

MATH 545 - ASSIGNMENT 4 SOLUTIONS

1. Compute and report the value of the AIC model selection criterion for the collection of $\text{ARMA}(p, q)$ models for $0 \leq p, q \leq 2$ using the data set contained in the file

<https://www.math.mcgill.ca/dstephens/545/Assignments/A4/Q1.txt>

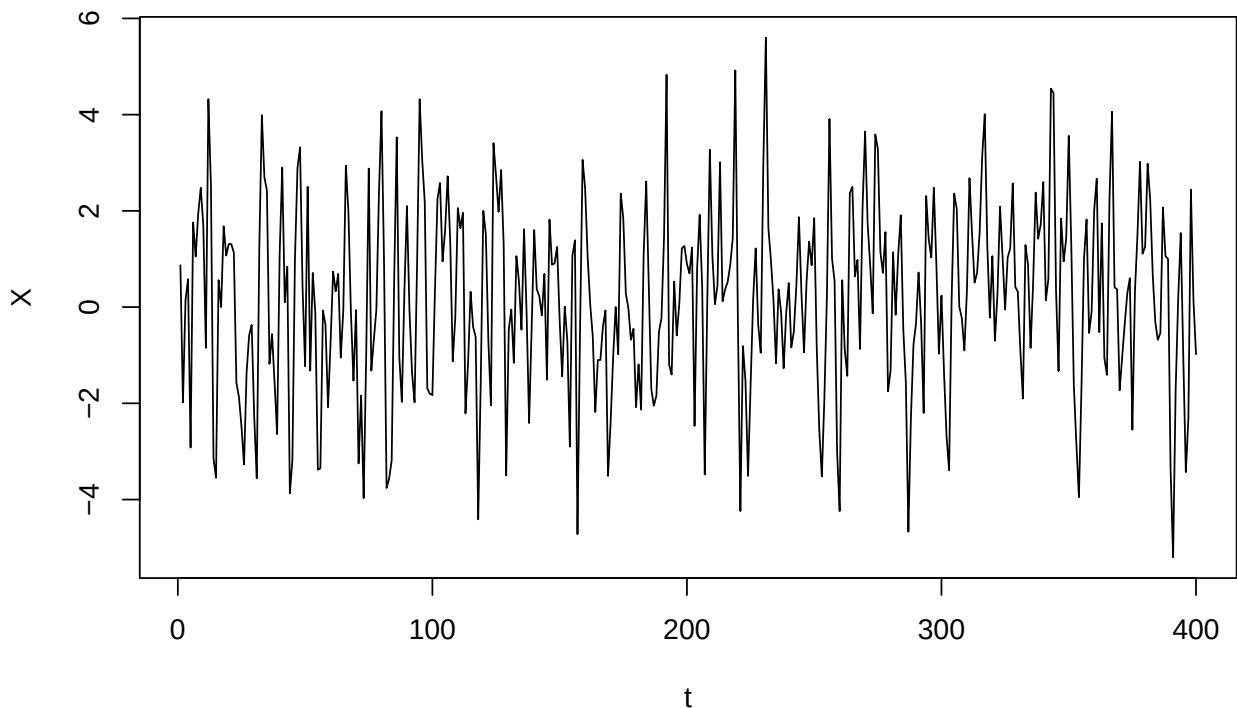
and hence report the selected model from those fitted.

4 Marks

```
file.url<-'https://www.math.mcgill.ca/dstephens/545/Assignments/A4/Q1.txt'
X<-scan(file.url)
n<-length(X)
n

+ [1] 400

par(mar=c(4, 4, 2, 0))
plot(X, type='l', xlab='t')
```



In R, the function `arima` may be used to compute the AIC quantity.

Inspect the sample ACF of the residuals for the two most favoured models under the AIC criterion, and hence comment on the suitability of these models.

