Linear Prediction.

↳ given a signal $x(n)$. we wish to predict future values of $x(n)$ from the previous samples. define a linear (F/R)

$$\hat{x}(n) = -\sum_{k=1}^{p} a_k x(n-k)$$

what we do is to find the coefficient a_k.

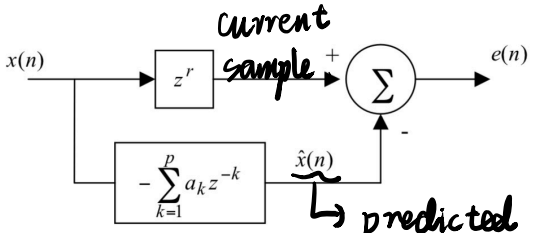
pth order prediction. there are $(p+1)$ coefficients, $a_0 = 1$.

a_k can be designed by minimising over the samples n.

the function to minimise can be written:

$$E = \sum_{n} (x(n+r) - \hat{x}(n))^2 \rightarrow \text{sum of error squared}$$

we use autocorrelation method to choose the range n.



for linear predictor.

↳ predicted signal.

$$x(z) \rightarrow \boxed{A(z)} \rightarrow E(z)$$

set $\frac{\partial E}{\partial a_k} = 0$.

input $x(n)$, get output prediction error function $e(n)$.

One of the most powerful models currently in use is where s_n is considered to be the output of some system with some unknown input u_n

$$s_n = -\sum_{k=1}^{p} a_k s_{n-k} + G \sum_{i=0}^{q} b_i u_{n-i} \quad b_0 = 1 \qquad (1)$$

a_k, $k \in [1, p]$, b_i, $i \in [1, q]$, and G are parameters,

↳ we can see that "output" s_n is a linear function of

past outputs. and present and past input.

↓↓

That is how to predict the signal S_n.

for equation 0. we can specified in frequency domain, by z transform. on both side.

$$H(z) = \frac{S(z)}{U(z)} = G \frac{1 + \sum_{l=1}^{q} b_l z^{-l}}{1 + \sum_{k=1}^{p} a_k z^{-k}}.$$

$$\begin{cases} \text{all-zero model}: & a_k = 0 \quad k \in [1, p] \\ \text{all-pole model}: & b_l = 0 \quad l \in [1, q] \end{cases}$$

what we will use is all-pole model:

$$S_n = -\sum_{k=1}^{p} a_k S_{n-k} + G \cdot u_n.$$

$$H(z) = \frac{G}{1 + \sum_{k=1}^{p} a_k z^{-k}}$$

* given a S_n, the problem is to find a_k and G. the derivations can be gotten by intuitive least squares approach:

assume input u_n is unknown, therefore. the signal S_n can be predicted only approximately from a linearly weighted summation of past samples; the approximation of S_n is $\tilde{S}_n$:

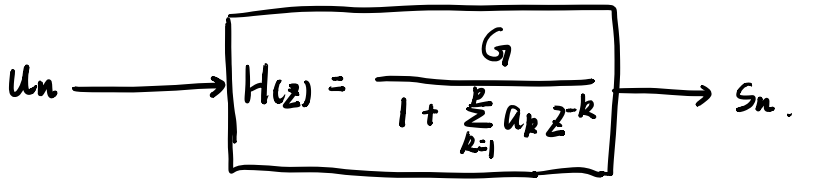
$$\tilde{S}_n = -\sum_{k=1}^{p} a_k \cdot S_{n-k}.$$

for the error:

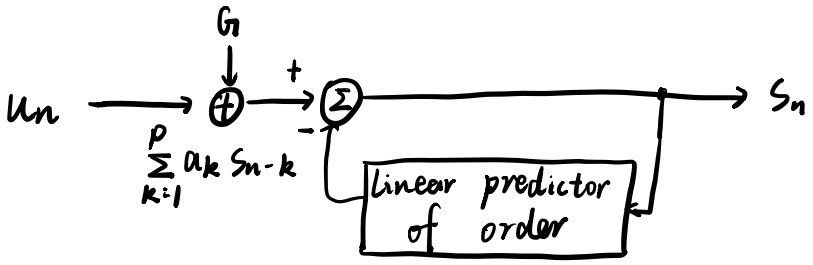
$$e_n = s_n - \tilde{s}_n$$
$$= s_n + \sum_{k=1}^{p} a_k s_{n-k}.$$

e_n is residual, a_k are obtained as a result of the minimization of the mean or total squared error with respect to each parameters

for frequency domain:



for time domain



the analysis is assumed that s_n is a deterministic signal and then give analogous derivations assuming that s_n is sample from a random process

