ECE211 – Pset 8

Jonathan Lam

May 9, 2020

A real WSS signal $x[n]$ is modeled as:

$$x[n] = v[n] + 0.5v[n-1] + 0.4v[n-2] - 0.2x[n-1] - 0.8x[n-2]$$

where $v[n]$ is the 0-mean white noise with $\sigma_v^2 = 4$.

Questions

1. It has both AR ($x[n-N]$ terms) and MA ($v[n-N]$ terms) components, so it is ARMA.

2. Since it outputs x, it is the innovations filter (not the whitening filter).

3. Rearranging to solve for the inverse filter $v[n]$:

$$v[n] = x[n] + 0.2x[n-1] + 0.8x[n-2] - 0.5v[n-1] - 0.4v[n-2]$$

4. Transfer function:

$$H(z) = \frac{z^2 + 0.5z + 0.4}{z^2 + 0.2z + 0.8}$$

5. PSD:

$$S_x(\omega) = \sigma_v^2 \frac{|B(\omega)|^2}{|A(\omega))|^2} = 4\frac{\left|1 + 0.5e^{-j\omega} + 0.4e^{-2j\omega}\right|^2}{\left|1 + 0.2e^{-j\omega} + 0.8e^{-2j\omega}\right|^2}$$

PSD peak

(Part 4b-c) The peak of the pwelch-estimated PSD ($\omega_0 = 1.6322$), the PSD calculated from the transfer function ($\omega_0 = 1.6690$), and the angle of the pole ($\arg p = \pm 1.6828$) are very close (within a range of 0.05 for this specific x). This makes sense, as the pole makes the signal's magnitude (and thus power) blow up near the pole. Also note that the exact value of the calculated PSD peak and the argument of the pole do not have to be exactly the same; the behavior of the signal may make the exact position of the peak vary, but it should be fairly close to $\arg p$.

```
% 2
H = tf([1 0.5 0.4], [1 0.2 0.8])
```

```
H =

  s^2 + 0.5 s + 0.4
  -----------------
  s^2 + 0.2 s + 0.8

Continuous-time transfer function.
```

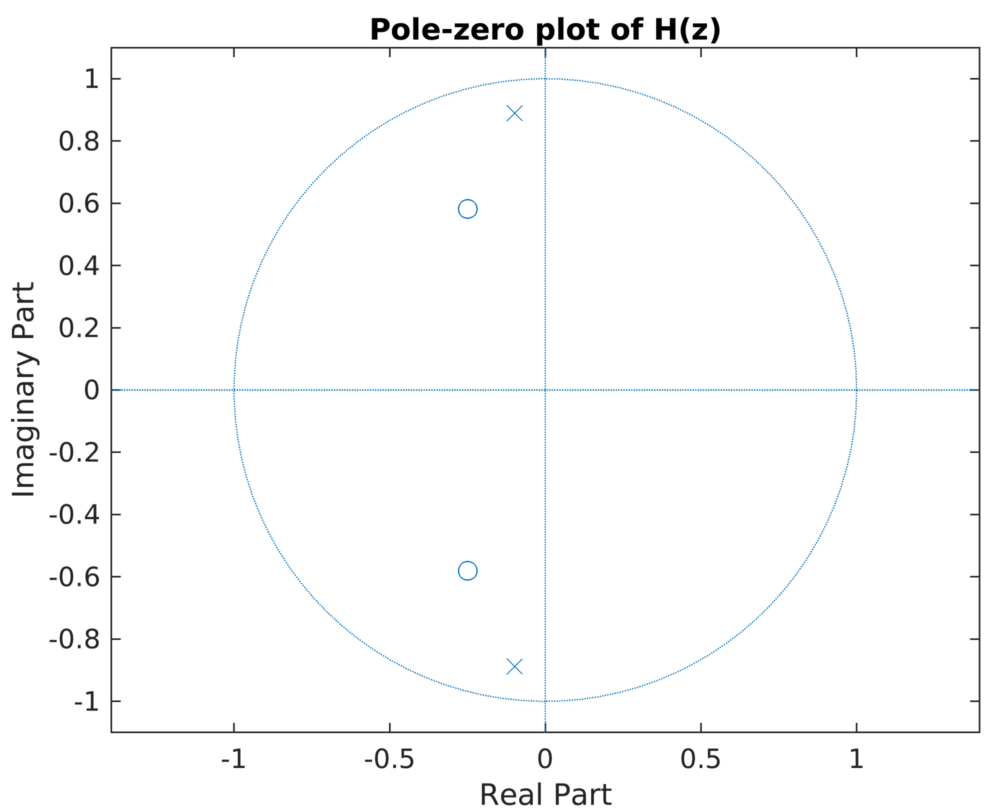
```
p = pole(H)
```

```
p = 2×1 complex
  -0.1000 + 0.8888i
  -0.1000 - 0.8888i
```

```
z = zero(H)
```