2 Marks

Solution: We compute as follows

```
file.url<-'https://www.math.mcgill.ca/dstephens/545/Assignments/A4/Q1.txt'
X<-scan(file.url)
aic.mat<-matrix(0,3,3)
for(p in 0:2){
  for(q in 0:2){
    aic.mat[p+1,q+1]<-arima(X,order=c(p,0,q))$aic
  }
}
rownames(aic.mat)<-paste('p=',0:2,sep=' ');
colnames(aic.mat)<-paste('q=',0:2,sep=' ')
aic.mat

+           q=0           q=1           q=2
+ p=0 1661.143 1564.137 1565.813
+ p=1 1598.839 1565.889 1564.639
+ p=2 1567.807 1565.689 1566.014

pval<-rep(0:2,3)
qval<-rep(0:2,each=3)
aic.table<-as.data.frame(cbind(pval,qval,as.vector(aic.mat)))
names(aic.table)<-c('p','q','AIC')
aic.table<-aic.table[order(aic.table$AIC),]
aic.table

+   p q      AIC
+ 4 0 1 1564.137
+ 8 1 2 1564.639
+ 6 2 1 1565.689
+ 7 0 2 1565.813
+ 5 1 1 1565.889
+ 9 2 2 1566.014
+ 3 2 0 1567.807
+ 2 1 0 1598.839
+ 1 0 0 1661.143
```

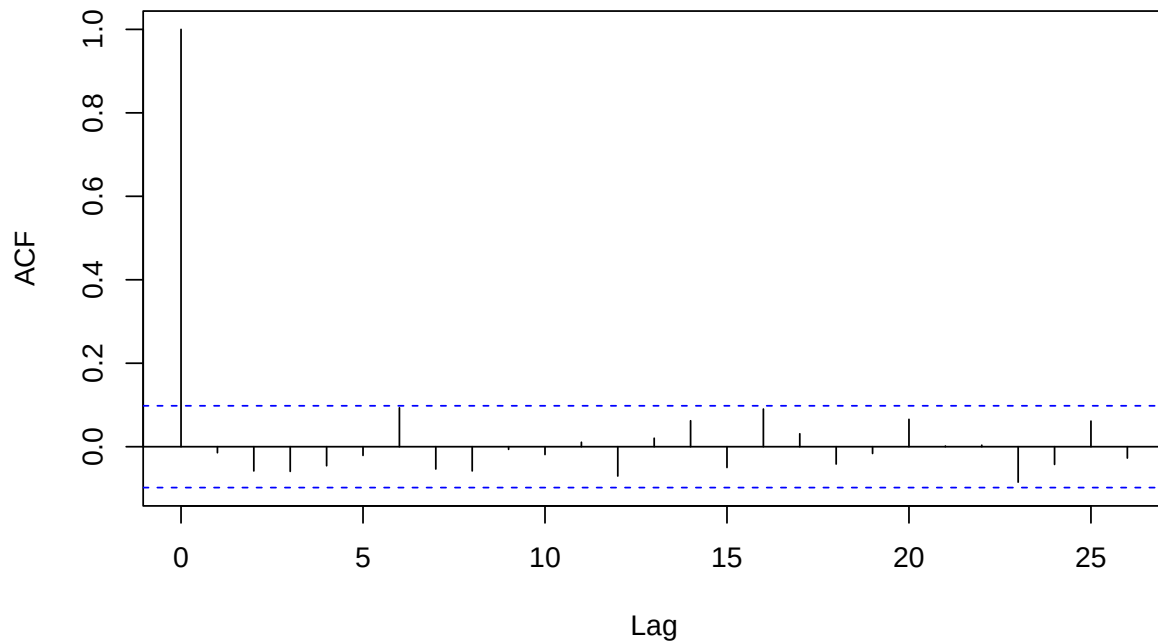
The model with the smallest AIC is selected: this is the ARMA(0,0,1) (that is, the MA(1) model) with AIC equal to 1564.137; however, the ARMA(1,0,2) model with AIC equal to 1564.639 is competitive.

4 Marks

We can compute the residuals from the two fits, and plot their sample ACFs.