0. deterministic signal

denote the total squared error by E:

$$E = \sum_n e_n^2 = \sum_n \left(s_n + \sum_{k=1}^{p} a_k s_{n-k} \right)^2$$

so. E is minimized by setting

$$\frac{\partial E}{\partial a_i} = 0. \qquad i \in [1, p]$$

from the above two equation, we can get:

$$\sum_{k=1}^{p} a_k \sum_n s_{n-k} s_{n-i} = - \sum_n s_n \cdot s_{n-i}.$$

for any definition of the signal s_n, the above

equation forms a set of p equations in p unknowns which can be solved for predictor coefficients $\{a_k, 1 \le k \le p\}$, which minimize E.

the minimum total squared error

$$E = \sum_n s_n^2 + \sum_{k=1}^{p} a_k \sum_n s_n s_{n-k}$$

now, we need to find n.

there are two method:

- Autocorrelation method
- Covariance method

in this experiment, we use the first method.

Autocorrelation method:

for error:

$$E = \sum_n e_n^2 = \sum_n \left(s_n + \sum_{k=1}^{p} a_k s_{n-k} \right)^2$$

assume the error is minimized over $-\infty < n < \infty$.

let $R(i) = \sum_{n=-\infty}^{\infty} s_n \cdot s_{n+i}$

so, $\sum_{k=1}^{p} a_k R(i-k) = -R(i) \quad 1 \le i \le p.$

$$E_p = R(0) + \sum_{k=1}^{p} a_k R(k)$$

☆ $R(i) = \sum_{n=-\infty}^{\infty} s_n s_{n+i}$. is just the autocorrelation function of signal s_n.

in practice, s_n is finite. we can use a window function w_n to multiply the s_n.

$$s_n' = s_n \cdot w_n \qquad 0 \le n \le N-1$$

$$\vdots \qquad \vdots \qquad \vdots \qquad \vdots$$

$$R(i) = \begin{cases} \sum_{n=0}^{N-1-i} s_n' s_{n+1}' & i \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

Gain computation:

we assume the input was unknown.

$$\therefore \quad e_n = s_n - \tilde{s}_n = s_n + \sum_{k=1}^{p} a_k s_{n-k}$$

$$\therefore \quad s_n = -\sum_{k=1}^{p} a_k s_{n-k} + e_n$$

Hence. $G_{RUN} = e_n$

which means. input signal is proportional to the error signal

if we want the output signal energy be equal to the original signal energy.

↳ $G_{RUN} = e_n$.

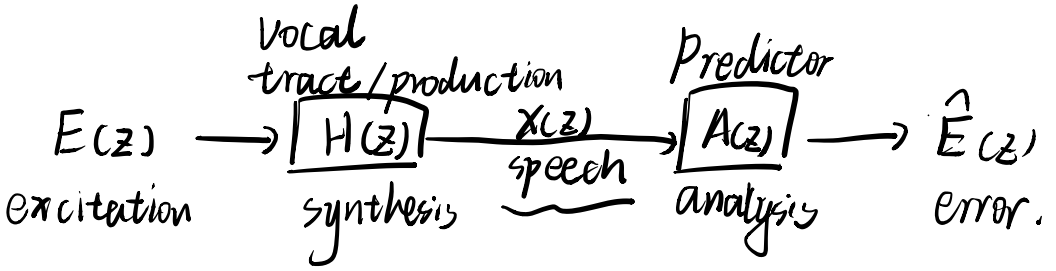
Computation of predictor Parameters.

the predictor coefficients a_k, $1 \leq k \leq p$ can be computed by solving a set of p equations with p unknown.

And. when we compute. we can use matrix; for autocorrelation normal equation:

$$\begin{bmatrix} R_0 & R_1 & R_2 & \cdots & R_{p-1} \\ R_1 & R_0 & R_1 & \cdots & R_{p-2} \\ R_2 & R_1 & R_0 & \cdots & R_{p-3} \\ \vdots & \vdots & \vdots & & \vdots \\ R_{p-1} & R_{p-2} & R_{p-3} & \cdots & R_0 \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = - \begin{bmatrix} R_1 \\ R_2 \\ R_3 \\ \vdots \\ R_p \end{bmatrix}$$

toeplitz



$E(z)$, $\hat{E}(z)$ will be identical. when $H(z)$ and $A(z)$ matched.

