MÜHENDİSLİK FAKÜLTESİ
FACULTY OF ENGINEERING
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ BÖLÜMÜ
DEPARTMENT OF ELECTRICAL AND
ELECTRONICS ENGINEERING

ORTA DOĞU TEKNİK ÜNİVERSİTESİ
MIDDLE EAST TECHNICAL UNIVERSITY

DUMLUPINAR BULVARI 06800
ÇANKAYA ANKARA/TURKEY
T: +90 312 210 23 02
F: +90 312 210 23 04
ee@metu.edu.tr
www.eee.metu.edu.tr

EXPERIMENT 5. OPTIMUM FILTERING: FIR WIENER FILTER IMPLEMENTATION FOR NOISE REMOVAL

1. Introduction

In this experiment, optimum filtering concept is investigated. Optimum filtering uses the statistical characteristics of the input signal to extract the useful information. Hence, some statistical information regarding to the input and the desired signal should be known in order to apply this type of filters.

2. Preliminary Work

- Read the following information on optimum filtering.

2.1. Optimum Filtering: Wiener Filters

The most common types of filter encountered in practice are deterministic filters in the form of a lowpass, highpass, etc. While these filters are easily designed and used, they are far from performing in the best manner especially when there is some information regarding to the signal statistics. Optimum filters perform the best with respect to a certain optimality criteria and they are designed using the signal statistics. While there are several other optimality criteria, we concentrate on a special one, namely the mean square error, MSE. In the following part, the theoretical background for MSE optimum filters or Wiener Filters is presented.

2.2. Derivation of Wiener Filters

The problem for MSE optimum filters can be outlined as follows. Given the input sequence, $x[n]$, desired response, $d[n]$ and the form of the filter structure, i.e.,

$$y[n] = \sum_{k=0}^{\infty} h[k]x[n-k], \quad n = 0,1,... \qquad (1)$$

the target is to find filter coefficients, $h[k]$, such that the MSE is minimized. Note that, y[n] is the Wiener filter output. While it is possible to have IIR Wiener filters, FIR Wiener filters are considered in the following parts for simplicity. The definitions for the error and MSE are given as follows, i.e.,

$$e[n] = d[n] - y[n] = d[n] - \sum_{k=0}^{\infty} h[k]x[n-k]$$
$$MSE = J = E\{e[n]^2\} \quad (2)$$

where $E\{.\}$ is the expectation operator. In order to find the optimum coefficients, derivative of J with respect to $h[k]$ should be equated to zero, i.e.,

$$\nabla_k J = \frac{\partial J}{\partial h_k} = 2E\left\{e[n]\frac{\partial e[n]}{\partial h_k}\right\} = -2E\{e[n]x[n-k]\} = 0 \quad (3)$$

Above is the principle of orthogonality. In other words, J attains its minimum iff the estimation error $e[n]$ is orthogonal to the input, $x[n]$. If we insert, $e[n]$ into (3), we obtain,

$$E\left\{\left(d[n] - \sum_{i=0}^{\infty} h[i]x[n-i]\right)x[n-k]\right\} = 0$$
$$\sum_{i=0}^{\infty} h[i]E\{x[n-i]x[n-k]\} = E\{d[n]x[n-k]\} \quad (4)$$
$$\sum_{i=0}^{\infty} h[i]R_x[k-i] = r_{dx}[k], \quad k = 0,1,2,....$$

In the above equation, $R_x[.]$ is the autocorrelation function of the input and $r_{dx}[.]$ is the cross-correlation function of the desired and input signals. Let $\mathbf{R_x}$ denote the autocorrelation matrix of the input where $R_x[k-i]$ is the k^{th} row and i^{th} column element of $\mathbf{R_x}$. (Here, index of columns and zeros of autocorrelation matrix are assumed to start from index 0). Let $\mathbf{r_{dx}}$ is the cross-correlation vector of the desired and input signals where $r_{dx}[k]$ is the k^{th} element of $\mathbf{r_{dx}}$, again starting from 0 index. The last equation in (4) is called as the Wiener-Hopf equation and can be more compactly written in terms of matrix-vector notation as,

$$\mathbf{R_x h} = \mathbf{r_{dx}} \quad (5)$$

where $\mathbf{h}$ is the vector whose elements are $h[i]$.
The minimum MSE for the Wiener filtering is given as,

