EE635: Digital Signal Processing II
Computer Assignment-1

University of New Haven
Instructor: Dr. Alain Barthelemy
Student: Wiwat Tharateeraparb 888-80-6557

Table of Contents

List of Figures

Chapter 1: Introduction

This project explores autoregressive (AR) modeling, its associated correlation functions and the Levinson-Durbin algorithm. For this work, we use two types of correlation sequence estimates from an AR process:

(i) Asymptotic estimate obtained from the Gohberg-Semencule relation [1].
(ii) Block estimate [2, chapter 2] obtained from simulated data.

Then we obtain the plots of poles and their frequency responses. In the analytic aspects, a given correlation sequence can be used to estimate AR coefficients in order to form an AR model. To solve for AR coefficients, we perform and compare two types of method:

(i) Levinson-Durbin algorithm [2],[3].
(ii) Solving of Yule-Walker equation [2, chap.2] by Guassian elimination method.

Form both algorithm, we observe the number of operations via MATLAB *flops()* function (floating point operation count) and compare the computational efficiency. We also use the result from Levinson-Durbin algorithm to model an AR process by selecting the AR model order via the AIC or MDL criterion, then compare their frequency responses to the designed responses.

Chapter 2: Autoregressive (AR) Models

We represent the AR model as a linear combination of the time series as

$$u(n)+a_1^* u(n-1)+\ldots +a_M^* u(n-M) = v(n) \text{ ,[2, eq.(2.42)]} \quad (1)$$

where $a_1, a_2, \ldots, a_M$ are the AR coefficients or AR parameters, v(n) is the input white-noise process, and u(n) is the output of the AR process or the desired signal. This AR model is said to be an all-pole filter because its transfer function of $H(z) = U(z)/V(z)$ yields

$$H(z) = 1/(1+ a_1^* z^{-1}+ a_2^* z^{-2}+\ldots+ a_M^* z^{-M}) \quad (2)$$

$$= 1/(1- p_1 z^{-1})(1- p_2 z^{-1})\ldots(1- p_M z^{-1}) \quad (3)$$

The parameter $p_1, p_2, \ldots, p_M$ are poles of $H(z)$. For the all-pole AR process to be stable, all poles must lie inside the unit circle in the z-plane. Moreover, in the wide-sense of AR stationary process, all poles must satisfy the condition of asymptotic stationarity, say, $|p_k| < 1$ for all k, in which case the autocorrelation sequence r(k) approaches zero as the lag k approaches infinity.

2.1 AR Process

In this project, we will explore the AR process through the asymptotic and simulate data with given AR process parameters. First, we use a white-noise random signal as the input, $v(n)$, of this AR process (1). In this case, we generate the unit-variance, zero-mean random signal then put this through the AR process (1), in which has the characteristic as follows:

(a) Second order AR process with the coefficients $a_R = [-0.7, 0.3]$. (4)
(b) Second order AR process with the coefficients $a_C = [-0.25j, 0.5+0.3j]$. (5)

Note that in AR process, a_0 is equal to 1. All processes of generating the simulated data, $u(n)$, from AR process is shown on the MATLAB source file (A.1). To ensure that these second order AR processes are stable, we illustrate their poles plot in z-plane and the frequency responses in the figure1 to figure4. It seems that the AR process with all real coefficients has the characteristic of the *lowpass* FIR filter where its poles locate within the unit circle. While the complex-coefficients AR process yields the *bandpass* FIR filter.