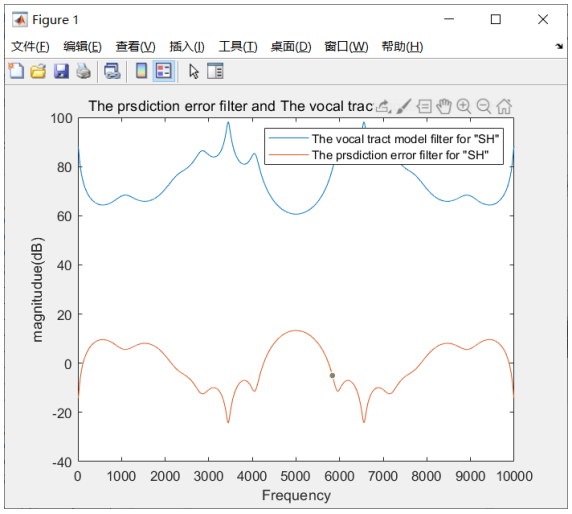
$$H(z) = \frac{G}{A(z)} \Rightarrow \text{error signal is selected.}$$

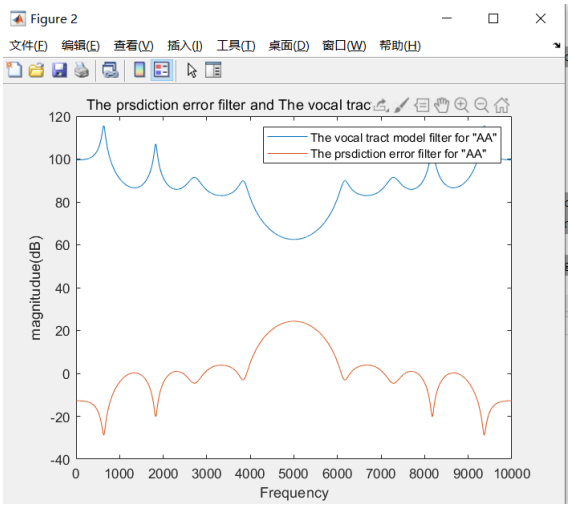
$$\hat{E}(z) = G\,E(z)$$

we can see that prediction error and vocal tract model are two totally different process. the results should be exactly inverse.

Result:

Exercise 2:



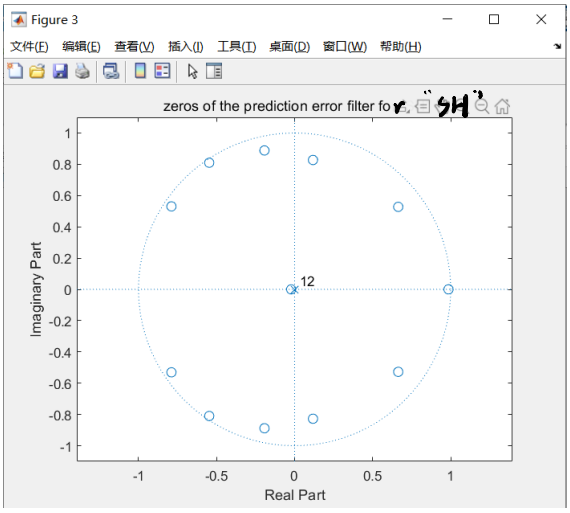


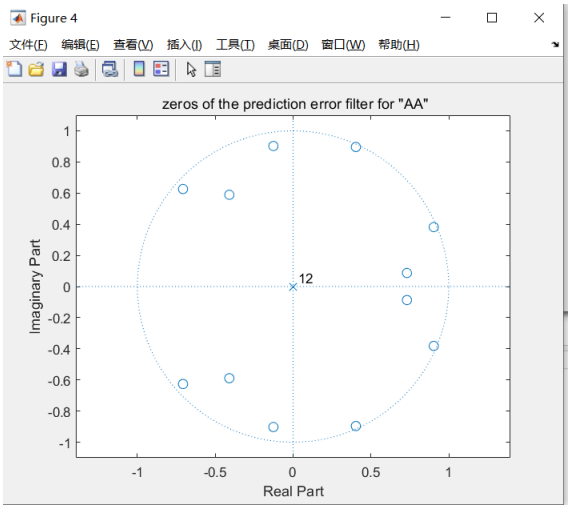
we can see that the figure of vocal tract model filter and the prediction error filter is exactly symmetrical.

because in the LPC processing, we can think the vocal tract model filter is just the inverse process of prediction error

