Contents


```
% Endsem CH5440
% Ojas Phadake - CH22B007

clc;
clear all;
close all;

load arx.mat
```

A

Scaling appropriately the measurements:

```
ymeass = ymeas/stdey;
umeass = umeas/stdeu;

L = 10; % lag
nsamples = 1024;

ZL = [];
for i = L+1:-1:1
    ZL = [ZL ymeass(i:nsamples+i-L-1)];
end
for i = L+1:-1:1
    ZL = [ZL umeass(i:nsamples+i-L-1)];
end
[u s v] = svd(ZL/sqrt(nsamples-L),'econ');
lambda = diag(s).^2;

% Hypothesis test
alpha = 0.05;
nvar = size(ZL,2);
tau = zeros(nvar-2,1);
crit = zeros(nvar-2,1);
for d = nvar-1:-1:2
   nu = (d-1)*(d+2)/2;
   nprime = nsamples-nvar - (2*nvar+11)/6;
   lbar = mean(lambda(nvar-d+1:end));
   tau(d-1) = nprime*(d*log(lbar)-sum(log(lambda(nvar-d+1:end))));
   crit(d-1) = chi2inv(1-alpha,nu);
end

disp(table(crit, tau))

% Hypothesis testing
flag = 1;
d = nvar-2;
dest = 1;

while flag
    if ( tau(d) > crit(d) )
        d = d-1;
        if ( d < 2)
           flag = 0;
        end
    else
        dest = d + 1;
        flag = 0;
    end
```

```
end

d1 = dest;
eta1 = L - d1 + 1;

% Last PCA applied using known eta
Zeta1 = [];
for i = eta1+1:-1:1
    Zeta1 = [Zeta1 ymeass(i:nsamples+i-eta1-1)];
end
for i = eta1+1:-1:1
    Zeta1 = [Zeta1 umeass(i:nsamples+i-eta1-1)];
end
[u s v] = svd(Zeta1/sqrt(nsamples-eta1),'econ');
theta1 = v(:,end)';
theta1(1:eta1+1) = theta1(1:eta1+1)/stdey;
theta1(eta1+2:end) = theta1(eta1+2:end)/stdeu;
theta1 = theta1/theta1(1);
```

crit	tau
5.9915	0.0034248
11.07	0.54265
16.919	0.86741
23.685	1.1445
31.41	5.828
40.113	10.784
49.802	16.382
60.481	25.254
72.153	211.49
84.821	3810.1
98.484	6408.8
113.15	8481.4
128.8	10273
145.46	11718
163.12	12792
181.77	13730
201.42	14635
222.08	15493
243.73	16332
266.38	17184

Now applying for L = 20

```
fprintf("Now carrying out the same for L = 20. ")
L = 20; % Modified lag

ZL = [];
for i = L+1:-1:1
    ZL = [ZL ymeass(i:nsamples+i-L-1)];
end
for i = L+1:-1:1
    ZL = [ZL umeass(i:nsamples+i-L-1)];
end
[u s v] = svd(ZL/sqrt(nsamples-L),'econ');
lambda = diag(s).^2;

% Hypothesis test
alpha = 0.05;
nvar = size(ZL,2);
tau = zeros(nvar-2,1);
crit = zeros(nvar-2,1);
for d = nvar-1:-1:2
   nu = (d-1)*(d+2)/2;
   nprime = nsamples-nvar - (2*nvar+11)/6;
```

```
    lbar = mean(lambda(nvar-d+1:end));
    tau(d-1) = nprime*(d*log(lbar)-sum(log(lambda(nvar-d+1:end))));
    crit(d-1) = chi2inv(1-alpha,nu);
end

disp(table(crit, tau))

% Hypothesis testing
flag = 1;
d = nvar-2;
dest = 1;

while flag
    if ( tau(d) > crit(d) )
        d = d-1;
        if ( d < 2)
           flag = 0;
        end
    else
        dest = d + 1;
        flag = 0;
    end
end

d = dest;
eta = L - d + 1;

fprintf("We get the value of eta as %0.1d", eta);
fprintf("\nThis means that the system depends on the memory of inputs and outputs of the last 2 time instants. ")

% Last PCA applied using known eta
Zeta = [];
for i = eta+1:-1:1
    Zeta = [Zeta ymeass(i:nsamples+i-eta-1)];
end
for i = eta+1:-1:1
    Zeta = [Zeta umeass(i:nsamples+i-eta-1)];
end
[u s v] = svd(Zeta/sqrt(nsamples-eta),'econ');
theta = v(:,end)';
theta(1:eta+1) = theta(1:eta+1)/stdey;
theta(eta+2:end) = theta(eta+2:end)/stdeu;
theta = theta/theta(1);

fprintf("\nThe obtained coefficient vector is: ")
disp(theta)
fprintf("\nThus we notice that considering the lag as 10 and 20 gave different values of eta.\n")

fprintf("**************************************************************************************************************************")
```

```
Now carrying out the same for L = 20.      crit       tau
    ______    ________

    5.9915    0.050355
     11.07      8.0547
    16.919      12.433
    23.685      15.847
     31.41       19.42
    40.113      26.532
    49.802      32.862
    60.481      46.097
    72.153      60.371
    84.821      75.831
    98.484      90.645
    113.15       108.6
     128.8      127.98
    145.46      149.53
    163.12      170.53
    181.77      199.78
```

```
  201.42      234.46
  222.08      266.79
  243.73      532.47
  266.38      4768.5
  290.03      7927.5
  314.68       11493
  340.33       14270
  366.98       16731
  394.63       18990
  423.27       20879
  452.92       22462
  483.57       23802
  515.22       24975
  547.87       25986
  581.51       26973
  616.16       27989
  651.81       28891
  688.45       29831
   726.1       30691
  764.75       31592
   804.4       32470
  845.04       33278
  886.69       34033
  929.33       34856

We get the value of eta as 7
This means that the system depends on the memory of inputs and outputs of the last 2 time instants.
The obtained coefficient vector is:   Columns 1 through 7

    1.0000    1.4350    1.3663    0.3725   -0.4855   -0.9946   -0.5913

  Columns 8 through 14

   -0.1439   -0.0603   -2.0450   -2.0657   -1.5427    0.5462    1.5785

  Columns 15 through 16

    2.0306    0.7632


Thus we notice that considering the lag as 10 and 20 gave different values of eta.
********************************************************************************************************************************
```