$$MSE_o = \sigma_e^2 = R_d[0] - \mathbf{h}^{\mathbf{T}}\mathbf{r_{dx}} = R_d[0] - \sum_{m=0}^{P-1} h[m]r_{dx}[m] \quad (6)$$

where $R_d[0]$ is the 0th term of the autocorrelation sequence of the desired signal, and P is the length of the FIR Wiener filter. Hence (6) can be used to compute theoretical minimum MSE. It is also possible to compute least squares error which is a deterministic counterpart of MSE which is computed by using signal samples, i.e.,

$$LSE = \frac{1}{N}\sum_{n=0}^{N-1} e[n]^2 \qquad (7)$$

2.3. Applications of Wiener Filters

Wiener filters can be used in a variety of applications. Filtering of a signal in noise (or noise removal), prediction of a signal in noise, smoothing of a signal in noise, and linear prediction are some examples of these applications. In this experiment, two applications namely the filtering of a signal in noise and prediction are considered. In the following part, examples of such applications are given. However, first a signal with a known correlation function is discussed.

2.3.1. Innovations Process

Innovations process is obtained as the output of a filter when the filter input is white Gaussian noise.

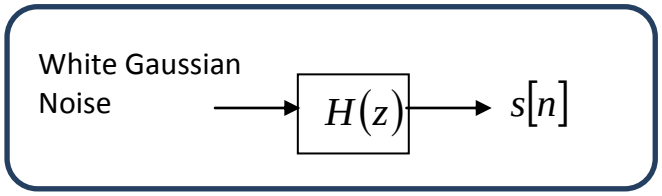

Fig.1. Generation of the innovations process, s[n].

Let $H(z)$ be selected as,

$$H(z) = \frac{1}{1-0.2z^{-1}} \qquad (8)$$

Then it is possible to show that the autocorrelation sequence for $s[n]$ is given as,

$$R_s[m] = \frac{1}{1-(0.2)^2}(0.2)^{|m|} = \frac{1}{0.96}(0.2)^{|m|} \qquad (9)$$

Assume that signal is corrupted by white Gaussian sequence $v[n]$ where $R_v[m] = \sigma_v^2\delta[m]$. Therefore observed signal, $x[n]$, is given as,

$$x[n] = s[n] + v[n] \qquad (10)$$

Noise is uncorrelated with the signal $s[n]$.

2.3.2. Filtering of a signal in noise

In this case, desired signal is $d[n]=s[n]$ and $R_d[m]=R_s[m]$. Note that the problem is as follows. Given the noisy signal $x[n]$, its correlation function, $R_x[m]$ and the cross-correlation $r_{dx}[m]$, find the MSE optimum filter coefficients $h[k]$ such that the filter output is the desired signal. The desired filter length is given as $P=3$. It is possible to write $r_{dx}[m]$ and $R_x[m]$ as follows, i.e.,

$$r_{dx}[m]=E\{s[n]x[n-m]\}=E\{s[n](s[n-m]+v[n-m])\}=R_s[m]=\frac{1}{0.96}(0.2)^{|m|} \qquad (11)$$

$$R_x[m]=E\{(s[n]+v[n])(s[n-m]+v[n-m])\}=R_s[m]+R_v[m]=\frac{1}{0.96}(0.2)^{|m|}+\sigma_v^2\delta[m] \quad (12)$$

Note that $v[n]$ and $s[n]$are assumed to be uncorrelated. Using the Wiener-Hopf equations in (4) or (5), we obtain,

$$\begin{bmatrix} R_x[0] & R_x[-1] & R_x[-2] \\ R_x[1] & R_x[0] & R_x[-1] \\ R_x[2] & R_x[1] & R_x[0] \end{bmatrix}\begin{bmatrix} h[0] \\ h[1] \\ h[2] \end{bmatrix}=\begin{bmatrix} r_{dx}[0] \\ r_{dx}[1] \\ r_{dx}[2] \end{bmatrix} \quad (13)$$

Note that correlation matrix is Hermitian symmetric, i.e., $R_x[-1]=R_x^*[1]$. Once the numerical values for the above expression are used, Wiener filter coefficients can be easily obtained through matrix inversion, i.e., $\mathbf{h_{opt}}=\mathbf{R_x^{-1}r_{dx}}$. Theoretical MSE is obtained from (6).

