Rensselaer Polytechnic Institute
Department of Electrical, Computer, and Systems Engineering
ECSE 4530: Digital Signal Processing, Fall 2020

Homework #6: due Monday, Dec. 14th, at the beginning of class.

1. (10 points) Use the bilinear transformation with $T = 0.1$ to convert the analog filter with transfer function (in the Laplace domain)

$$H(s) = \frac{s+0.1}{(s+0.1)^2+9}$$

into a digital IIR filter. Compare the locations of the zeros in $H(z)$ with the locations of the zeros obtained by applying the impulse invariance method in the conversion of $H(s)$.

The bilinear transform involves plugging in $s = \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}} = 20\frac{1-z^{-1}}{1+z^{-1}}$ into the transfer function

$$H(s) = \frac{s+0.1}{(s+0.1)^2+9}$$

That is,

$$\begin{aligned}
H_{\text{bilinear}}(z) &= \frac{20\frac{1-z^{-1}}{1+z^{-1}}+0.1}{\left(20\frac{1-z^{-1}}{1+z^{-1}}+0.1\right)^2+9}\\
&= \frac{20(1-z^{-1})(1+z^{-1})+0.1(1+z^{-1})^2}{(20(1-z^{-1})+0.1(1+z^{-1}))^2+9(1+z^{-1})^2}\\
&= \frac{20(1-z^{-2})+0.1(1+2z^{-1}+z^{-2})}{400(1-z^{-1})^2+4(1-z^{-2})+9.01(1+z^{-1})^2}\\
&= \frac{20.1+0.2z^{-1}-19.9z^{-2}}{413.01-781.98z^{-1}+405.01z^{-2}}\\
&= \frac{(1+z^{-1})(1-0.995z^{-1})}{1-2az^{-1}+(a^2+b^2)z^{-2}}
\end{aligned}$$

where $a = 0.9467$, $b = 0.2905$, i.e. zeros at -1 and 0.995 and poles at $0.9467 \pm j0.2905$. Let $r = \sqrt{a^2+b^2} = 0.99$ and $a = r\cos(\omega_0) = 0.9467$. Solving $\cos(\omega_0) = 0.9467/0.99$ gives $\omega_0 = 0.3$. Therefore, the transfer function for the digital filter is given as

$$H_{\text{bilinear}}(z) = \frac{(1+z^{-1})(1-rz^{-1})}{1-2r\cos(\omega_0)z^{-1}+r^2z^{-2}}, \quad \omega_0 = 0.3, \quad r = 0.99.$$

$$H(s) = \frac{1}{2}\left[\frac{1}{s+0.1-j3}+\frac{1}{s+0.1+j3}\right]$$

$H(s)$ has two zeros at -0.1 and ∞ and two poles $-0.1 \pm j3$. The matched z-transform for the impulse invariance (which involves plugging in $z = e^{sT}$) maps these into:

$$\begin{aligned}
\tilde{z}_1 &= e^{-0.1T} = e^{-0.01} = 0.99\\
\tilde{z}_2 &= e^{-\infty T} = 0\\
\tilde{p}_1 &= e^{(-0.1+j3)T} = 0.99e^{j0.3}\\
\tilde{p}_2 &= 0.99e^{-j0.3}.
\end{aligned}$$

From the impulse invariance method we obtain

$$\begin{aligned}
H_{\text{impulse invariance}}(z) &= \frac{1}{2}\left[\frac{1}{1-e^{-0.1T}e^{j3T}z^{-1}} + \frac{1}{1-e^{-0.1T}e^{-j3T}z^{-1}}\right] \\
&= \frac{1 - r\cos(\omega_0)z^{-1}}{1-2r\cos(\omega_0)z^{-1}+r^2z^{-2}}, \quad \omega_0 = 0.3, \quad r = 0.99.
\end{aligned}$$

The poles are the same, but the zero is different.

2. (30 points) Consider an autoregressive process that satisfies the difference equation

$$x[n] = 0.6x[n-1] + v[n]$$

where $v[n]$ is a white noise process with variance $\sigma_v^2 = 0.64$. Assume further that we observe the signal $y[n]$ which is given by

$$y[n] = x[n] + w[n]$$

where $w[n]$ is a white noise process with variance $\sigma_w^2 = 1$.