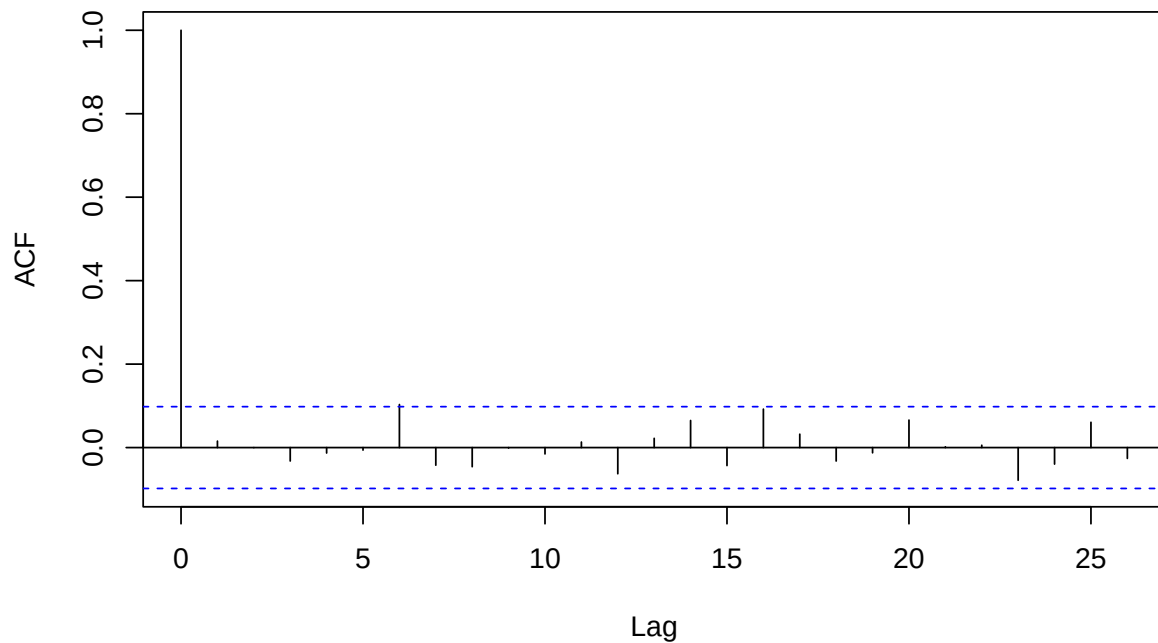
```
res1<-residuals(arima(X,order=c(0,0,1)))
res2<-residuals(arima(X,order=c(1,0,2)))
acf(res1,main='ACF for residuals from ARMA(0,0,1) fit')
```

ACF for residuals from ARMA(0,0,1) fit



```
acf(res2, main='ACF for residuals from ARMA(1,0,2) fit')
```

ACF for residuals from ARMA(1,0,2) fit



These residuals seem to be plausibly white noise series, so both of the models seem to be adequate explanations. 2 Marks

2. Compute the first part of the autocovariance matrix sequence, $\Gamma_h, h = 0, 1, 2, 3$, for following bivariate processes $\{\mathbf{X}_t\}$ if, in each case, $\mathbf{Z}_t \sim \text{Normal}_2(\mathbf{0}, \mathbf{I}_2)$.

(a) The VAR(1) process

$$\mathbf{X}_t = \begin{bmatrix} 0.90 & 0.40 \\ 0.00 & 0.20 \end{bmatrix} \mathbf{X}_{t-1} + \mathbf{Z}_t$$

4 Marks

(b) The VMA(1) process

$$\mathbf{X}_t = \mathbf{Z}_t + \begin{bmatrix} 1.00 & -0.50 \\ 0.50 & -1.00 \end{bmatrix} \mathbf{Z}_{t-1}$$

4 Marks

(c) The VARMA(1, 1) process

$$\mathbf{X}_t = \begin{bmatrix} 0.50 & 0.50 \\ 0.00 & 0.50 \end{bmatrix} \mathbf{X}_{t-1} + \mathbf{Z}_t + \begin{bmatrix} 0.50 & 0.50 \\ 0.00 & 0.50 \end{bmatrix} \mathbf{Z}_{t-1}$$