2.3.3. Prediction of a signal in noise

In this case, the problem is to predict the signal samples before K-steps. Hence the desired signal is $d[n]=s[n+K]$ where K=2 is selected for simplicity. $R_x[m]$ does not change and it is the same as in (12). $r_{dx}[m]$ changes and it is given below,

$$r_{dx}[m]=E\{s[n+2]x[n-m]\}=E\{s[n+2](s[n-m]+v[n-m])\}=R_s[m+2]=\frac{1}{0.96}(0.2)^{|m+2|} \quad (14)$$

It is possible to write a similar expression to (13) in this case and find the Wiener filter coefficients. MSE in this case is

$$MSE=R_d[0]-\sum_{m=0}^{2}h[m]\frac{1}{0.96}(0.2)^{|m+2|} \quad (15)$$

3. Experimental Work

PART 1

MATLAB Programming Tasks

- Write a program for optimum filtering.

i) The inputs for this program are:
- length of the input signal, N,
- length of the optimum filter, P,
- pole of the innovations filter, a, $\left(H(z) = \frac{1}{1 - az^{-1}} \right)$
- standard deviation of the noise, σ_v,
- the prediction step, P_{step}.

ii) Take P_{step}=0 for filtering of a signal in noise.

iii) Obtain the desired signal as shown in Fig. 1 and equation (8).

iv) Add noise to obtain $x[n] = s[n] + v[n]$.

v) Find $\mathbf{R_x}$, $\mathbf{r_{dx}}$ and compute **h** from (5).

vi) Obtain Wiener filter output, y[n], by convolving x[n] with h[n]. Obtain the LSE from the samples using (7). Compare it with MSE using the equation (6). Plot s[n], x[n], y[n], and e[n].

vii) Design a lowpass filter using windowing technique (or Parks-Mcclellan algorithm) and filter x[n] with this new filter to obtain g[n]. Compute LSE for g[n] and compare with the previous results. Explain your findings.

viii) Repeat iii-vii for prediction of a signal in noise by selecting P_{step}=2. Explain your plots and findings. What happens when P_{step} is increased?

PART 2

Real-time Programming Task

In this part, you need to program MyRIO for the optimum filtering similar to the MATLAB implementation. The programming can use MyRIO CPU and hence the project structure may look like as shown in Fig. 2.

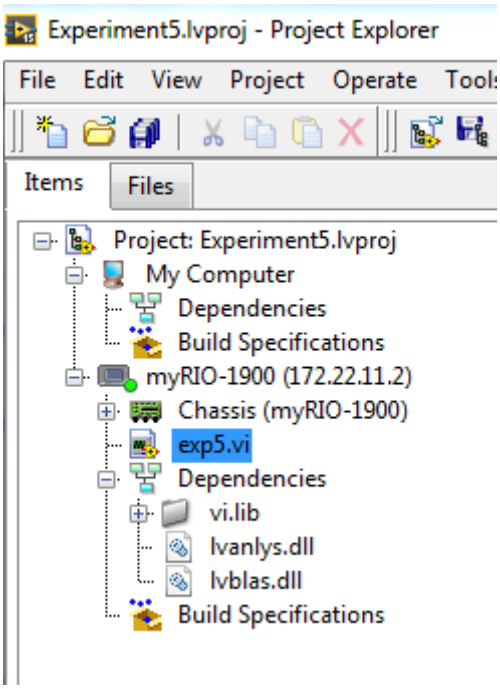

Fig. 2. Project Structure for Experiment 5.

The front panel for the Experiment 5 may look like the one shown in Fig. 3. The input parameters for the system are described in MATLAB section and they are the same in MyRIO implementation. ***"milliseconds to wait"*** is added to slow down the execution in order to have a better visualization of the waveforms. The Wiener filter coefficients are reported as in Fig.3. The theoretical MSE, LSE of the Wiener filter and LSE of the lowpass Parks-Mcclellan (PM) filter are shown as well. The parameters of the PM filter are also presented. The two plots show the time response of the Wiener filter as well as the error signals.