(a) Compute the correlation coefficients $r_x[0]$ and $r_x[1]$ for $\{x[n]\}$ by using the Yule-Walker equations. Then, obtain the value of $r_x[0]$ by using the equation for the variance of the white noise process in terms of the $r_x[j]$.
We write the Yule-Walker equations for $M = 2$ for the process $x[n]$. We have the 1 dimensional matrix

$$r_x[0]a_1 = -r_x[1].$$

Hence, we have the following 2 equations:

$$\begin{aligned}
&r_x[0] + a_1 r_x[1] = 0.64 \quad \text{(variance of the noise process } v[n]) \\
&a_1 r_x[0] + r_x[1] = 0
\end{aligned}$$

where $a_1 = -0.6$ (in Yule-Walker equations (Lecture 22)). Solving above equations we get

$$r_x[0] = 1, \quad r_x[1] = 0.6$$

(b) Design a Wiener filter with length $M = 2$ to estimate $\{x[n]\}$.
Note that $y[n] = x[n] + w[n]$. Denoting by $r[0]$ and $r[1]$ the autocorrelations for $y[n]$,

$$r[0] = \mathbb{E}[y[n]y[n]] = \mathbb{E}[(x[n]+w[n])^2] = r_x(0) + \sigma_w^2 = 2$$

and

$$r[1] = \mathbb{E}[y[n]y[n-1]] = \mathbb{E}[(x[n]+w[n])(x[n-1]+w[n-1])] = r_x(1) = 0.6$$

where $w[n-1]$ and $w[n]$ are independent.
The Wiener filter coefficients satisfy

$$\begin{bmatrix} r[0] & r[1] \\ r[1] & r[0] \end{bmatrix}\begin{bmatrix} \hat{h}[0] \\ \hat{h}[1] \end{bmatrix} = \begin{bmatrix} p[0] \\ p[-1] \end{bmatrix}.$$

where $\sum_{i=0}^{1} \hat{h}[i]r[i-k] = \hat{h}[0]r[k] + \hat{h}[1]r[1-k] = p[-k]$ for $k = 0, 1$.

$$2\hat{h}[0] + \hat{h}[1]0.6 = p[0] = E[y[0]x[0]] = r[0] = 1$$
$$\hat{h}[0]0.6 + 2\hat{h}[1] = p[-1] = E[y[0]x[-1]] = r[1] = 0.6$$

Solving above equations, we get

$$\hat{h}[0] = 0.451, \quad \hat{h}[1] = 0.165.$$

(c) Determine the minimum mean square error (MMSE) for $M = 2$.
The corresponding MMSE is

$$1 - \hat{h}[0]r[0] - \hat{h}[1]r[1] = 1 - 0.451 - 0.165 \cdot 0.6 = 0.45$$

The error can be reduced by increasing the number of filter taps M.

3. (60 points) The LMS (least-mean-square) and the RLS (recursive-least-square) algorithms.
Consider the setup in Figure 1 where

- Input signal x$[n]$ is a zero mean and independent and identically distributed (i.i.d.) wide sense stationary (WSS) process with variance 1.
Assume that $x_n = [x[0], \ldots, x[N-1]^\mathsf{T}]$. Then, the autocorrelation matrix of x_n is given by

$$R_x = \mathbb{E}[x_n x_n^\mathsf{T}] = \begin{bmatrix} \mathbb{E}[x[0]x[0]] & \mathbb{E}[x[0]x[1]] & \ldots & \mathbb{E}[x[0]x[N-1]] \\ \mathbb{E}[x[1]x[0]] & \mathbb{E}[x[1]x[1]] & \ldots & \mathbb{E}[x[1]x[N-1]] \\ \vdots & \vdots & \vdots & \vdots \\ \mathbb{E}[x[N-1]x[0]] & \mathbb{E}[x[N-1]x[1]] & \ldots & \mathbb{E}[x[N-1]x[N-1]] \end{bmatrix} = \begin{bmatrix} 1 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 \end{bmatrix} = I$$

where I is an $N \times N$ identity matrix. This is because $\mathbb{E}[x[n]x[m]] = \mathbb{E}[x[n]]\mathbb{E}[x[m]] = 0$ when $n \neq m$ and $\mathbb{E}[x[n]^2] = Var(x[n]) = 1$.
- Noise v$[n]$ is a zero mean and i.i.d. WSS process with variance $\sigma_v^2 = 10^{-4}$.
- Observation signal y$[n]$ is the output of the system with frequency response H(ω) plus noise v$[n]$.
Defining $y_n = [y[0], \ldots, y[N-1]^\mathsf{T}]$, and $v_n = [v[0], \ldots, v[N-1]^\mathsf{T}]$ and x_n as above, we have

$$y_n = Hx_n + v_n,$$

where

$$H = \begin{bmatrix} h[0] & 0 & \ldots & 0 \\ h[1] & h[0] & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ h[N-1] & h[N-2] & \ldots & h[0] \end{bmatrix}.$$