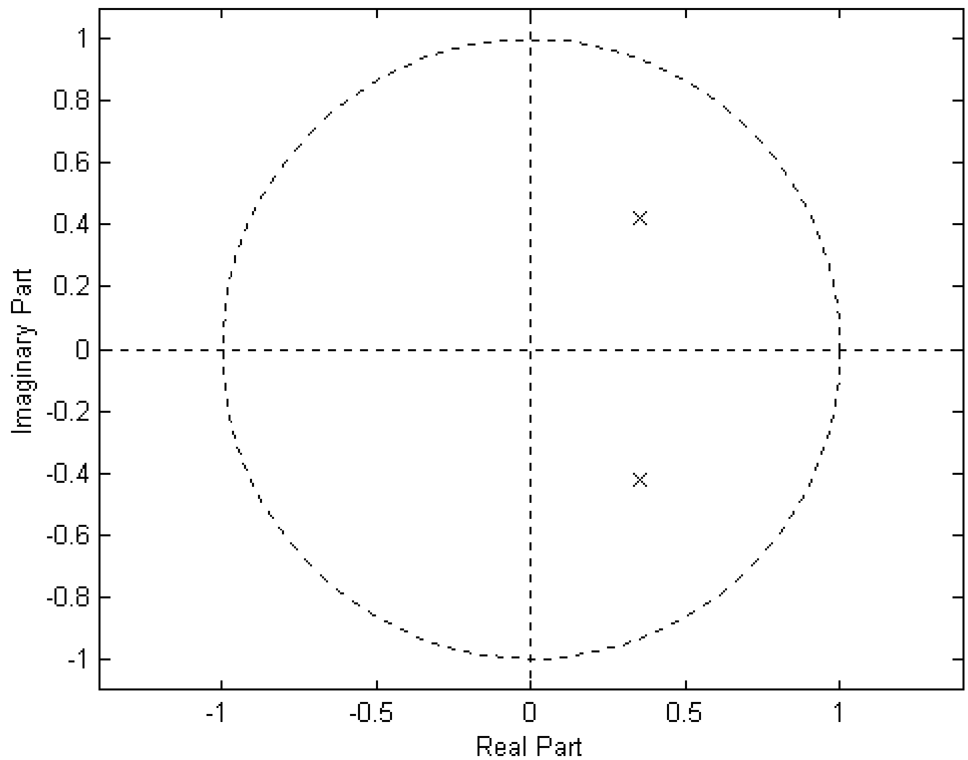

Figure 1. Poles of the real-coefficients AR model

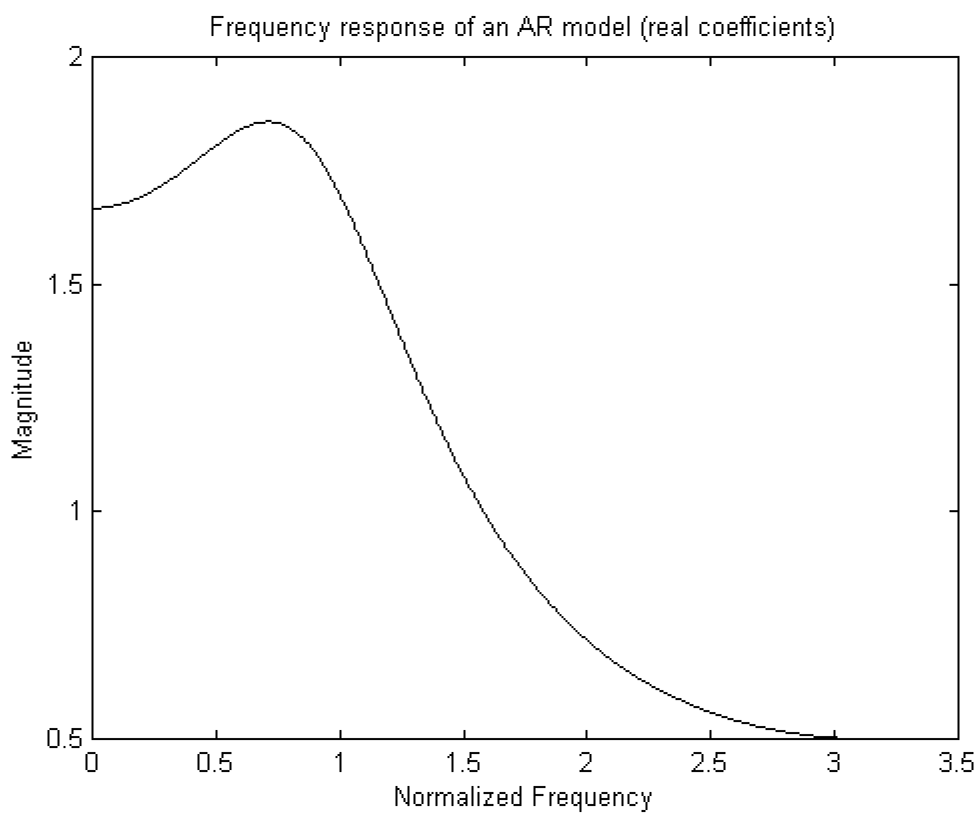

Figure 2. Frequency response of the real-coefficients AR model

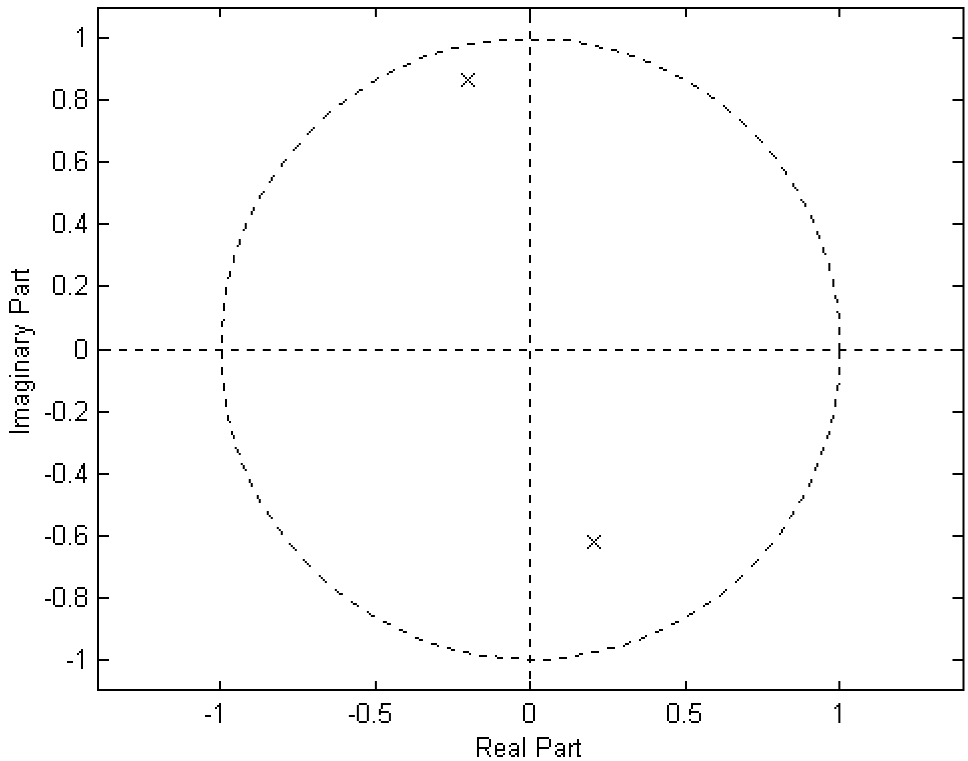

Figure 3. Poles of the complex-coefficients AR model

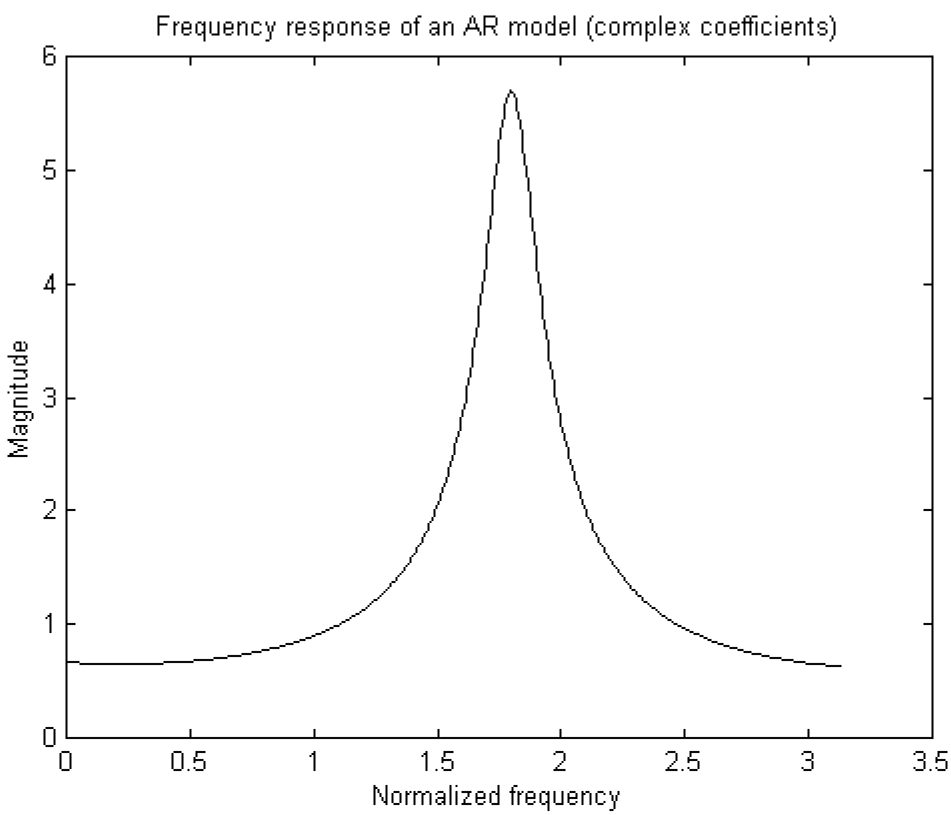

Figure 4. Frequency response of the complex-coefficients AR model

2.2 Autocorrelation Matrix

One way to define or express the AR process uniquely, in which we need to know the AR coefficients as well as the variance of the white noise used as excitation, is the second order statistic, namely, autocorrelation function which is:

$$\boldsymbol{R} = E[\boldsymbol{u}(n)\boldsymbol{u}^H(n)] \tag{6}$$

The matrix $\boldsymbol{R}$ is the autocorrelation of the signal $\boldsymbol{u}(n)$ and H denotes the *Hermitian transposition*. If we let $\boldsymbol{u}(n)$ = [u(n), u(n-1),…,u(n-(M-1))]^T , the matrix **R** may be written in the form:

$$\boldsymbol{R} = \begin{vmatrix} r(0) & r(1) & \dots & r(M-1) \\ r^*(1) & r(0) & \dots & r(M-2) \\ . & . & . & . \\ . & . & . & . \\ r^*(M-1) & r^*(M-2) & \dots & r(0) \end{vmatrix} \tag{7}$$

where r(k) = E[u(n)u*(n-k)] denotes the correlation sequence with lag k.