6 Marks

Solution:

(a) From results in lectures, we know that

$$\Gamma_0 = \Sigma + \sum_{j=1}^{\infty} \Phi^j \Sigma (\Phi^T)^j = \mathbf{I}_2 + \sum_{j=1}^{\infty} (\Phi \Phi^T)^j$$

and as

$$\Phi \Phi^T = \begin{bmatrix} 0.90 & 0.40 \\ 0.00 & 0.20 \end{bmatrix} \begin{bmatrix} 0.90 & 0.00 \\ 0.40 & 0.20 \end{bmatrix} = \begin{bmatrix} 0.97 & 0.08 \\ 0.08 & 0.04 \end{bmatrix}$$

we can compute Γ_0 as follows

```
Phi<-matrix(c(0.9,0.4,0.0,0.2),2,2,byrow=T)
Gamma0<-Ptmp<-diag(c(1,1))
for(j in 1:10000){
  Ptmp<-Phi %*% (Ptmp %*% t(Phi))
  Gamma0<-Gamma0+Ptmp
}
Gamma0

##           [,1]      [,2]
## [1,]  6.525460  0.101626
## [2,]  0.101626  1.041667
```

Similarly for $h = 1, 2, 3$, we have

$$\Gamma_h = \Phi \Gamma_{h-1} = \Phi^h \Gamma_0$$

which we can compute as follows:

```
Gammah<-array(0,c(3,2,2))
Gammah[1,,]<-Phi %*% Gamma0
for(h in 2:3){
  Gammah[h,,]<-Phi %*% Gammah[h-1,,]
}
Gammah[1,,]
```

```
##           [,1]      [,2]
## [1,]  5.9135644 0.5081301
## [2,]  0.0203252 0.2083333

Gammah[2,,]

##           [,1]      [,2]
## [1,]  5.330338040 0.54065041
## [2,]  0.004065041 0.04166667

Gammah[3,,]

##           [,1]      [,2]
## [1,]  4.7989302525 0.503252033
## [2,]  0.0008130081 0.008333333
```

Hence

$$\mathbf{\Gamma}_0 = \begin{bmatrix} 6.5255 & 0.1016 \\ 0.1016 & 1.0417 \end{bmatrix}$$

and

$$\mathbf{\Gamma}_1 = \begin{bmatrix} 5.9136 & 0.5081 \\ 0.0203 & 0.2083 \end{bmatrix} \quad \mathbf{\Gamma}_2 = \begin{bmatrix} 5.3303 & 0.5407 \\ 0.0041 & 0.0417 \end{bmatrix} \quad \mathbf{\Gamma}_3 = \begin{bmatrix} 4.7989 & 0.5033 \\ 0.0008 & 0.0083 \end{bmatrix}$$

4 Marks

We can check the solution using the MTS package in R:

```
library(MTS)
VARMAcov(Phi = Phi, Sigma = diag(c(1,1)), lag = 3)

## Auto-Covariance matrix of lag:  0
##           [,1]      [,2]
## [1,]  6.52546 0.10163
## [2,]  0.10163 1.04167
## Auto-Covariance matrix of lag:  1
##           [,1]      [,2]
## [1,]  5.91356 0.50813
## [2,]  0.02033 0.20833
## Auto-Covariance matrix of lag:  2
##           [,1]      [,2]
## [1,]  5.33034 0.54065
## [2,]  0.00407 0.04167
## Auto-Covariance matrix of lag:  3
##           [,1]      [,2]
## [1,]  4.79893 0.50325
## [2,]  0.00081 0.00833
## cross correlation matrix of lag:  0
##           [,1]      [,2]
## [1,]  1.000 0.039
## [2,]  0.039 1.000
## cross correlation matrix of lag:  1
##           [,1]      [,2]
## [1,]  0.9062 0.1949
## [2,]  0.0078 0.2000
## cross correlation matrix of lag:  2
```

```
##           [,1]      [,2]
## [1,] 0.8169 0.2074
## [2,] 0.0016 0.0400
## cross correlation matrix of lag: 3
##           [,1]      [,2]
## [1,] 0.7354 0.193
## [2,] 0.0003 0.008
```

