DSP Simulation Project #3

2018142023 조성민

1. AR (Auto Regressive) modeling

AR model의 coefficient $a[n]$는 다음과 같은 과정을 통해 구할 수 있습니다. 주어진 신호를 $x[n]$이라고 할 때, $a[n]$과 $x[n]$의 관계를 unknown input $u[n]$을 이용하여 다음과 같이 나타낼 수 있습니다.

$$x[n] = -\sum_{k=1}^{p} a[k]x[n-k] + u[n] \text{ for } \forall n$$

Orthogonality Principle에 의해 Autocorrelation $r_{xx}[m]$은 다음과 같습니다.

$$r_{xx}[m] = -\sum_{k=1}^{p} a[k]r_{xx}[m-k]$$

$$\begin{aligned}
\text{m=1}: \ & r_{xx}[1] = -a[1]r_{xx}[0] - a[2]r_{xx}[-1] - \dots - a[p]r_{xx}[-p+1] \\
\text{m=2}: \ & r_{xx}[2] = -a[1]r_{xx}[1] - a[2]r_{xx}[0] - \dots - a[p]r_{xx}[-p+2] \\
& \vdots \\
\text{m=p}: \ & r_{xx}[p] = -a[1]r_{xx}[p-1] - a[2]r_{xx}[p-2] - \dots - a[p]r_{xx}[0] \\
& \vdots
\end{aligned}$$

1-1. 위의 $m=1 \sim p$까지 나열된 식을 참고하여 R matrix (square matrix)를 구하고 이를 이용해 Yule-Walker equation을 나타내시오. (**보고서에 설계한 행렬식 첨부**)

Yule-Walker equation: $R \cdot a = r$

$$R = \begin{bmatrix} r_{xx}[0] & r_{xx}[-1] & r_{xx}[-2] & \dots & r_{xx}[-p+1] \\ r_{xx}[1] & r_{xx}[0] & r_{xx}[-1] & \dots & r_{xx}[-p+2] \\ r_{xx}[2] & r_{xx}[1] & r_{xx}[0] & \dots & r_{xx}[-p+3] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{xx}[p-1] & r_{xx}[p-2] & r_{xx}[p-3] & \dots & r_{xx}[0] \end{bmatrix}, \ a = \begin{bmatrix} -a[1] \\ -a[2] \\ -a[3] \\ \vdots \\ -a[p] \end{bmatrix}, \ r = \begin{bmatrix} r_{xx}[1] \\ r_{xx}[2] \\ r_{xx}[3] \\ \vdots \\ r_{xx}[p] \end{bmatrix}$$

where $r_{xx}[m] = -a[1]r_{xx}[m-1] - a[2]r_{xx}[m-2] - \dots - a[p]r_{xx}[m-p]$

1-2. $m=1\sim 2p$까지 $r_{xx}[m]$에 대한 식을 나열하고 $\mathbf{R}'$ matrix (non-square matrix)를 구하시오. (**보고서에 나열된 식과 설계한 행렬 첨부**)

$$m=1:\ r_{xx}[1]=-a[1]r_{xx}[0]-a[2]r_{xx}[-1]-\dots-a[p]r_{xx}[-p+1]$$
$$m=2:\ r_{xx}[2]=-a[1]r_{xx}[1]-a[2]r_{xx}[0]-\dots-a[p]r_{xx}[-p+2]$$
$$\vdots$$
$$m=p:\ r_{xx}[p]=-a[1]r_{xx}[p-1]-a[2]r_{xx}[p-2]-\dots-a[p]r_{xx}[0]$$
$$\vdots$$
$$m=2p:\ r_{xx}[2p]=-a[1]r_{xx}[2p-1]-a[2]r_{xx}[2p-2]-\dots-a[p]r_{xx}[p]$$

$$R'=\begin{bmatrix} r_{xx}[0] & r_{xx}[-1] & r_{xx}[-2] & \dots & r_{xx}[-p+1] \\ r_{xx}[1] & r_{xx}[0] & r_{xx}[-1] & \dots & r_{xx}[-p+2] \\ r_{xx}[2] & r_{xx}[1] & r_{xx}[0] & \dots & r_{xx}[-p+3] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{xx}[p-1] & r_{xx}[p-2] & r_{xx}[p-3] & \dots & r_{xx}[0] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{xx}[2p-1] & r_{xx}[2p-2] & r_{xx}[2p-3] & \dots & r_{xx}[p] \end{bmatrix}$$