In order to obtain the autocorrelation matrix, we use two types of method:

(a) Asymptotic estimate autocorrelation derived from the Gohberg-Semencule relation [1] that represents the infinite number of samples used to obtain the autocorrelation.
(b) Block-estimate method [2, eq.6.63]

2.2.1 Results from Gohberg-Semencule Relation

The Gohberg-Semencule relation [1] is the method used to obtain the asymptotic correlation matrix. The result of the Gohberg-Semencule is the inverse of the correlation matrix of an autoregressive (AR) with only the coefficients of an AR process and the white-noise variance used as the excitation are given. It is:

$$\boldsymbol{R}^{-1} = (1/\sigma)[\boldsymbol{L}_1 \boldsymbol{L}_1^T - \boldsymbol{L}_2 \boldsymbol{L}_2^T] \tag{8}$$

Where σ is the variance of the white-noise used as an input of AR process and $\boldsymbol{L}_1, \boldsymbol{L}_2$ are defined as the lower triangular Toeplitz matrices:

$$\boldsymbol{L}_1 = \begin{vmatrix} 1 & & & & \bigcirc \\ -a_1 & 1 & & & \\ -a_2 & -a_1 & 1 & & \\ . & . & . & & \\ . & . & . & . & \\ -a_N & -a_{N-1} & \dots & & 1 \end{vmatrix}, \quad \boldsymbol{L}_2 = \begin{vmatrix} 0 & & & & \bigcirc \\ -a_N & 0 & & & \\ -a_{N-1} & -a_N & 0 & & \\ . & . & . & & \\ . & . & . & . & \\ -a_1 & -a_2 & & \dots & 0 \end{vmatrix}$$

where N is a length of the filter or an order of an AR process and a_N denotes a coefficient of AR process.

Let N = 24 is the dimension of the problem or concerns the correlation sequence within the lag of 23 and the result from Gohberg-Semencule would be a 24x24 matrix. Let $\mathbf{R}_1$, $\mathbf{R}_2$ (24x24 matrices) are the inverse of the results of (8) with the AR coefficients in (4) and (5) respectively. We use MATLAB to compute these correlation matrices (see appendix A.1) and plot the correlation sequences r(k). Figure5 contains the plot of correlation sequence r(k) which is the first column or row of the correlation matrix $\mathbf{R}_1$ (dash line). In the figure6, r(k) of $\mathbf{R}_2$ is shown as well. As the lag k of autocorrelation function r(k) increases, the magnitude approaches zero.

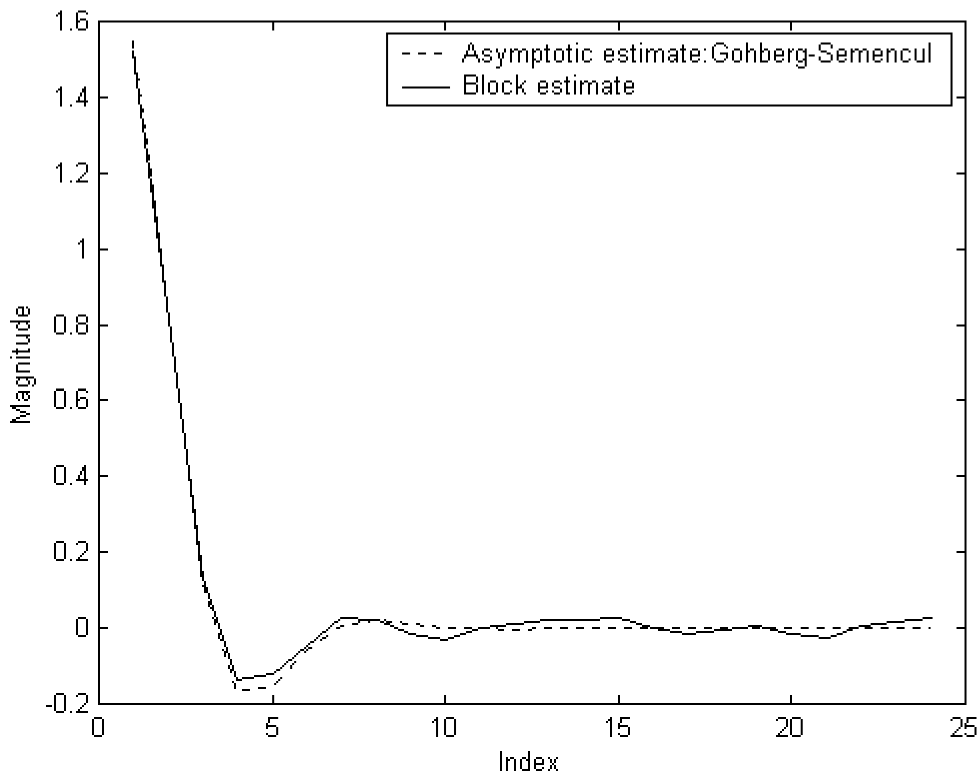

Figure 5. Correlation sequence r(k) of real-coefficients AR process via Gohberg-Semencule relation and Block estimate method

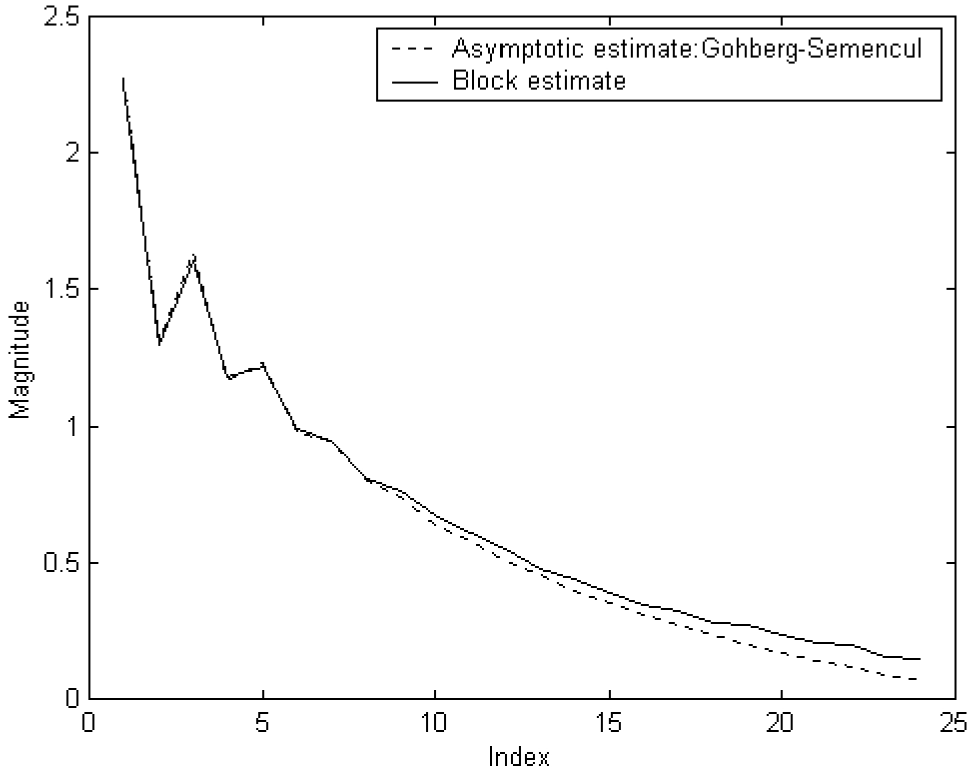

Figure 6. Correlation sequence of complex-coefficients AR process via the Gohberg-Semencule relation and Block estimate method

2.2.2 Results from Block-estimate Method

We use the Block-estimate method [2, eq.6.63] to obtain the correlation matrix if the raw data is given or in order to analyze a signal, estimate and model an AR model in practice. For this project, we use the simulated data from (1) to form a correlation matrix via the Block estimate equation:

$$r(k) = (1/M)(\Sigma_{k=0,N-1}\mathbf{u}(n)\mathbf{u}^H(n-k)) \quad (9)$$

where M is the total length of the input time series and N is the number of the problem or the values of autocorrelation function for lags 0, 1, .., N-1. u(n) is the input data vector of length M. In particular, we have r(0), r(1), …, r(N-1) to form an estimated autocorrelation matrix **R** in (7).