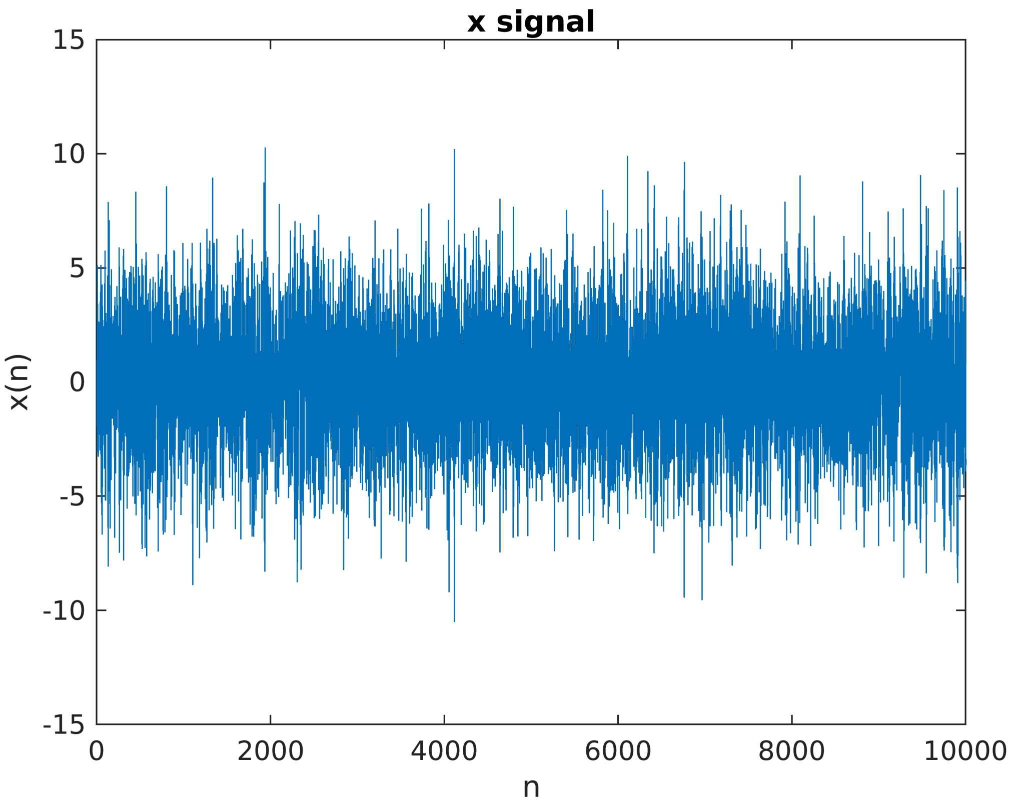
```
z = 2×1 complex
  -0.2500 + 0.5809i
  -0.2500 - 0.5809i
```

```
zplane(z, p)
title('Pole-zero plot of H(z)')
```