Additional details: We can compute the corresponding ACF matrices

$$\rho_h = \mathbf{D}^{-1}\Gamma_h\mathbf{D}^{-1}$$

where $\mathbf{D} = \text{diag}(\sqrt{\Gamma_{11}(0)}, \sqrt{\Gamma_{22}(0)})$.

```
D<-diag(sqrt(diag(Gamma0)))
Dinv<-solve(D)
Rho0<-Dinv %*% (Gamma0 %*% Dinv)
Rho0<-Gammah*0
for(h in 1:3){
  Rhoh[h,,]<-Dinv %*% (Gammah[h,,] %*% Dinv)
}
Rho0

##           [,1]      [,2]
## [1,] 1.00000000 0.03897938
## [2,] 0.03897938 1.00000000

Rhoh[1,,]

##           [,1]      [,2]
## [1,] 0.906229508 0.1948969
## [2,] 0.007795876 0.2000000

Rhoh[2,,]

##           [,1]      [,2]
## [1,] 0.816852459 0.2073703
## [2,] 0.001559175 0.0400000

Rhoh[3,,]

##           [,1]      [,2]
## [1,] 0.7354163934 0.1930259
## [2,] 0.0003118351 0.0080000
```

(b) For the VMA(1) process we have

$$\Gamma_0 = \Sigma + \Theta\Sigma\Theta^T$$

and

$$\Gamma_1 = \Theta\Sigma\Theta_0 = \Theta$$

where $\Theta_0 = \mathbf{I}_2$, with $\Gamma_h = \mathbf{0}$ for $h > 1$.

```
Gamma0<-diag(c(1,1))+Th %*% t(Th)
Gammah<-array(0,c(3,2,2))
Gammah[1,,]<-Th
Gamma0; Gammah[1,,]
```

```
##      [,1] [,2]
## [1,] 2.25 1.00
## [2,] 1.00 2.25
##      [,1] [,2]
## [1,] 1.0  -0.5
## [2,] 0.5  -1.0
```

Hence

$$\Gamma_0 = \begin{bmatrix} 2.25 & 1.00 \\ 1.00 & 2.25 \end{bmatrix} \quad \Gamma_1 = \begin{bmatrix} 1.00 & -0.50 \\ 0.50 & -1.00 \end{bmatrix}$$

with $\Gamma_2 = \Gamma_3 = \mathbf{0}$.

4 Marks

```
library(MTS)
VARMAcov(Theta = -Th, Sigma = diag(c(1,1)), lag = 3)

## Auto-Covariance matrix of lag:  0
##      [,1] [,2]
## [1,] 2.25 1.00
## [2,] 1.00 2.25
## Auto-Covariance matrix of lag:  1
##      [,1] [,2]
## [1,] 1.0  -0.5
## [2,] 0.5  -1.0
## Auto-Covariance matrix of lag:  2
##      [,1] [,2]
## [1,] 0    0
## [2,] 0    0
## Auto-Covariance matrix of lag:  3
##      [,1] [,2]
## [1,] 0    0
## [2,] 0    0
## cross correlation matrix of lag:  0
##      [,1] [,2]
## [1,] 1.0000 0.4444
## [2,] 0.4444 1.0000
## cross correlation matrix of lag:  1
##      [,1] [,2]
## [1,] 0.4444 -0.2222
## [2,] 0.2222 -0.4444
## cross correlation matrix of lag:  2
##      [,1] [,2]
## [1,] 0    0
## [2,] 0    0
## cross correlation matrix of lag:  3
##      [,1] [,2]
## [1,] 0    0
## [2,] 0    0
```

Additional details: We can compute the corresponding ACF matrices

```
D<-diag(sqrt(diag(Gamma0)))
Dinv<-solve(D)
Rho0<-Dinv %*% (Gamma0 %*% Dinv)
Rho0<-Gamma0
```

```

for(h in 1:1){
  Rhoh[h,,]<-Dinv %*% (Gammah[h,,] %*% Dinv)
}
Rho0

##           [,1]      [,2]
## [1,] 1.0000000 0.4444444
## [2,] 0.4444444 1.0000000

Rhoh[1,,]

##           [,1]      [,2]
## [1,] 0.4444444 -0.2222222
## [2,] 0.2222222 -0.4444444