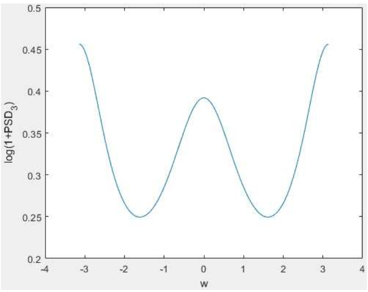
1-3. 식 (1)을 이용하여 각 입력신호에 대한 AR(3), AR(7), AR(12)의 coefficient $a[n]$을 구하시오. 3, 7, 12는 pole의 개수를 의미 ($a[0]=1$)

AR(3)=
```
 1.8311
-0.9115
 0.0756
```

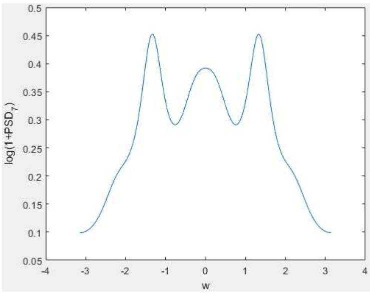
AR(7)=
```
 1.8816
-1.3255
 0.9800
-0.6160
 0.1279
-0.2026
 0.1497
```

AR(12)=
```
 2.0576
-1.7753
 1.5389
-1.2778
 0.8426
-0.9558
 1.2024
-1.2361
 1.1783
-1.0664
 0.6364
-0.1494
```

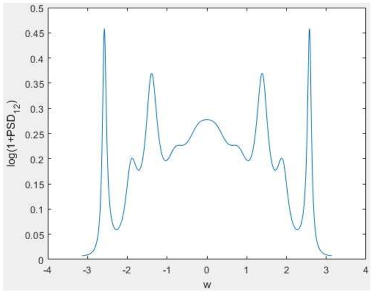
1-4. 위의 **1-3**에서 구한 계수들을 이용하여 PSD(Power Spectral Density)의 그래프를 log scale로 그리시오. ($a[0]=1$)



$PSD_3(w)$



$PSD_7(w)$



$PSD_{12}(w)$

$$\begin{aligned}\beta^2 &= \sum_{m=0}^{p} a[m]r_{xx}[-m] \\ &= a[0]r_{xx}[0] + \sum_{m=1}^{p} a[m]r_{xx}[-m] \\ &= r_{xx}[0] + a[1]r_{xx}[1] + a[2]r_{xx}[2] + \cdots + a[p]r_{xx}[p]\end{aligned}$$

1-5. 소문제 a), b), c)의 각 m의 범위에 대해 각 입력신호에 대한 AR(7)의 coefficient $a[n]$을 다음의 조건에 대해 구하시오. ($a[0]=1$)

a) 식(1), $2 \le m \le p+1$, p개의 Yule-Walker equation 사용
($r_{xx}[2], r_{xx}[3], \cdots, r_{xx}[p+1], p=7$)

b) 식(1), $3 \le m \le p+2$, p개의 Yule-Walker equation 사용
($r_{xx}[3], r_{xx}[4], \cdots, r_{xx}[p+2], p=7$)

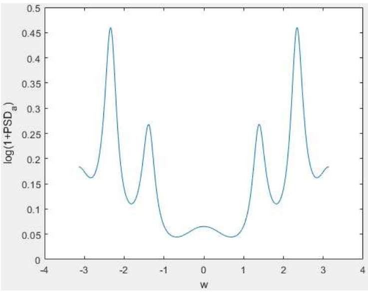
c) 식(2), $1 \le m \le 2p$, $2p$개의 Yule-Walker equation 사용
($r_{xx}[1], r_{xx}[2], \cdots, r_{xx}[2p], p=7$)

a) $a[n] = \begin{matrix} 0.8872 \\ 0.5574 \\ -0.3488 \\ 0.2870 \\ -0.3494 \\ -0.2743 \\ 0.2302 \end{matrix}$

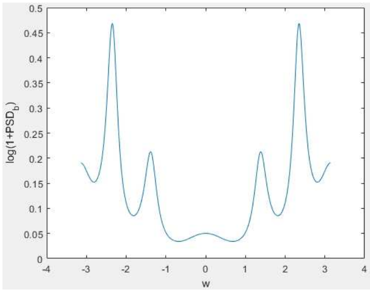
b) $a[n] = \begin{matrix} 0.8974 \\ 0.5195 \\ -0.2998 \\ 0.2519 \\ -0.3261 \\ -0.2846 \\ 0.2309 \end{matrix}$

c) $a[n] = \begin{matrix} 2.1895 \\ -2.1013 \\ 1.8775 \\ -1.4162 \\ 0.6941 \\ -0.4546 \\ 0.2073 \end{matrix}$