Let M = 10000, N = 24 and **u**(n) is given from (1) . We generate this data on MATLAB (see appendix A.1) and let $\mathbf{R}_3$, $\mathbf{R}_4$ (24x24 matrices) are the results of Block estimate with the AR coefficients in (4) and (5) respectively. Figure5 contains the plot of correlation sequence r(k) which is the first column or row of the correlation matrix $\mathbf{R}_3$ (solid line). In the figure6, r(k) of $\mathbf{R}_4$ is shown as well. As the lag k of autocorrelation function r(k) increases, the magnitude approaches zero.

Obviously, the correlation sequence from asymptotic estimate is smoother than Block-estimate method. The more time samples used in the Block estimate, the closer result (correlation function) to the asymptotic estimate.

Chapter 3: AR Coefficients Estimation

3.1 Levinson-Durbin Algorithm

We have found that the property of a forward prediction-error filter [2, chap.6], operating on a stationary discrete-time stochastic process, is intimately relate to the *autoregressive (AR) modeling* of the process we have done so far. The prediction-error filter is an *all-zero filter* with an impulse response of finite duration. On the other hand, the inverse of prediction-error yields the AR model which is an all-zero filter with an impulse response of infinite duration. From this relation, we adopt the Levinson-Durbin algorithm [2, pp.913] to compute the estimate AR coefficients which is:

Initialize the algorithm by setting

$$a_{0,0} = 1$$
$$P_0 = r(0)$$

Hence, compute for m = 1, 2, 3,…, M:

$$K_m = -(1/P_{m-1})\Sigma_{i=1,m-1} r(i-m)\, a_{m-1,i}$$

$$a_{m,i} = \begin{cases} 1 & \text{for } i = 1 \\ a_{m-1,i} + K_m\, a^*_{m-1,i-1} & \text{for } i = 1,2,..,m-1 \\ K_m & \text{for } i = m \end{cases}$$

$$P_m = P_{m-1}(1 - |K_m|^2)$$

where $a_{M,k}$, k = 1, 2,.., M denotes the estimated AR coefficients
P_m is the prediction error power and $P_M = \sigma^2{}_v$
K_m is the reflection coefficient

From the project, we have the two types of correlation matrix, namely, $\mathbf{R}_1$, $\mathbf{R}_2$ that is generated from asymptotic estimate (Gohberg-Semencule) and $\mathbf{R}_3$, $\mathbf{R}_4$ generated from Block estimate. By applying the Levinson-Durbin algorithm to the correlation sequence $r^*(0)$, $r^*(1)$, ..., $r^*(23)$ of each matrix (fig.5-6), we can estimate the AR models of the given data. We implement the Levinson-Durbin algorithm in MATLAB as in Appendix A.3 and the results are:

(a) From $\mathbf{R}_1$,the estimated AR coefficients are $a\hat{}_R$=[1.0000 -0.7000 0.3000
-0.0000 0.0000 -0.0000 0.0000 0.0000 -0.0000 0.0000 0.0000
-0.0000 0.0000 0.0000 0.0000 -0.0000 0.0000 0.0000 -0.0000
0.0000 0.0000 -0.0000 0.0000 0.0000]

(b) From $\mathbf{R}_2$,the estimated AR coefficients are $a\hat{}_C$=[1.0000 0.0002 - 0.2505i

0.4997 + 0.2999i   0.0003 - 0.0003i   -0.0000 - 0.0000i -0.0000+ 0.0000i
-0.0000 + 0.0000i -0.0000+ 0.0000i   0.0000 - 0.0000i -0.0000 - 0.0000i
0.0000 + 0.0000i -0.0000 - 0.0000i   0.0000 - 0.0000i -0.0000 - 0.0000i
0.0000 + 0.0000i -0.0000 - 0.0000i   0.0000 - 0.0000i -0.0000 - 0.0000i
0.0000 + 0.0000i -0.0000 - 0.0000i   0.0000 - 0.0000i -0.0000 - 0.0000i
-0.0070 + 0.0042i -0.0079+ 0.0024i]

(c) From $\mathbf{R}_3$,the estimated AR coefficients are $\hat{a}_R$=[1.0000   -0.7131   0.3227
-0.0224   0.0151   -0.0040   -0.0104   0.0072   0.0128   -0.0323   0.0072
-0.0104   0.0202   -0.0030   -0.0286   0.0268   -0.0171   0.0259   -0.0462
0.0571   -0.0393   0.0240   -0.0104   -0.0004]

(d) From $\mathbf{R}_4$,the estimated AR coefficients are $\hat{a}_C$=[1.0000   -0.0233 - 0.2521i
0.5004 + 0.3099i -0.0335 - 0.0085i -0.0067 - 0.0189i -0.0357 - 0.0138i
0.0075 - 0.0325i -0.0297 - 0.0040i   0.0004 - 0.0022i -0.0464+ 0.0022i
-0.0189 + 0.0156i -0.0287 - 0.0095i   0.0052+ 0.0098i   0.0057 - 0.0150i
0.0101 + 0.0045i   0.0005 + 0.0135i   0.0083+ 0.0111i   0.0006 - 0.0112i
-0.0099 + 0.0142i   0.0599 - 0.0318i -0.0195+ 0.0074i   0.0521 + 0.0011i
-0.0260 + 0.0054i   0.0113 + 0.0150i]

The asymptotic estimate yields the closest to (4) and (5) while the Block estimate yields close estimate of AR coefficients. In analysis, we estimate the parameters of the AR model by processing a given time series of finite length. We need an appropriate measure of the fit between the model and observed data. An Information-Theoretic criterion [2, pp 128] such as AIC and MDL criterion is the measure of selecting the model order. It is implemented in MATLAB (see Appendix A.1) and the results are graphed in figure7 and figure8 which display the *optimum order* of the model at third order. By AIC and MDL criterion results, we select the AR model of *three* (3rd order AR process) to characterize the observed data (simulated data) in Chapter 2. Then we investigate the poles plots and impulse responses of the estimated AR models in figure9 to figure12. Both estimated real-coefficients and complex-coefficients AR model perform the close responses in figure1 to figure4 satisfactorily.