```

(c) Let

$$\Phi = \begin{bmatrix} 0.50 & 0.50 \\ 0.00 & 0.50 \end{bmatrix}$$

so that we have the VARMA(1, 1) process

$$\mathbf{X}_t = \Phi \mathbf{X}_{t-1} + \mathbf{Z}_t + \Phi \mathbf{Z}_{t-1}$$

or equivalently

$$(\mathbf{I}_2 - \Phi B)X_t = (\mathbf{I}_2 + \Phi B)\mathbf{Z}_t.$$

We have the linear process-type representation in terms of the operator $\Psi(B)$, where

$$\Psi(z) = (\mathbf{I}_2 - \Phi z)^{-1}(\mathbf{I}_2 + \Phi z).$$

Now

$$\begin{aligned} (\mathbf{I}_2 - \Phi z)^{-1}(\mathbf{I}_2 + \Phi z) &= (\mathbf{I}_2 + \Phi z + \Phi^2 z^2 + \Phi^3 z^3 \dots)(\mathbf{I}_2 + \Phi z) \\ &= \mathbf{I}_2 + 2\Phi z + 2\Phi^2 z^2 + 2\Phi^3 z^3 \dots \end{aligned}$$

so therefore in the linear process representation $\Psi_0 = \mathbf{I}_2$, and

$$\Psi_j = 2\Phi^j \quad j = 1, 2, \dots$$

Thus

$$\mathbf{X}_t = \mathbf{Z}_t + 2 \sum_{j=1}^{\infty} \Phi^j \mathbf{Z}_{t-j}$$

Hence, taking variances on both sides of this representation

$$\Gamma_0 = \mathbf{I}_2 + 4 \sum_{j=1}^{\infty} \Phi^j (\Phi^j)^\top$$

and using the usual approach (for the zero-mean process)

$$\mathbb{E}[\mathbf{X}_t \mathbf{X}_{t-h}^\top] = \mathbb{E} \left[\left(\mathbf{Z}_t + 2 \sum_{j=1}^{\infty} \Phi^j \mathbf{Z}_{t-j} \right) \left(\mathbf{Z}_{t-h} + 2 \sum_{k=1}^{\infty} \Phi^k \mathbf{Z}_{t-h-k} \right)^\top \right]$$

Using the usual uncorrelatedness arguments, we have for $h \geq 1$

$$\begin{aligned}
 \mathbb{E}[\mathbf{X}_t \mathbf{X}_{t-h}^\top] &= 2\mathbb{E}\left[\sum_{j=1}^{\infty} \Phi^j \mathbf{Z}_{t-j} \mathbf{Z}_{t-h}^\top\right] + 4\mathbb{E}\left[\sum_{j=1}^{\infty} \sum_{k=1}^{\infty} \Phi^j \mathbf{Z}_{t-j} \mathbf{Z}_{t-h-k}^\top (\Phi^k)^\top\right] \\
 &= 2\Phi^h + 4\sum_{j=1}^{\infty} \sum_{k=1}^{\infty} \Phi^j \mathbb{E}\left[\mathbf{Z}_{t-j} \mathbf{Z}_{t-h-k}^\top\right] (\Phi^k)^\top \\
 &= 2\Phi^h + 4\sum_{k=1}^{\infty} \Phi^{k+h} (\Phi^k)^\top \quad \text{only } j = h + k \text{ contributes} \\
 &= 2\Phi^h + \Phi^h \sum_{k=1}^{\infty} \Phi^k (\Phi^k)^\top.
 \end{aligned}$$

```

Phi<-matrix(c(0.5,0.5,0.0,0.5),2,2,byrow=T)
Ptmp<-diag(c(1,1))
Psum<-Ptmp*0
for(j in 1:10000){
  Ptmp<-Phi %*% Ptmp
  Psum<-Psum+Ptmp %*% t(Ptmp)
}
Gamma0<-diag(c(1,1))+4*Psum
Gamma0

##           [,1]      [,2]
## [1,]  5.296296  1.777778
## [2,]  1.777778  2.333333