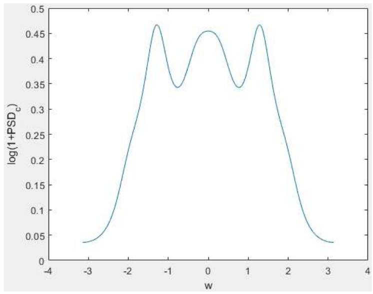
1-6. 위의 **1-5**에서 구한 계수들을 사용하여 PSD(Power Spectral Density) 그래프를 log scale로 그리시오.



$PSD_a(w)$

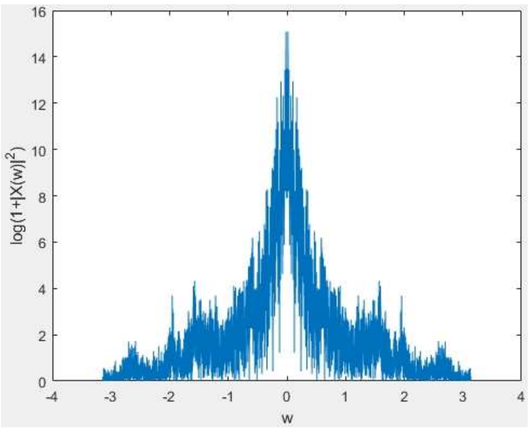


$PSD_b(w)$



$PSD_c(w)$

2. 주어진 음성신호의 원본에 대해 magnitude의 제곱 그래프 ($|X(\omega)|^2$ 그래프)를 log scale로 그린 후, 각 modeling의 PSD 그래프와 비교하시오. (AR modeling의 특징을 파악한 후 결과 그래프를 바탕으로 토의 과정을 자세히 작성하세요.)



$|X(w)|^2$

- $PSD_a(w)$와 $PSD_b(w)$를 보면 m의 시작점에 따라 coefficient $a[n]$이 달라지기 때문에 두 그래프의 형상이 서로 다른 것을 확인할 수 있다.

- 또한 $PSD_a(w)$와 $PSD_b(w)$를 구할 때 m의 범위가 충분히 크지 않기 때문에 $|X(w)|^2$의 형상과 차이가 많이 나는 것을 알 수 있다.

- $PSD_c(w)$를 $PSD_a(w)$, $PSD_b(w)$와 비교해보면 m의 범위가 늘어남에 따라 그래프의 형상이 $|X(w)|^2$에 가까워지는 것을 알 수 있다.

- 추가로 $PSD_c(w)$를 1-4번의 $PSD_{12}(w)$와 비교했을 때, $2p=14$이므로 $|X(w)|^2$와 형상이 더 비슷한 것을 확인할 수 있다.

<Appendix>

Q 1-3

```
clc;
clear;

[x,Fs]=audioread('x.wav');
N=length(x);

% rxx[0]
n=1:N;
z=x(n).*conj(x(n));
rxx_0=mean(z);

% AR(3)
p_3=3;
r_3=r_xx(x,p_3);
R_3=zeros(p_3,p_3);
for i=1:p_3
    for j=1:p_3
        k=abs(i-j);
        if k==0
            R_3(i,j)=rxx_0;
        else
            R_3(i,j)=r_3(k);
        end
    end
end
a_3=inv(R_3)*r_3;

% AR(7)
p_7=7;
r_7=r_xx(x,p_7);
R_7=zeros(p_7,p_7);
for i=1:p_7
    for j=1:p_7
        k=abs(i-j);
        if k==0
            R_7(i,j)=rxx_0;
        else
            R_7(i,j)=r_7(k);
        end
    end
end
a_7=inv(R_7)*r_7;

% AR(12)
p_12=12;
r_12=r_xx(x,p_12);
R_12=zeros(p_12,p_12);
for i=1:p_12
    for j=1:p_12
        k=abs(i-j);
        if k==0
            R_12(i,j)=rxx_0;
        else
            R_12(i,j)=r_12(k);
        end
    end
end
a_12=inv(R_12)*r_12;

disp(a_3)
disp(a_7)
disp(a_12)
```