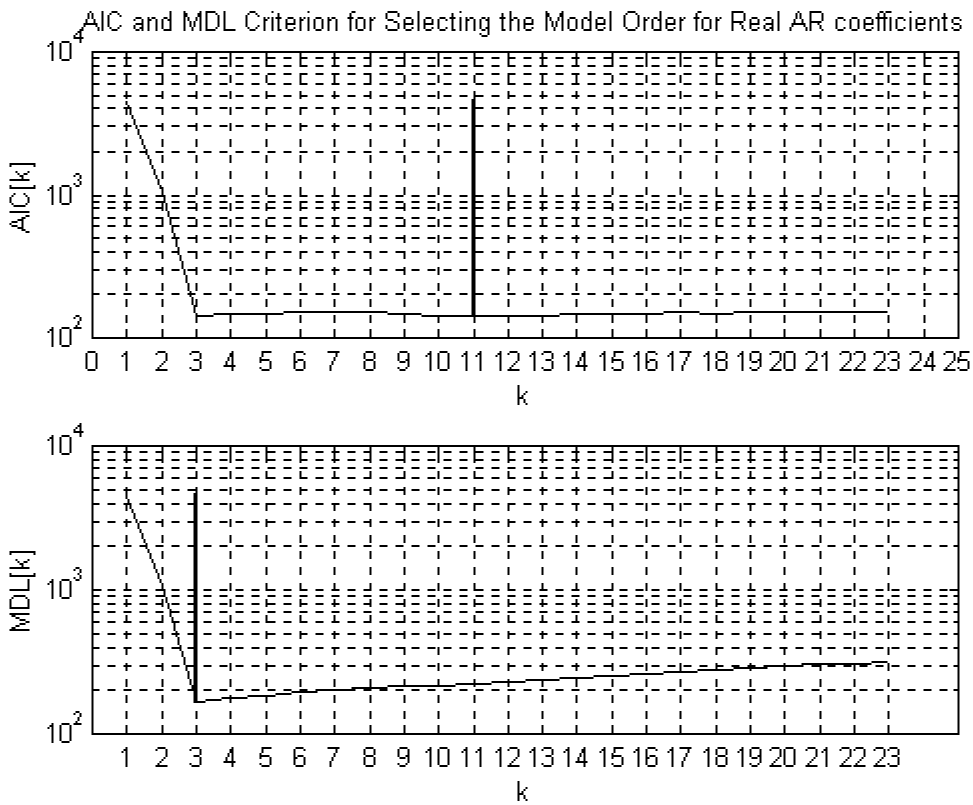

Figure 7. AIC and MDL criterion for selecting the AR model order

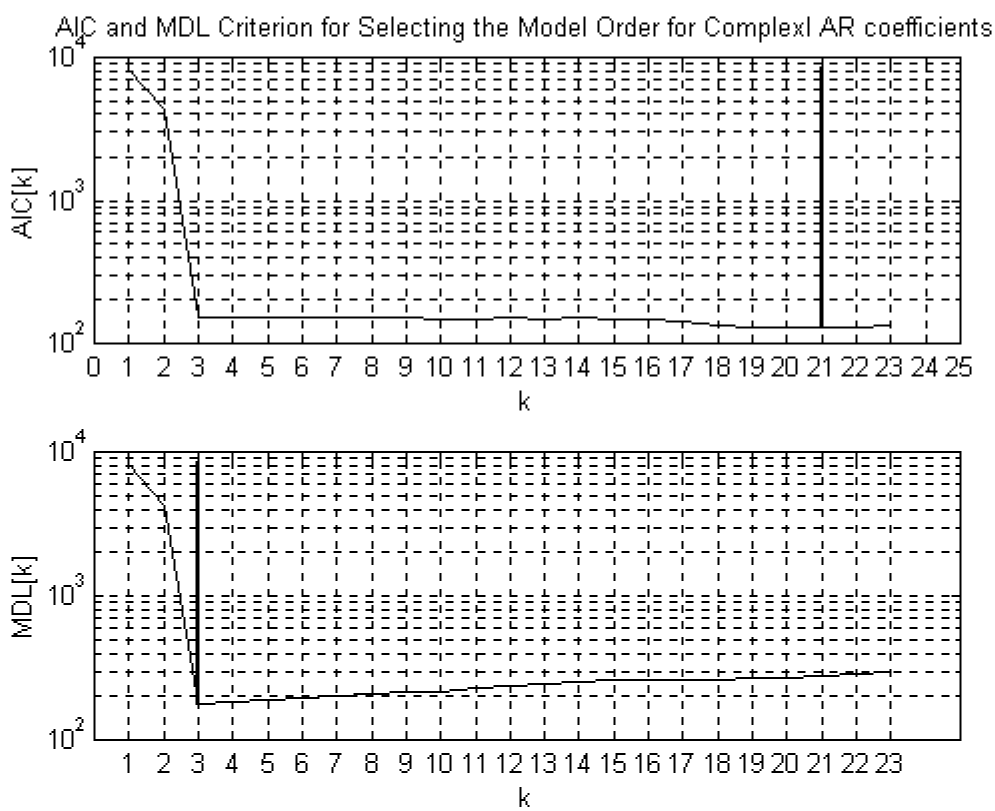

Figure 8. AIC and MDL criterion for selecting the AR model order

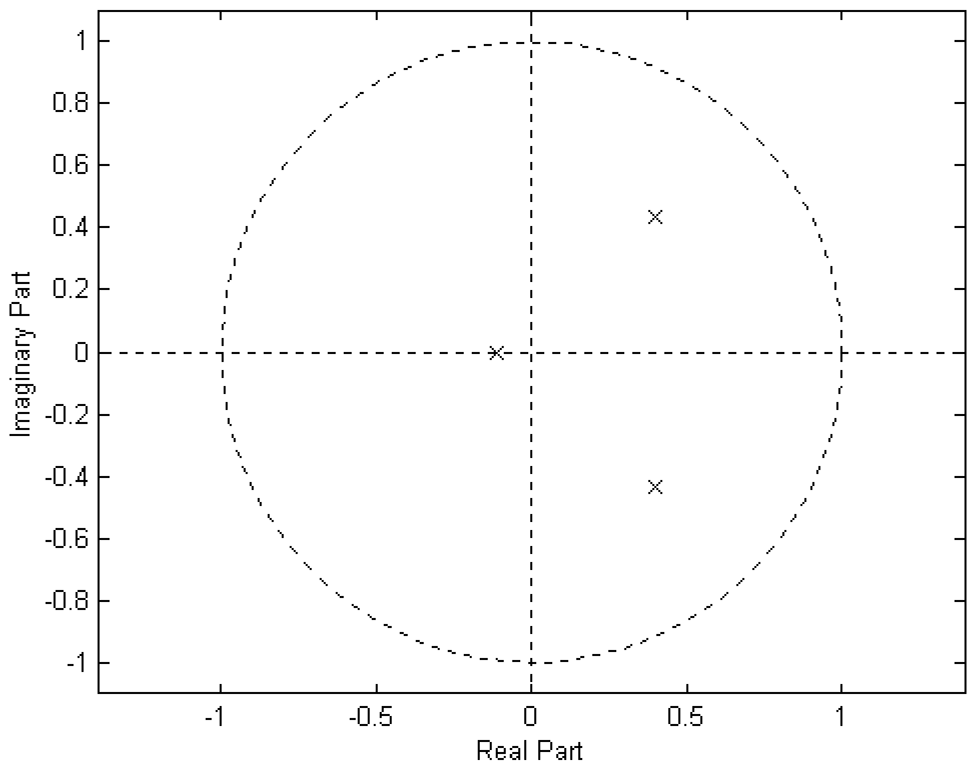

Figure 9. Poles of the estimated real-coefficients AR model

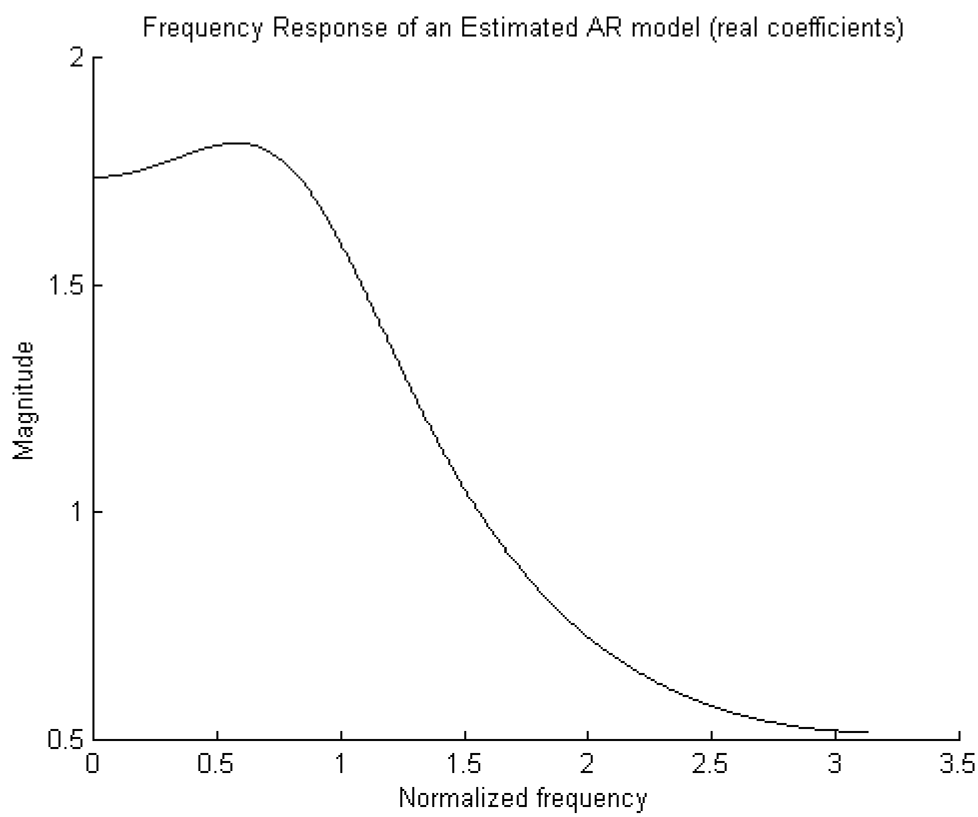

Figure 10. Frequency response of the estimated real-coefficients AR model

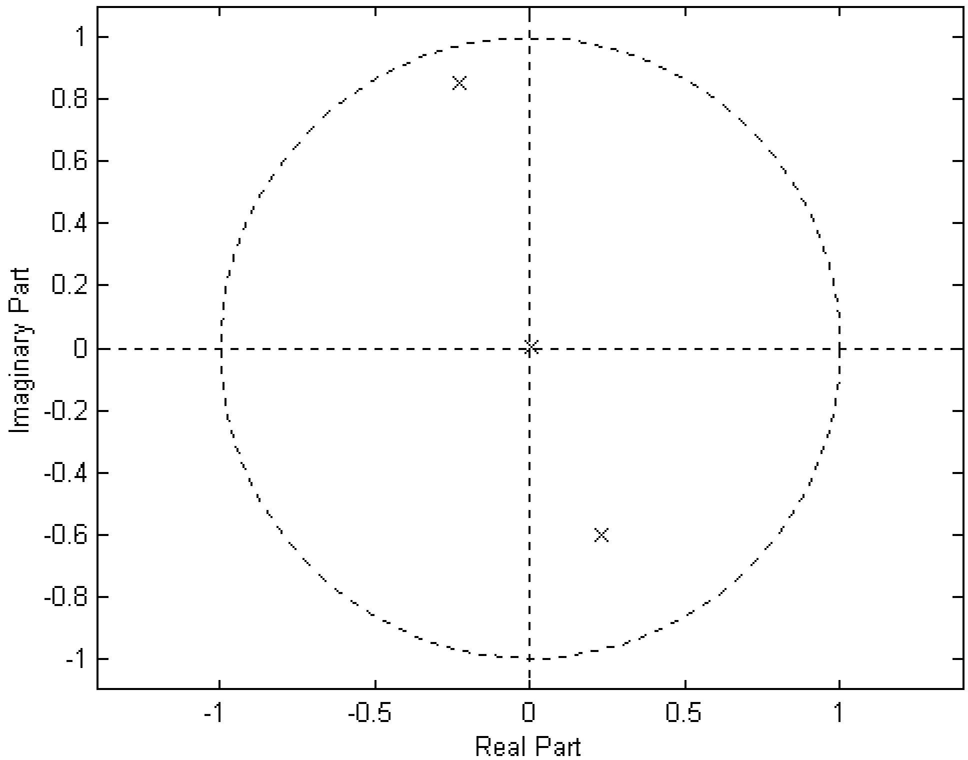

Figure 11. Poles of the estimated complex-coefficients AR model

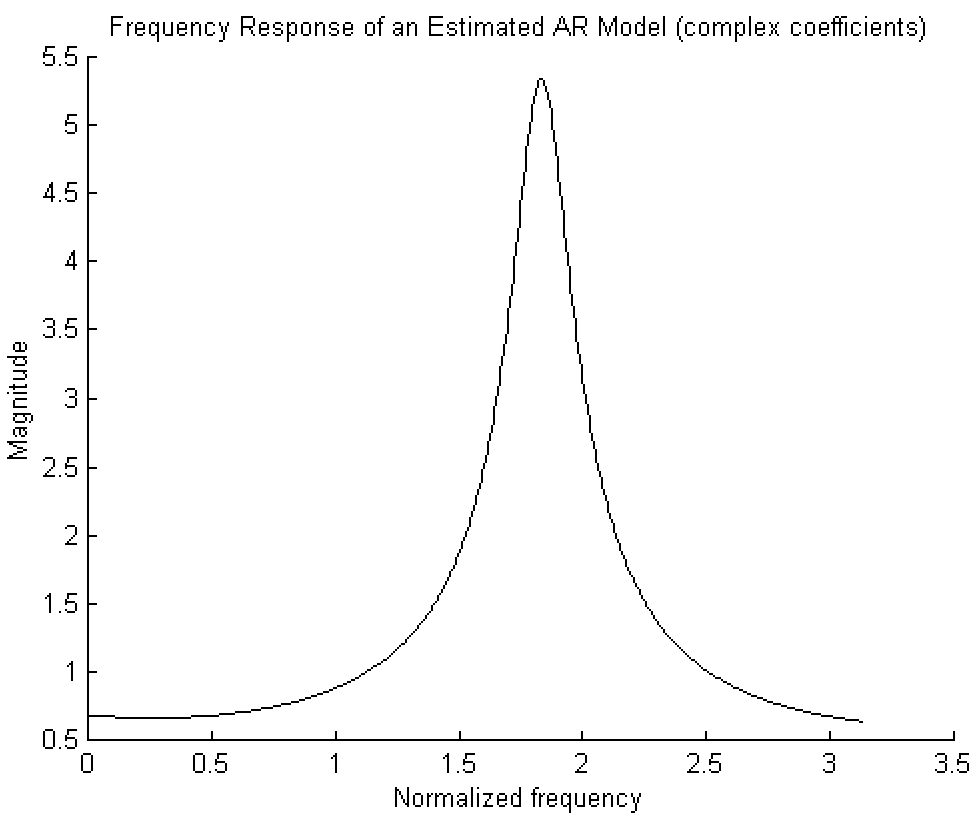

Figure 12. Frequency response of the estimated complex-coefficients AR model

3.2 Yule-Walker Equation and Gaussian Elimination

Another measure of determine or estimate the AR coefficients is solving the Yule-Walker equation [2, pp. 119] which is formulated in matrix form as:

$$\boldsymbol{Rw} = \boldsymbol{r} \tag{10}$$

Where $\boldsymbol{R}$ is the known autocorrelation defined in (7).

$\boldsymbol{w} = [\, w_1\ w_2\ \ldots\ w_M\,]^T$; $w_k = -a_k$ where a_k is the unknown AR coefficients.

$\mathbf{r} = [\, r^*(1)\ r^*(2)\ \ldots\ r^*(M)\,]^T$ is the autocorrelation sequence.