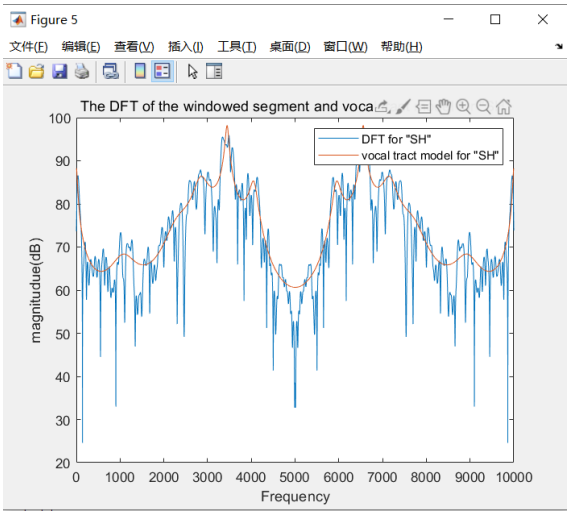
for vocal tract model filter $H = \frac{G}{A}$

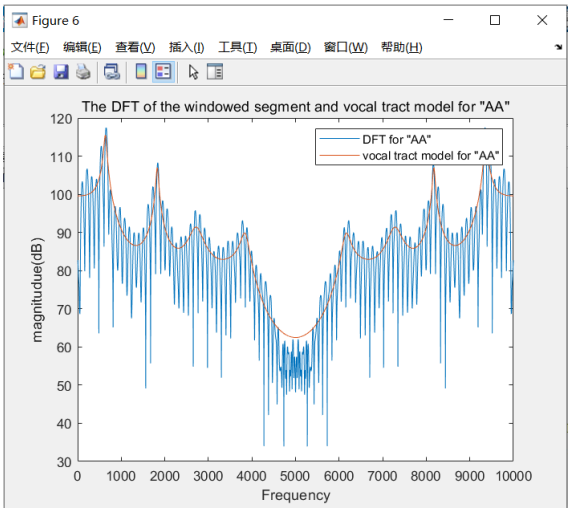
we can see that $H \propto \frac{1}{A}$





Exercise 3:

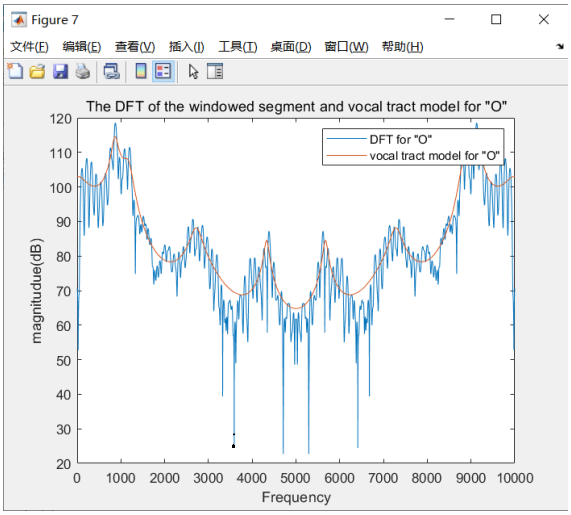


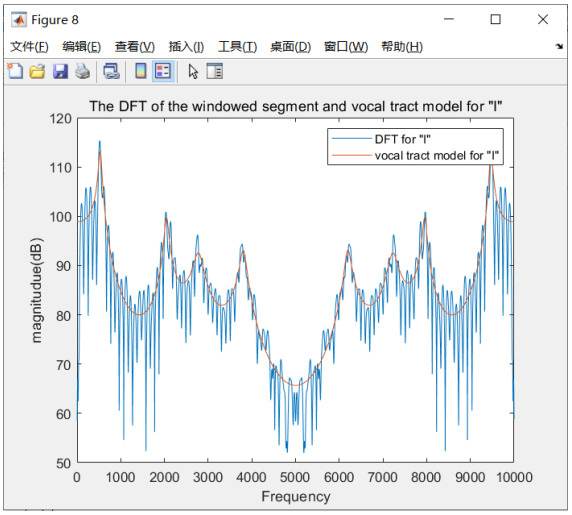


we can see that for both "SH" and "AA" the DFT of the windowed segment and the vocal tract model are very fitting.

because the building of vocal tract model is just on the base of fft.