Q 1-4

```
clc;
clear;

[x,Fs]=audioread('x.wav');
N=length(x);

% rxx[0]
n=1:N;
z=x(n).*conj(x(n));
rxx_0=mean(z);

% AR(3)
p_3=3;
r_3=r_xx(x,p_3);
R_3=zeros(p_3,p_3);
for i=1:p_3
    for j=1:p_3
        k=abs(i-j);
        if k==0
            R_3(i,j)=rxx_0;
        else
            R_3(i,j)=r_3(k);
        end
    end
end
a_3=inv(R_3)*r_3;

b=1;
Az_3=[1 a_3.'];
[Aw_3,w_3]=freqz(b,Az_3,'whole',Fs);
Aw_3=circshift(Aw_3,fix(Fs/2));
```

```
s_3=zeros(p_3,1);
for i=1:p_3
    s_3(i)=a_3(i).*r_3(i);
end
beta_3=rxx_0+sum(s_3);

P_3=beta_3.*(abs(Aw_3).^2);
alpha_3=1.5;

figure(1)
plot(w_3-pi,log(1+alpha_3.*P_3));
xlabel('w');
ylabel('log(1+PSD_3)');

% AR(7)
p_7=7;
r_7=r_xx(x,p_7);
R_7=zeros(p_7,p_7);
for i=1:p_7
    for j=1:p_7
        k=abs(i-j);
        if k==0
            R_7(i,j)=rxx_0;
        else
            R_7(i,j)=r_7(k);
        end
    end
end
a_7=inv(R_7)*r_7;

b=1;
Az_7=[1 a_7.'];
[Aw_7,w_7]=freqz(b,Az_7,'whole',Fs);
Aw_7=circshift(Aw_7,fix(Fs/2));
```

```
s_7=zeros(p_7,1);
for i=1:p_7
    s_7(i)=a_7(i).*r_7(i);
end
beta_7=rxx_0+sum(s_7);

P_7=beta_7.*(abs(Aw_7).^2);
alpha_7=1.5;

figure(2)
plot(w_7-pi,log(1+alpha_7.*P_7));
xlabel('w');
ylabel('log(1+PSD_7)');

% AR(12)
p_12=12;
r_12=r_xx(x,p_12);
R_12=zeros(p_12,p_12);
for i=1:p_12
    for j=1:p_12
        k=abs(i-j);
        if k==0
            R_12(i,j)=rxx_0;
        else
            R_12(i,j)=r_12(k);
        end
    end
end
a_12=inv(R_12)*r_12;

b=1;
Az_12=[1 a_12.'];
[Aw_12,w_12]=freqz(b,Az_12,'whole',Fs);
Aw_12=circshift(Aw_12,fix(Fs/2));
```

```
s_12=zeros(p_12,1);
for i=1:p_12
    s_12(i)=a_12(i).*r_12(i);
end
beta_12=rxx_0+sum(s_12);

P_12=beta_12.*(abs(Aw_12).^2);
alpha_12=1;

figure(3)
plot(w_12-pi,log(1+alpha_12.*P_12));
xlabel('w');
ylabel('log(1+PSD_1_2)');
```

Q 1-5

```
clc;
clear;

[x,Fs]=audioread('x.wav');
N=length(x);

% rxx[0]
n=1:N;
z=x(n).*conj(x(n));
rxx_0=mean(z);

p_7=7;
r_7=r_xx(x,p_7+1);
rs_7=zeros(p_7,1);
for i=1:p_7
    rs_7(i)=r_7(i+1);
end

R_7=zeros(p_7,p_7);
for i=1:p_7
    for j=1:p_7
        k=abs(1+i-j);
        if k==0
            R_7(i,j)=rxx_0;
        else
            R_7(i,j)=r_7(k);
        end
    end
end

a_7=inv(R_7)*rs_7;
disp(a_7);
```

(a)

```
clc;
clear;

[x,Fs]=audioread('x.wav');
N=length(x);

% rxx[0]
n=1:N;
z=x(n).*conj(x(n));
rxx_0=mean(z);

p_7=7;
r_7=r_xx(x,p_7+2);
rs_7=zeros(p_7,1);
for i=1:p_7
    rs_7(i)=r_7(i+2);
end

R_7=zeros(p_7,p_7);
for i=1:p_7
    for j=1:p_7
        k=abs(2+i-j);
        if k==0
            R_7(i,j)=rxx_0;
        else
            R_7(i,j)=r_7(k);
        end
    end
end

a_7=inv(R_7)*rs_7;
disp(a_7);
```