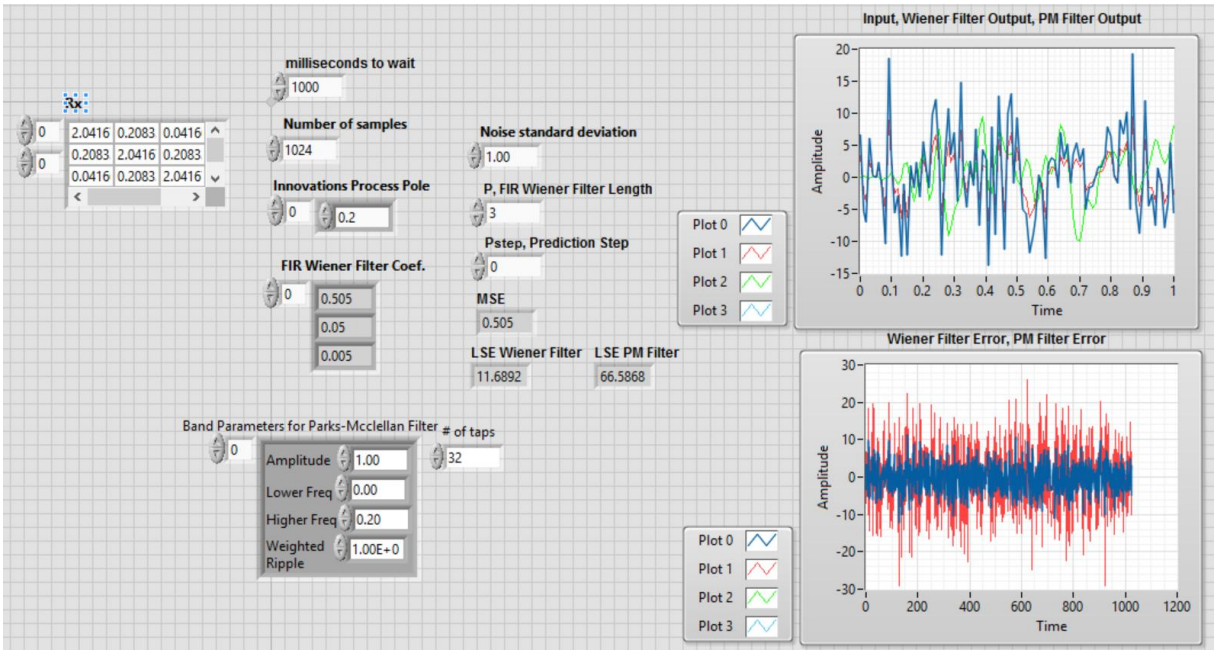

Fig. 3. Experiment 5 front panel.

- You can use ***Gaussian White Noise.vi*** function in Fig. 4 to generate random Gaussian white noise.

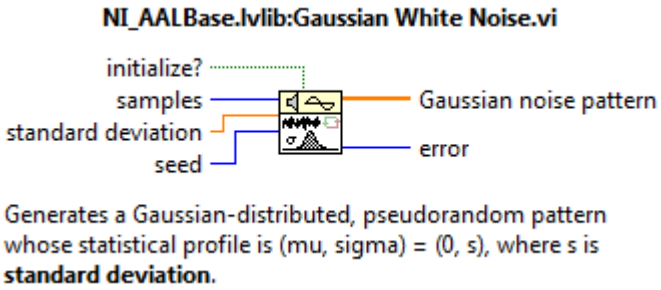

Fig. 4. Gaussian White Noise.

- You can use ***IIR Fiter.vi*** and ***FIR Filter.vi*** functions shown in Fig. 5, respectively.

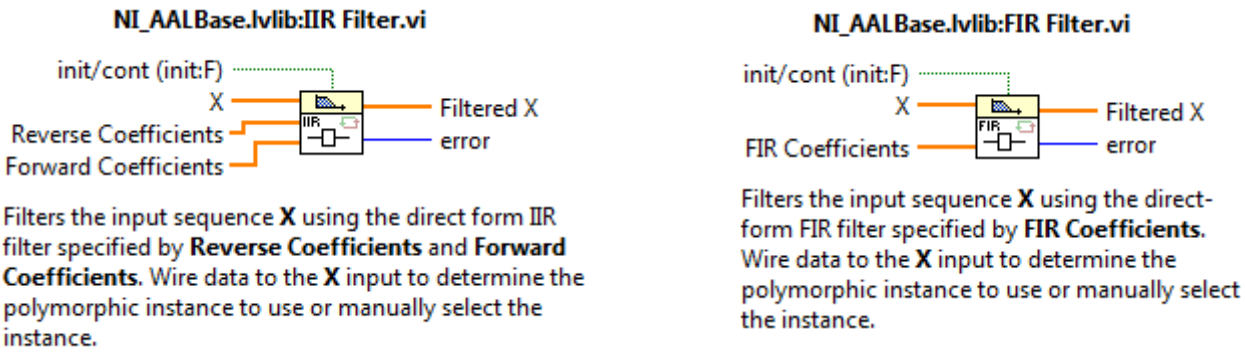

Fig. 5. IIR and FIR filters.