```

```

Gammah<-array(0,c(3,2,2))
Gammah[1,,]<-2*Phi + 4*Phi %*% Psum
Ptmp<-Phi
for(h in 2:3){
  Ptmp<-Phi%*% Ptmp
  Gammah[h,,]<-2*Ptmp + 4 * Ptmp %*% Psum
}

Gammah[1,,]

##           [,1]      [,2]
## [1,]  4.0370370  2.555556
## [2,]  0.8888889  1.666667

Gammah[2,,]

##           [,1]      [,2]
## [1,]  2.4629630  2.111111
## [2,]  0.4444444  0.8333333

Gammah[3,,]

##           [,1]      [,2]
## [1,]  1.4537037  1.4722222
## [2,]  0.2222222  0.4166667

```

Thus

$$\Gamma_0 = \begin{bmatrix} 5.2963 & 1.7778 \\ 1.7778 & 2.3333 \end{bmatrix}$$

and

$$\Gamma_1 = \begin{bmatrix} 4.0370 & 2.5556 \\ 0.8889 & 1.6667 \end{bmatrix} \quad \Gamma_2 = \begin{bmatrix} 2.4630 & 2.1111 \\ 0.4444 & 0.8333 \end{bmatrix} \quad \Gamma_3 = \begin{bmatrix} 1.4537 & 1.4722 \\ 0.2222 & 0.4167 \end{bmatrix}$$

6 Marks

We can check the details using VARMAcov

```
library(MTS)
VARMAcov(Phi = Phi, Theta=-Phi, Sigma = diag(c(1,1)), lag = 3, trun=1000)

## Auto-Covariance matrix of lag:  0
##      [,1]      [,2]
## [1,] 5.29630 1.77778
## [2,] 1.77778 2.33333
## Auto-Covariance matrix of lag:  1
##      [,1]      [,2]
## [1,] 4.03704 2.55556
## [2,] 0.88889 1.66667
## Auto-Covariance matrix of lag:  2
##      [,1]      [,2]
## [1,] 2.46296 2.11111
## [2,] 0.44444 0.83333
## Auto-Covariance matrix of lag:  3
##      [,1]      [,2]
## [1,] 1.45370 1.47222
## [2,] 0.22222 0.41667
## cross correlation matrix of lag:  0
##      [,1]      [,2]
## [1,] 1.0000 0.5057
## [2,] 0.5057 1.0000
## cross correlation matrix of lag:  1
##      [,1]      [,2]
## [1,] 0.7622 0.7270
## [2,] 0.2529 0.7143
## cross correlation matrix of lag:  2
##      [,1]      [,2]
## [1,] 0.4650 0.6005
## [2,] 0.1264 0.3571
## cross correlation matrix of lag:  3
##      [,1]      [,2]
## [1,] 0.2745 0.4188
## [2,] 0.0632 0.1786
```

```
D<-diag(sqrt(diag(Gamma0)))
Dinv<-solve(D)
Rho0<-Dinv %*% (Gamma0 %*% Dinv)
Rho<-Gammah*0
for(h in 1:3){
  Rho[h,,]<-Dinv %*% (Gammah[h,,] %*% Dinv)
}
Rho0
```

```

##           [,1]      [,2]
## [1,]  1.0000000  0.5057116
## [2,]  0.5057116  1.0000000

Rho[1,,]

##           [,1]      [,2]
## [1,]  0.7622378  0.7269605
## [2,]  0.2528558  0.7142857

Rho[2,,]

##           [,1]      [,2]
## [1,]  0.4650350  0.6005326
## [2,]  0.1264279  0.3571429

Rho[3,,]

##           [,1]      [,2]
## [1,]  0.27447552  0.4187924
## [2,]  0.06321395  0.1785714

```