Exercise 4:

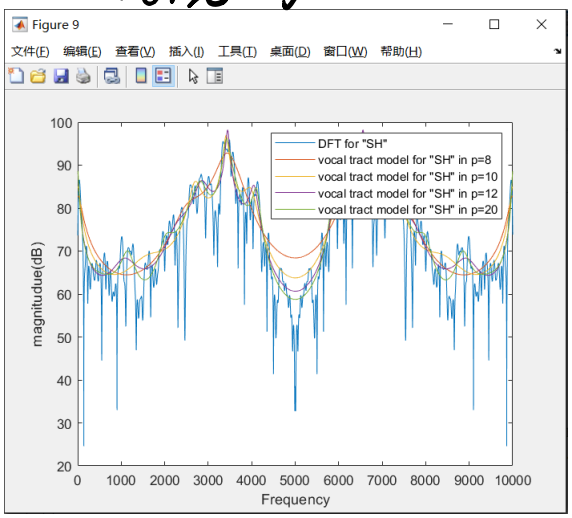




for "O" we select from 7200 to 7520

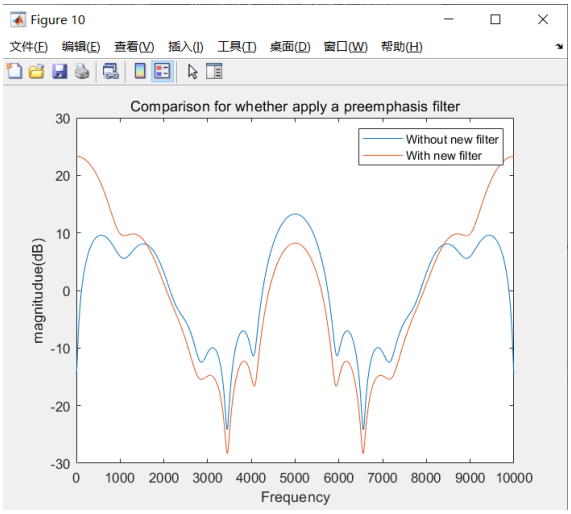
for "I" we select from 14500 to 14820
and the figures are just like the exercise 3.
the DFT and the vocal tract model filter are fitting

Exercise 5:



we take "SH" as example.
we can see that. with the increasing of order - p.
the vocal tract model filter are more and more close to the DFT of the windowed signal.
because. with the higher order. after the filter, the signal has more similarity to the original signals.

Exercise 6.



the filter used is Gaussian filter.

we can see that. with a filter.
the signal is smoother.
because the filter filtered out some signals. as we set [1, -0.98]
which means we reduce the variance of the signal

Exercise 1: Autolpc function:

```
function [A, G, r, a] = autolpc(x, p)
%AUTOLPC Autocorrelation Method for LPC
% Usage: [A, G, r, a] = autolpc(x, p)
% x : input samples
% p : order of LPC
% A : prediction error filter, (A = [1; -a])
% G : rms prediction error
% r : autocorrelation coefficients
% a : predictor coefficients

x = x(:);
L = length(x);
r = zeros(p+1,1);
for i=0:p
   r(i+1) = x(1:L-i)' * x(1+i:L);
end
R = toeplitz(r(1:p));
a = R\r(2:p+1);
A = [1; -a];
G = sqrt(sum(A.*r));
end
```