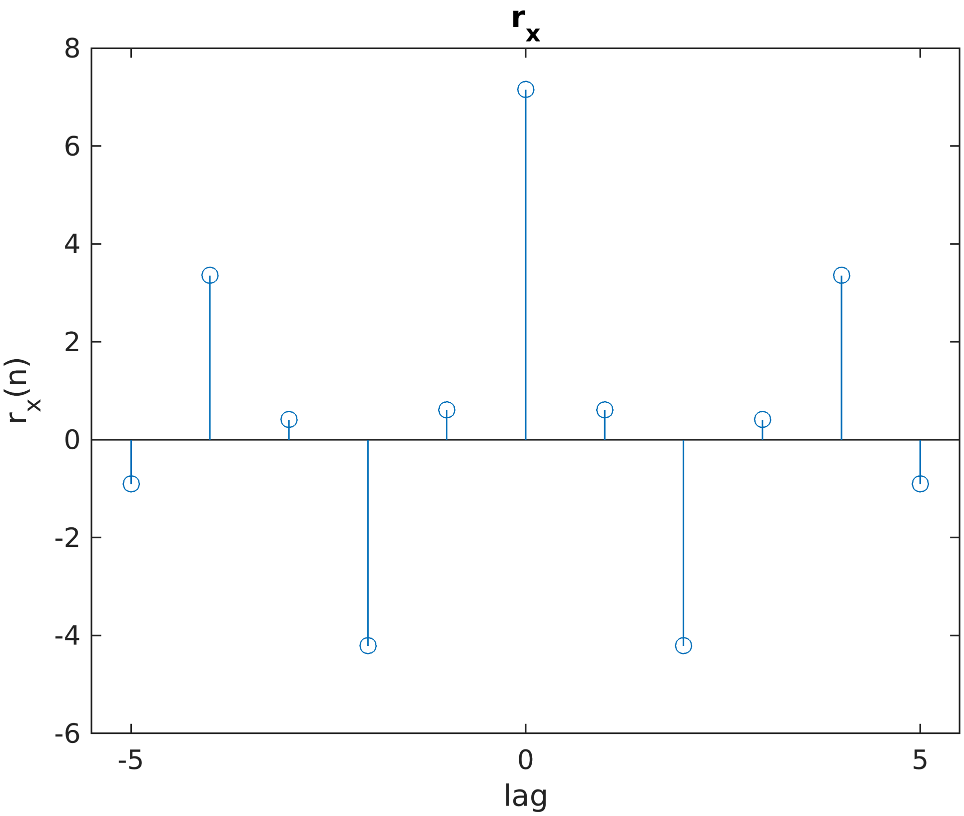



```
% 3a
N = 10000;
v = 2*randn(N, 1);
x = filter([1 0.5 0.4], [1 0.2 0.8], v);
```

```
plot(x);
title('x signal');
xlabel('n');
ylabel('x(n)');
```




```
% 3b) equivalent to doing:
% [xc, lags] = xcorr(x)
% stem(-5:5, xc(lags>=-5 & lags<=5))
lags = -5:5;
xcor = zeros([11 1]);
for lag=0:5
    xcor(lags==lag|lags==-lag) = dot(x(1:N-lag),x(lag+1:N))/(N-lag);
end
% 3c
stem(lags, xcor)
title('r_x')
xlabel('lag')
ylabel('r_x(n)')
xlim([-5.5 5.5])
```




```
% 3d
% M=6 here
tm = toeplitz(xcor(lags>=0))
```