Instead of computing inverse of $\boldsymbol{R}$, say, $\boldsymbol{R}^{-1}$, and substitute in $\boldsymbol{w} = \boldsymbol{R}^{-1}\boldsymbol{r}$, we use Gaussian elimination or Back-substitution method to solve for $\boldsymbol{w}$. Gaussian elimination method uses the idea of decomposition of the matrix $\boldsymbol{R}$ into lower and upper triangular matrices (LU factorization) [4, chap.2.4 and 2.5]. MATLAB solves the equation $\boldsymbol{Rw} = \boldsymbol{r}$ by Gaussian elimination: ***w = R\r*** (see Appendix A.1). The result from solving (10) with the given correlation matrices $\boldsymbol{R}_1$, $\boldsymbol{R_2}$, $\boldsymbol{R}_3$ and $\boldsymbol{R}_4$ are:

w from R1	***w*** from R2	***w*** from R3	***w*** from R4
0.7000	-0.0000 + 0.2500i	0.7080	0.0087 + 0.2554i
-0.3000	-0.5000 - 0.3000i	-0.3116	-0.4876 - 0.3082i
-0.0000	-0.0000 + 0.0000i	0.0114	0.0194 - 0.0118i
0.0000	0.0000 - 0.0000i	-0.0102	0.0096 + 0.0130i
-0.0000	-0.0000 + 0.0000i	-0.0007	0.0422 + 0.0092i
0.0000	0.0000 - 0.0000i	0.0175	0.0174 + 0.0107i
-0.0000	0.0000 - 0.0000i	-0.0083	0.0315 - 0.0025i
0.0000	0.0000 - 0.0000i	-0.0156	0.0143 - 0.0039i
-0.0000	0.0000 + 0.0000i	0.0313	0.0306 - 0.0049i
0.0000	0.0000 - 0.0000i	0.0052	0.0132 - 0.0262i
0.0000	0.0000 - 0.0000i	-0.0051	-0.0062 - 0.0063i
-0.0000	0.0000 - 0.0000i	-0.0154	0.0023 - 0.0067i
-0.0000	0.0000 - 0.0000i	-0.0066	-0.0200 + 0.0199i
0.0000	-0.0000 - 0.0000i	0.0355	0.0043 + 0.0120i
0.0000	0.0000 + 0.0000i	-0.0208	-0.0063 - 0.0084i
-0.0000	-0.0000 - 0.0000i	0.0076	-0.0108 - 0.0122i
0.0000	0.0000 + 0.0000i	-0.0217	-0.0206 + 0.0072i
-0.0000	0.0000 - 0.0000i	0.0341	0.0013 - 0.0146i
-0.0000	-0.0000 + 0.0000i	-0.0436	-0.0572 + 0.0213i
-0.0000	0.0000 - 0.0000i	0.0236	0.0190 + 0.0089i
0.0000	0.0000 + 0.0000i	-0.0118	-0.0182 - 0.0074i
-0.0000	0.0090 - 0.0040i	-0.0027	0.0156 + 0.0063i
0.0000	0.0088 - 0.0065i	0.0115	0.0080 - 0.0098i

Note that $w_k = -a_k$ where k = 1, 2,…, 24 and $a_0 = 1$.

3.3 Computational Efficiency

To compare the computational cost of Levinson-Durbin algorithm and Gaussian elimination used in MATLAB, we use MATLAB function *flops()* to determine the floating point operations. The results are:

-The number of operations by Levinson-Durbin equals *1289* flops.
-The number of operations by Gaussian elimination equals *12008* flops.

We can see that the Levinson-Durbin algorithm takes less computation than the Guassian Elimination (about 10 times). Basically, the important virtue of Levinson-Durbin algorithm is its computational efficiency, in that it results in a big saving in number of operations (multiplication and division) and storage locations compared to Gaussian eliminatoin.

Chapter 4: Summary

The autoregressive (AR) model is a practical method to analyze, synthesize the signal and also forward prediction. It can be characterized by second order statistic such as the correlation function that can be determined by Gohberg-Semencule relation with given AR coefficients. In analysis, to obtain an AR model from observed data, we use its correlation matrix to solve for the AR coefficients via Levinson-Durbin algorithm, which consumes much less computation than other standard method such as Gaussian elimination. In practice, the more time samples will results in more precise of the AR modeling. It can be visualized in figure13, the smoothness of the impulse response increases as the number of samples increase and so does accuracy of the prediction and modeling.

