{h_{\mathrm{opt}}^T R_v h_{\mathrm{opt}}}$$

```
R_ss = toeplitz(r_ss);
R_vv = toeplitz(r_vv);
SNR_o = (h_opt'*R_ss*h_opt) / (h_opt'*R_vv*h_opt)
```

```
SNR_o = 2.0991
```

The filter increases the SNR but only slightly. This is expected and is an excellent way to check your results.