Therefore, the autocorrelation matrix for y_n becomes

$$\begin{aligned} R_y = \mathbb{E}[y_n y_n^\mathsf{T}] &= \mathbb{E}[(Hx_n + v_n)(Hx_n + v_n)^\mathsf{T}] \\ &= \mathbb{E}[(Hx_n + v_n)(x_n^\mathsf{T} H^\mathsf{T} + v_n^\mathsf{T})] \\ &= \mathbb{E}[Hx_n x_n^\mathsf{T} H^\mathsf{T} + Hx_n v_n^\mathsf{T} + v_n x_n^\mathsf{T} H^\mathsf{T} + v_n v_n^\mathsf{T}] \\ &= H\mathbb{E}[x_n x_n^\mathsf{T}]H^\mathsf{T} + H\mathbb{E}[x_n v_n^\mathsf{T}] + \mathbb{E}[v_n x_n^\mathsf{T}]H^\mathsf{T} + \mathbb{E}[v_n v_n^\mathsf{T}] \\ &= HR_x H^\mathsf{T} + \sigma_v^2 I \\ &= HH^\mathsf{T} + \sigma_v^2 I. \end{aligned}$$

Last step follows from that $R_x = I$.

- $\mathrm{W}(\omega)$ is an FIR filter with coefficients $\mathrm{w}[0], \ldots, \mathrm{w}[N-1]$ that produces a delayed estimate of the input $\mathrm{x}[n]$, i.e. $\hat{x}[n-n_0]$ where n_0 is the delay.
  The optimal filter coefficients are given by the Wiener-Hopf equations:

$$w^* = R_y^{-1} p_y,$$

where R_y is the autocorrelation matrix for the input process (to the Wiener filter), and p_y is the cross correlation vector between the input process and the desired output $x[n-n_0]$:

$$\begin{aligned} p_y &= \mathbb{E}[y_n x[n-n_0]] = \mathbb{E}[(Hx_n + v_n)x[n-n_0]] \\ &= \mathbb{E}[Hx_n x[n-n_0] + v_n x[n-n_0]] \\ &= \mathbb{E}[Hx_n x[n-n_0]] + \mathbb{E}[v_n x[n-n_0]] \\ &= H\mathbb{E}\left[\begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{bmatrix} x[n-n_0]\right] = Hp, \end{aligned}$$

where using the notation $r[i] = \mathbb{E}[x[n+i]x[n]]$, the vector p is given by

$$p = \begin{bmatrix} r(n-n_o) \\ r(n-n_o-1) \\ \vdots \\ r(0) \\ r(-1) \\ \vdots \\ r(N-1-n+n_0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

i.e. only the $n_o + 1$st entry of p is 1 and the remaining entries are 0.

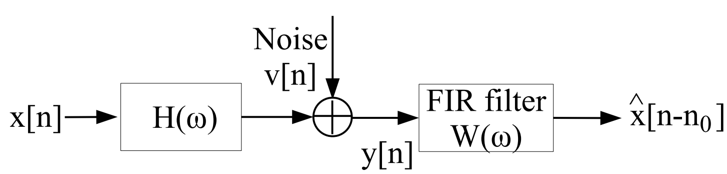


Figure 1: A linear system to recover $x[n]$ (also known as an equalizer).

(a) Write a MATLAB function to calculate MMSE FIR filter coefficients. Below is the prototype for the function:

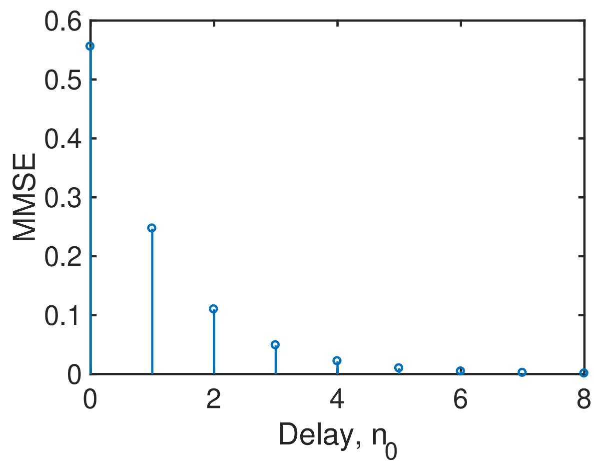
```
function [w, mmse]=findmmsefireq(h,Var,N,no)
% MATLAB function that calculates the MMSE FIR filter
% h : Impulse response of the linear system
% Var : Variance of the iid noise
% N : Length of the FIR filter
```

```
% no : Output delay
% w : FIR coefficients as output
% mmse: Minimum Mean Square Error as output
% This function assumes that the input signal x[n] has variance 1.
```

NOTE: the MATLAB codes for all parts are uploaded on Piazza. Read them carefully. The MSE plots are done in log scale.

(b) Given $h = [1, 1.5]$ and the FIR filter $W(\omega)$ with length $N = 10$, calculate MMSE for $n_0 = 0$ to 8 and plot it as a function of delay.