Exercise 2~6:

```
% Exercise 2
load('s5.mat');
L=320;                    % Hamming window of 320 samples
win= hamming(L);
SH=s5(15800:15800+319);   % select the suit length array for window
AA=s5(17000:17000+319);
SH_win=SH.*win;
AA_win=AA.*win;
[A1,G1,r1,a1]=autolpc(SH_win,12);
[A2,G2,r2,a2]=autolpc(AA_win,12);

SH_freq=freqz(A1,1,10000,'whole');        % use the function freqz to get the frequency
 responses
SH_vtm_filter=freqz(G1,A1,10000,'whole'); % A is prediction error filter, G/A is the vo
cal
figure(1);
plot(20*log10(abs(SH_vtm_filter)));       % get the log magnitude
hold on;
plot(20*log10(abs(SH_freq)));
title('The prsdiction error filter and The vocal tract model filter for"SH"');
xlabel('Frequency');
ylabel('magnitudue(dB)');
legend('The vocal tract model filter for "SH"','The prsdiction error filter for "SH"')

AA_freq=freqz(A2,1,10000,'whole');
AA_vtm_filter=freqz(G2,A2,10000,'whole');
figure(2);
plot(20*log10(abs(AA_vtm_filter)));
hold on;
```

```
plot(20*log10(abs(AA_freq)));
title('The prsdiction error filter and The vocal tract model filter for"AA"');
xlabel('Frequency');
ylabel('magnitudue(dB)');
legend('The vocal tract model filter for "AA"','The prsdiction error filter for "AA"')

figure(3)       % find the zeros of the prediction error filter for both cases
zplane(A1');
title(' zeros of the prediction error filter for "SH"');
figure(4)
zplane(A2')
title(' zeros of the prediction error filter for "AA"');

% Exercise 3
figure(5)                   % get DFT for SH
SH_DFT=fft(SH_win,10000);
plot(20*log10(abs(SH_DFT)));
hold on;
plot(20*log10(abs(SH_vtm_filter)));
title('The DFT of the windowed segment and vocal tract model for "SH"');
xlabel('Frequency');
ylabel('magnitudue(dB)');
legend('DFT for "SH"','vocal tract model for "SH"');

figure(6)                  % get DFT for AA
AA_DFT=fft(AA_win,10000);
plot(20*log10(abs(AA_DFT)));
hold on;
plot(20*log10(abs(AA_vtm_filter)));
title('The DFT of the windowed segment and vocal tract model for "AA"');
xlabel('Frequency');
ylabel('magnitudue(dB)');
legend('DFT for "AA"','vocal tract model for "AA"');


% Exercise 4
OO=s5(7200:7200+319);
OO_win=OO.*win;
OO_DFT=fft(OO_win,10000);
[A3,G3,r3,a3]=autolpc(OO_win,12);
OO_vtm_filter=freqz(G3,A3,10000,'whole');
figure(7)
plot(20*log10(abs(OO_DFT)));
hold on;
plot(20*log10(abs(OO_vtm_filter)));
title('The DFT of the windowed segment and vocal tract model for "O"');
xlabel('Frequency');
ylabel('magnitudue(dB)');
legend('DFT for "O"','vocal tract model for "O"');

II=s5(14500:14500+319);
II_win=II.*win;
II_DFT=fft(II_win,10000);
[A4,G4,r4,a4]=autolpc(II_win,12);
II_vtm_filter=freqz(G4,A4,10000,'whole');
figure(8)
plot(20*log10(abs(II_DFT)));
hold on;
plot(20*log10(abs(II_vtm_filter)));
title('The DFT of the windowed segment and vocal tract model for "I"');
```

```
xlabel('Frequency');
ylabel('magnitudue(dB)');
legend('DFT for "I"','vocal tract model for "I"');

% Exercise 5
% We choose SH in this part, [A1,G1,r1,a1]=autolpc(SH_win,12);
[A_p8,G_p8,r_p8,a_p8]=autolpc(SH_win,8);
SH_vtm_filter_p8=freqz(G_p8,A_p8,10000,'whole');
[A_p10,G_p10,r_p10,a_p10]=autolpc(SH_win,10);
SH_vtm_filter_p10=freqz(G_p10,A_p10,10000,'whole');
[A_p20,G_p20,r_p29,a_p20]=autolpc(SH_win,20);
SH_vtm_filter_p20=freqz(G_p20,A_p20,10000,'whole');
        figure(9)
        plot(20*log10(abs(SH_DFT)));
        hold on;
        plot(20*log10(abs(SH_vtm_filter_p8)));
        hold on;
        plot(20*log10(abs(SH_vtm_filter_p10)));
        hold on;
        plot(20*log10(abs(SH_vtm_filter)));
        hold on;
        plot(20*log10(abs(SH_vtm_filter_p20)));
        xlabel('Frequency');
        ylabel('magnitudue(dB)');
        legend('DFT for "SH"','vocal tract model for "SH" in p=8', ...
            'vocal tract model for "SH" in p=10', ...
            'vocal tract model for "SH" in p=12', ...
            'vocal tract model for "SH" in p=20');

        % Exercise 6
        y = filter([1, -0.98], 1, s5);
        SH_y=y(15800:15800+319);
        SH_y_win=SH_y.*win;
        [A_y,G_y,r_y,a_y]=autolpc(SH_y_win,12);
        SH_y_freq=freqz(A_y,1,10000,'whole');
        figure(10)
        plot(20*log10(abs(SH_freq)));
        hold on;
        plot(20*log10(abs(SH_y_freq)));
        title('Comparison for whether apply a preemphasis filter');
        xlabel('Frequency');
        ylabel('magnitudue(dB)');
        legend('Without new filter','With new filter');
```