- You can use ***Parks-Mcclellan.vi*** function as shown in Fig. 6.

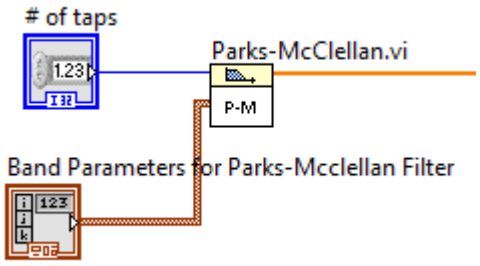

Fig. 6. Parks-McClellan filter.

- In order to generate autocorrelation matrix of the input, you can use ***Create Special Matrix.vi*** shown in Fig. 7 with ***matrix type*** as ***Toeplitz***.

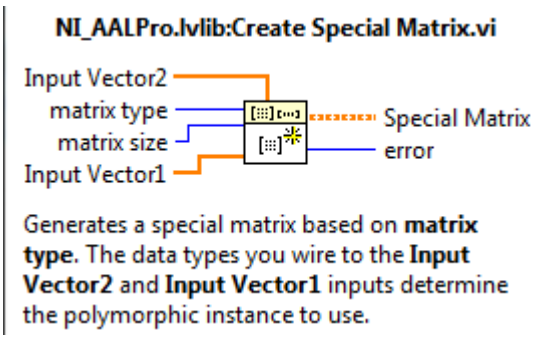

Fig. 7. Create Special Matrix.vi.

- You can use ***Inverse Matrix.vi*** and ***General Matrix-Vector Product.vi*** functions shown in Fig. 8 for matrix operations in optimum filtering.

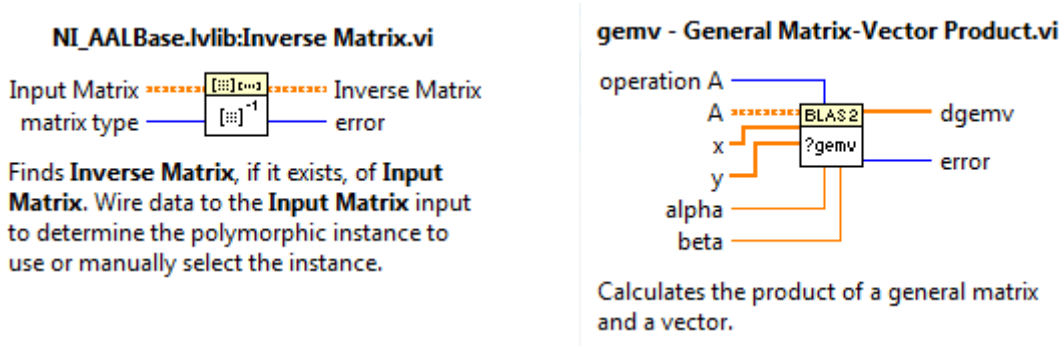

Fig. 8. Inverse Matrix.vi and General Matrix-Vector Product.vi.

Programming Tasks

a) Write the LabVIEW program for optimum filtering as described above.

b) In addition to the two plots in Fig. 3, add two plots for the frequency characteristics of the Wiener filter and PM filter.

c) Let the innovations process pole, a=0.8, Wiener filter length=3 and P_{step}=0. Obtain the filter coefficients, MSE and LSE values. Also include the waveform plots in your report. Can you design a PM filter which gives better LSE than Wiener filter?

d) Let P_{step}=2 and repeat c. Explain the differences between the results obtained in c and d.

e) Increase P_{step} to 100. Explain your observations.

f) Now increase the length of the Wiener filter one by one by noting the MSE. Explain the MSE result as the filter length increases.