(b)

```
clc;
clear;

[x,Fs]=audioread('x.wav');
N=length(x);

% rxx[0]
n=1:N;
z=x(n).*conj(x(n));
rxx_0=mean(z);

p_7=7;
r_7=r_xx(x,2*p_7);
R_ns=zeros(2*p_7,p_7);
for i=1:2*p_7
    for j=1:p_7
        k=abs(i-j);
        if k==0
            R_ns(i,j)=rxx_0;
        else
            R_ns(i,j)=r_7(k);
        end
    end
end

R_s=R_ns.'*R_ns;
a=inv(R_s)*R_ns.'*r_7;
disp(a);
```

(c)

Q 1-6

```
clc;
clear;

[x,Fs]=audioread('x.wav');
N=length(x);

% rxx[0]
n=1:N;
z=x(n).*conj(x(n));
rxx_0=mean(z);

p_7=7;
% a
r_a=r_xx(x,p_7+1);
rs_a=zeros(p_7,1);
for i=1:p_7
    rs_a(i)=r_a(i+1);
end
R_a=zeros(p_7,p_7);
for i=1:p_7
    for j=1:p_7
        k=abs(1+i-j);
        if k==0
            R_a(i,j)=rxx_0;
        else
            R_a(i,j)=r_a(k);
        end
    end
end
a_a=inv(R_a)*rs_a;

b=1;
Az_a=[1 a_a.'];
[Aw_a,w_a]=freqz(b,Az_a,'whole',Fs);
Aw_a=circshift(Aw_a,fix(Fs/2));

s_a=zeros(p_7,1);
for i=1:p_7
    s_a(i)=a_a(i).*r_a(i);
end
beta_a=rxx_0+sum(s_a);

P_a=beta_a.*(abs(Aw_a).^2);
alpha_a=0.21;

figure(1)
plot(w_a-pi,log(1+alpha_a.*P_a));
xlabel('w');
ylabel('log(1+PSD_a)');

% b
r_b=r_xx(x,p_7+2);
rs_b=zeros(p_7,1);
for i=1:p_7
    rs_b(i)=r_b(i+2);
end
R_b=zeros(p_7,p_7);
for i=1:p_7
    for j=1:p_7
        k=abs(2+i-j);
        if k==0
            R_b(i,j)=rxx_0;
        else
            R_b(i,j)=r_b(k);
        end
    end
end
a_b=inv(R_b)*rs_b;

b=1;
Az_b=[1 a_b.'];
[Aw_b,w_b]=freqz(b,Az_b,'whole',Fs);
Aw_b=circshift(Aw_b,fix(Fs/2));

s_b=zeros(p_7,1);
for i=1:p_7
    s_b(i)=a_b(i).*r_b(i);
end
beta_b=rxx_0+sum(s_b);

P_b=beta_b.*(abs(Aw_b).^2);
alpha_b=0.16;

figure(2)
plot(w_b-pi,log(1+alpha_b.*P_b));
xlabel('w');
ylabel('log(1+PSD_b)');

% c
r_c=r_xx(x,2*p_7);
R_ns=zeros(2*p_7,p_7);
for i=1:2*p_7
    for j=1:p_7
        k=abs(i-j);
        if k==0
            R_ns(i,j)=rxx_0;
        else
            R_ns(i,j)=r_c(k);
        end
    end
end
R_s=R_ns.'*R_ns;
a_c=inv(R_s)*R_ns.'*r_c;

b=1;
Az_c=[1 a_c.'];
[Aw_c,w_c]=freqz(b,Az_c,'whole',Fs);
Aw_c=circshift(Aw_c,fix(Fs/2));
```

```
s_c=zeros(p_7,1);
for i=1:p_7
    s_c(i)=a_c(i).*r_c(i);
end
beta_c=rxx_0+sum(s_c);

P_c=beta_c.*(abs(Aw_c).^2);
alpha_c=1.8;

figure(3)
plot(w_c-pi,log(1+alpha_c.*P_c));
xlabel('w');
ylabel('log(1+PSD_c)');
```

Q 2

```
clc;
clear;

[x,Fs]=audioread('x.wav');
N=length(x);
X=abs(fftshift(fft(x)));
w=2*pi*(-N/2:N/2-1)/N;

figure(1)
plot(w,log(1+X.^2));
xlabel('w');
ylabel('log(1+|X(w)|^2)');
```