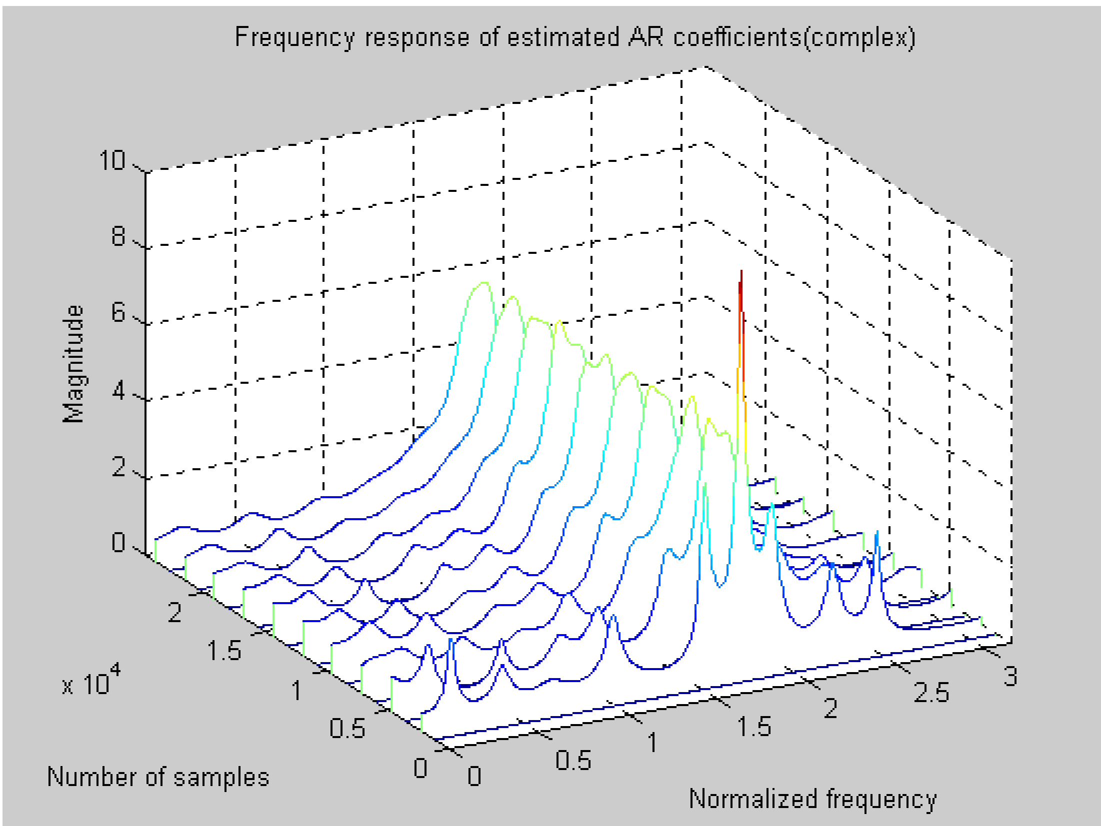

Figure 13. Frequency response of an AR model versus the number of samples

Appendix A

MATLAB software

A.1 Main Script

```
clear all;
close all;
clc;
% Parameters for AR process
a_R = [-0.7 0.3];
a_C = [-0.25*j 0.5+0.3*j];
N = 24; % dimension of problem
M = 1e4; % # of samples
P = N*M; % length of sequence

% Generate unit-variance, zero mean input sequence
x = sqrt(12)*(rand(P,1)-0.5);

% AR(2) process
y1 = filter(1,[1 a_R],x);
y2 = filter(1,[1 a_C],x);

% Produces correlation matrix;R,Via Gohberg-Semencul relation
R_1 = b_f(1,a_R.',N);
R_2 = b_f(1,a_C.',N);

% Produces correlation matrix;R,Via Block Estimate
z1 = reshape(y1,N,M);
z2 = reshape(y2,N,M);

R_3 = (z1*z1')/M;
R_4 = (z2*z2')/M;

% Plot R vesus index
figure;plot(1:N,R_1(1,:),1:N,R_3(1,:));
legend('Asymptotic estimate:Gohberg-Semencul','Block estimate');
figure;plot(1:N,abs(R_2(1,:)),1:N,abs(R_4(1,:)));
legend('Asymptotic estimate:Gohberg-Semencul','Block estimate');

% Plot poles
[Z1,P1,K1]=tf2zp(1,[1 a_R]);
figure;zplane(Z1,P1);
[Z2,P2,K2]=tf2zp(1,[1 a_C]);
figure;zplane(Z2,P2);

% Frequency response
```

```
[H,w]=freqz(1,[1 a_R]);
figure;plot(w,abs(H));
[H,w]=freqz(1,[1 a_C]);
figure;plot(w,abs(H));

% The Levinson-Durbin algorithm
flops(0); % reset flops
r = R_1(:,1); % given correlation sequence
[a_R_hat,AA] = LD(r);
disp('Estimated AR coefficients : a_R from R_1');a_R_hat
disp('The number of operations by Levinson-Durbin');flops

r = R_2(:,1); % given correlation sequence
[a_C_hat,AA] = LD(r);
disp('Estimated AR coefficients : a_C from R_2');a_C_hat

r = R_3(:,1); % given correlation sequence
[a_R_hat,RHO] = LD(r);
disp('Estimated AR coefficients : a_R from R_3');a_R_hat

r = R_4(:,1); % given correlation sequence
[a_C_hat,AA] = LD(r);
disp('Estimated AR coefficients : a_C from R_4');a_C_hat

% Compute AIC and MDL criteria
for k = 1:N-1,
   aic(k) = M*log(RHO(k))+2*k;
   mdl(k) = M*log(RHO(k))+k*log(M);
end
figure;
subplot(211);
aic_min = find(aic == min(aic));
s1 = semilogy(1:N-1,aic,[aic_min aic_min],[min(aic) max(aic)]);
set(gca,'XTick',1:N-1);
xlabel('k');
ylabel('AIC[k]');
grid;

subplot(212);
mdl_min = find(mdl == min(mdl));
s2 = semilogy(1:N-1,mdl,[mdl_min mdl_min],[min(mdl) max(mdl)]);
set(gca,'XTick',1:N-1);
xlabel('k');
ylabel('MDL[k]');
grid;

% Plot poles from estimated AR coefficients
[Z1,P1,K1]=tf2zp(1,[a_R_hat(1:mdl_min+1)]);
```

```
figure;zplane(Z1,P1);
[Z2,P2,K2]=tf2zp(1,[a_C_hat(1:mdl_min+1)]);
figure;zplane(Z2,P2);

% Frequency response
[H,w]=freqz(1,[a_R_hat(1:mdl_min)]);
figure;hold on;plot(w,abs(H));hold off;
[H,w]=freqz(1,[a_C_hat(1:mdl_min)]);
figure;hold on;plot(w,abs(H));hold off;

% Gaussian elimination method to solve for w in Rw = r
flops(0); % reset flops
r = R_1((2:N),1);
R_1 = R_1(1:N-1,1:N-1);
disp('Estimated AR coefficients : w (= -a_R) from R_1')
w = R_1\r
disp('The number of operations by Guassian elimination:');flops

r = R_2((2:N),1);
R_2 = R_2(1:N-1,1:N-1);
disp('Estimated AR coefficients : w (= -a_C) from R_2')
w = R_2\r

r = R_3((2:N),1);
R_3 = R_3(1:N-1,1:N-1);
disp('Estimated AR coefficients : w (= -a_R) from R_3')
w = R_3\r

r = R_4((2:N),1);
R_4 = R_4(1:N-1,1:N-1);
disp('Estimated AR coefficients : w (= -a_C) from R_4')
w = R_4\r
```

A.2 Gohberg-Semencule Relation

```
% b_f.m
% function y=b_f(varnoiz,a,N);
% Compute covariance matrix based on
% Gohberg-Semencul formular

function y=b_f(varnoiz,a,N);

if N==1,
  y=1;
  return;
end;
```

```
A1=A1_F(a,N); % Compute matrix A1
A3=A3_F(a,N); % Compute matrix A3
y=inv((1/varnoiz)*(A1*A1'-A3*A3')); % True noise cov. matrix

return;

%----------------------
function y=A1_F(a,N);

if N==1,
   y=1;
   return;
end;

% Compute the matrix A1
y=toeplitz([1;a;zeros((N-length(a)-1),1)],[1 zeros(1,N-1)]);
[a,b]=size(y);
if a ~= b,
   error('Matrix is not square!');
end;
return;

%----------------------
function y=A3_F(a,N);

if N==1,
   y=1;
   return;
end;

% Compute the matrix A3
y=toeplitz([zeros(N-length(a),1);flipud(a)],zeros(1,N));
[a,b]=size(y);
if a ~= b,
   error('Matrix is not square!');
end;
return;
```

A.3 Levinson-Durbin Algorithm

```
% The Levinson-Durbin algorithm
%
% function [a,P] = LD(r)
% Estimate AR coefficients from a correlation sequence
% input :
%       : "r" is a given correlation sequence r(0) r(1)..r(M-1)
```

```
% output:
%       : "a" is an AR coefficient sequence 1,a(1),a(2),..,a(M)
%       : "P" is the forward prediction error power
% By: Wiwat T. ,May 2000
function [a,P] = LD(r)

M = length(r)-1;
r = reshape(r,M+1,1);
a = 1; P = r(1); K = []; % initialization
for i = 1:M,
   Ki = -(a*r(i+1:-1:2))/P(i);
   K = [K,Ki];
   Pi = P(i)*(1-abs(Ki)^2);
   P = [P,Pi];
   a = [a,0];
   a = a + Ki.*conj(fliplr(a));
end
return;
```

A.4 Frequency Responses Versus Number of Samples

```
clear;close all;clc;
figure(1);
n=20;
m=512;
h=zeros(m,n)';
a_C = [-0.25*j 0.5+0.3*j];
N = 24; % dimension of problem
M = 1e4; % # of samples
P = N*M; % length of sequence

% Generate unit-variance, zero mean input sequence
x = sqrt(12)*(rand(P,1)-0.5);

% AR(2) process
y2 = filter(1,[1 a_C],x);

for i = 2:2:n,
M = i*50;
y = y2(1:M*N,1);
% Produces correlation matrix;R,Via Block Estimate
z2 = reshape(y,N,M);
R_4 = (z2*z2')/M;

% The Levinson-Durbin algorithm
r = R_4(:,1); % given correlation sequence
```

```
a_C_hat = LD(r);
%disp('Estimated AR coefficients : a_C from R_4');a_C_hat(1:3)

% Frequency response
[H,w]=freqz(1,a_C_hat,m);
h(i,:)=abs(H)';
end

waterfall(w,1200*[1:n],h);
title('Frequency response of estimated AR coefficients(complex)');
xlabel('Normalized frequency');
ylabel('Number of samples');
zlabel('Magnitude');
```

Bibliography

[1] B. Cernuschi-Frias, "A Derivation of the Gohberg-Semencule Relation," *IEEE Transaction on Signal Processing,* vol.39, pp. 190-192, Jan. 1991

[2] Simon Haykins. "*Adaptive Filter Theory."* Prentice-Hall, Inc., Englewood Cliffs, NJ, Third Edition, 1996. ISBN 0-13-322760-x.

[3] Boaz Porat. *"A Course in Digital Signal Processing."* John Wiley & Sons, Inc., New York, 1997. ISBN 0-41-14961-6.

[4] Jack L. Goldberg. *"Matrix Theory with Applications."* McGraw-Hill, Inc., NJ. ISBN 0-07-557200-1.