The MMSE as function of delay n_0 is shown below. As the delay of the desired signal increases, MMSE decreases since the adaptive model can tolerate higher delay and can provide a better estimate.

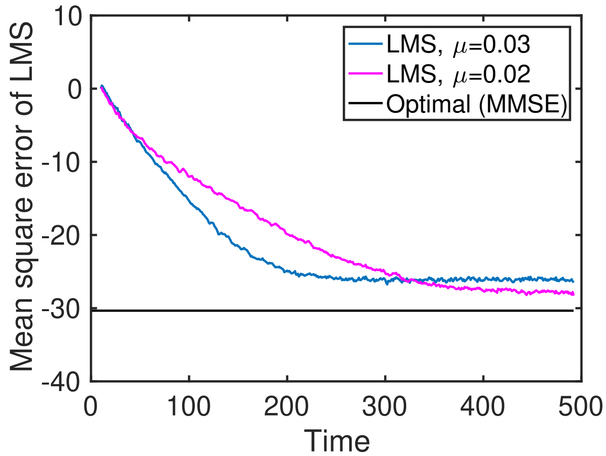


(c) For the same frequency response $\mathrm{H}(\omega)$,

- Generate i.i.d $\mathrm{x}[n]$'s with values -1 and 1.
- Filter it through $\mathrm{H}(\omega)$.
- Add $\mathrm{v}[n]$ to the output to obtain the observation signal $y[n]$. For $n_0 = 8$ and $\mu = 0.03$ perform LMS for 500 samples and obtain square error convergence curve as a function of time. Repeat this for 1000 times to average the square error curves.

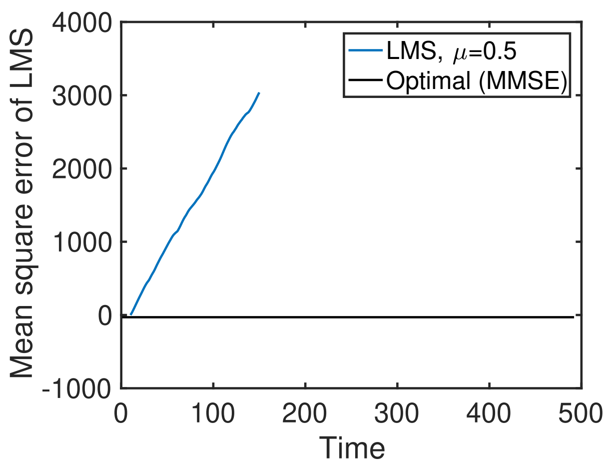
(d) Repeat part (c) for $\mu = 0.02$. Plot mmse convergence curves for part (c) and part (d) on the same graph (plot $10\log_{10}(mmse)$). On the same plot draw a line indicating the optimal mmse level. What are the excess MMSE and convergence times in each case.

The plot for parts (c)-(d) is shown below. We see that the mean square error of LMS decreases with time. As the step size μ increases from 0.02 to 0.03 we expect faster convergence. However, as shown in the plot, the mean square error for $\mu = 0.03$ is higher. The mean square error for optimal filter (i.e. the MMMS of part b) is also shown in black.

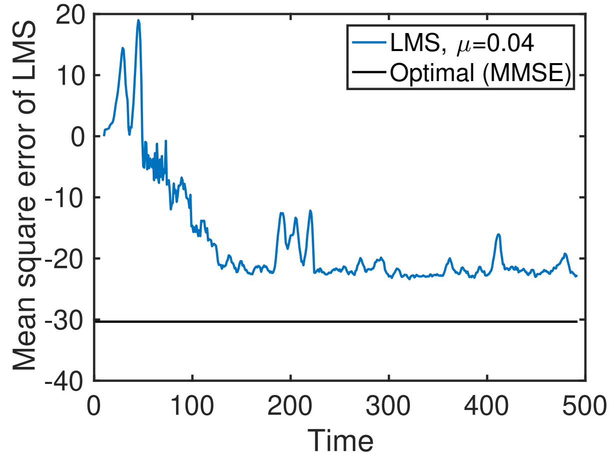


(e) Find the convergence curve for $\mu = 0.5$. What is your observation? Explain the reason for this behavior based on the statistics of the observation signal $y[n]$. Test to find the μ value which is the border value for convergence.

Here, the step size is much higher than the step sizes in the previous parts, which can cause overshooting. Not surprisingly, the LMS algorithm cannot converge and the MSE grows unboundedly.



Playing with the value of μ (the border value of μ for convergence should be somewhere in between 0.03 and 0.5), it seems that $\mu = 0.04$ is nearly the border value.



(f) Apply the RLS algorithm to obtain adaptive filter coefficients. Choose λ (forgetting factor) very close to 1.

The details of the RLS algorithm are available in Lecture 25. When you run the MATLAB code, you will see that the MSE behaves as below.