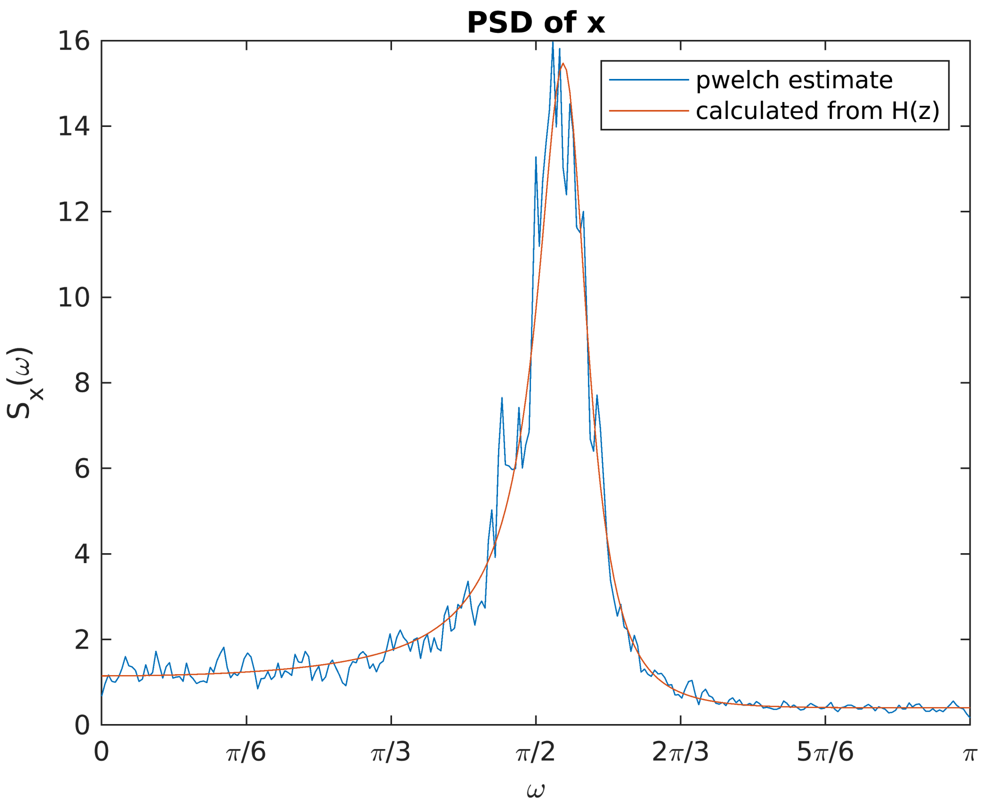
```
tm = 6×6
    7.1510    0.6043   -4.2092    0.4110    3.3563   -0.9066
    0.6043    7.1510    0.6043   -4.2092    0.4110    3.3563
   -4.2092    0.6043    7.1510    0.6043   -4.2092    0.4110
    0.4110   -4.2092    0.6043    7.1510    0.6043   -4.2092
    3.3563    0.4110   -4.2092    0.6043    7.1510    0.6043
   -0.9066    3.3563    0.4110   -4.2092    0.6043    7.1510
```

```
% 3e
ev = eig(tm)
```

```
ev = 6×1
    1.3598
    2.7045
    4.0701
    4.7076
   14.7870
   15.2768
```

```
% 4a
[s_est, w] = pwelch(x, hamming(512), 256, 512);
% sanity check; calculate from H(z) and normalize
s_x = 4*(abs(1+0.5*exp(-1j*w)+0.4*exp(-2j*w)) ...
    ./abs(1+0.2*exp(-1j*w)+0.8*exp(-2j*w))).^2/pi;
plot(w, s_est)
hold on;
```

```
plot(w, s_x)
hold off;
xlim([0 pi])
title('PSD of x')
xlabel('\omega')
xticks(0:pi/6:pi)
xticklabels({'0','\pi/6','\pi/3','\pi/2','2\pi/3','5\pi/6','\pi'})
ylabel('S_x(\omega)')
legend({'pwelch estimate', 'calculated from H(z)'})
```




```
% 4b
w_0_est = w(s_est==max(s_est))
```

```
w_0_est = 1.6322
```

```
w_0_calc = w(s_x==max(s_x))
```

```
w_0_calc = 1.6690
```

```
% 4c
angle(p)
```

```
ans = 2×1
    1.6828
   -1.6828
```