B

```
fprintf("For L = 10:\n")
fprintf("We get the value of eta as %0.1d", eta1);
fprintf("\nThis means that the system depends on the memory of inputs and outputs of the last 2 time instants. ")
fprintf("\nThe obtained coefficient vector is: ")
disp(theta1)

fprintf("For L = 20:\n")
fprintf("We get the value of eta as %0.1d", eta);
fprintf("\nThis means that the system depends on the memory of inputs and outputs of the last 2 time instants. ")
fprintf("\nThe obtained coefficient vector is: ")
disp(theta)
```

```
For L = 10:
We get the value of eta as 2
This means that the system depends on the memory of inputs and outputs of the last 2 time instants.
The obtained coefficient vector is:     1.0000   -0.5456   -0.2360    0.0236   -1.9700    1.8937

For L = 20:
We get the value of eta as 7
This means that the system depends on the memory of inputs and outputs of the last 2 time instants.
The obtained coefficient vector is:   Columns 1 through 7
```

```
    1.0000    1.4350    1.3663    0.3725   -0.4855   -0.9946   -0.5913

  Columns 8 through 14

   -0.1439   -0.0603   -2.0450   -2.0657   -1.5427    0.5462    1.5785

  Columns 15 through 16

    2.0306    0.7632
```

Bootstrapping

```
nboot = 100;
nsub = 700;

[N, nvar] = size(Zeta1); % 1019, 12
theta1 = zeros(nboot,nvar);

for i = 1:nboot
    ind = randperm(N);
    Zsub = Zeta1(ind(1:nsub),:);
    [u s v] = svd(Zsub/sqrt(nsub),'econ');
    theta1(i,:) = v(:,end)';
    theta1(i,:) = theta1(i,:)/theta1(i,1);  % Normalize theta vector so that first coefficient is unity
end

% Find mean and std of theta elements

thetamean = mean(theta1);
thetastd = std(theta1);
thetal = thetamean - 2*thetastd;
thetau = thetamean + 2*thetastd;
ind = 1:1:6;

fprintf("\nThe 95% confidence intervals using bootstrapping are as follows for L=10:\n")
to_show = table(ind', thetal', thetau');
disp(to_show);

nboot = 100;
nsub = 700;

[N, nvar] = size(Zeta);
theta = zeros(nboot,nvar);

for i = 1:nboot
    ind = randperm(N);
    Zsub = Zeta(ind(1:nsub),:);
    [u s v] = svd(Zsub/sqrt(nsub),'econ');
    theta(i,:) = v(:,end)';
    theta(i,:) = theta(i,:)/theta(i,1);  % Normalize theta vector so that first coefficient is unity
end

% Find mean and std of theta elements

thetamean = mean(theta);
thetastd = std(theta);
thetal = thetamean - 2*thetastd;
thetau = thetamean + 2*thetastd;
ind = 1:1:16;

fprintf("\nThe 95% confidence intervals using bootstrapping are as follows for L=20:\n")
to_show = table(ind', thetal', thetau');
disp(to_show);
```

```
The 95    Var1      Var2        Var3
    ____    ________    ________
```

```
    1           1          1
    2    -0.88614   -0.28085
    3    -0.35811    -0.1413
    4    -0.01954   0.046789
    5     -1.1972    -1.1223
    6     0.79451     1.5234


 The 95    Var1      Var2       Var3
     ____    ________    _______

     1           1         1
     2     -12.036    14.004
     3     -22.898    23.196
     4      -18.56    17.671
     5     -12.535    11.266
     6     -6.6497    4.8388
     7     -18.683    19.271
     8     -5.1176    5.2988
     9    -0.42507   0.39247
    10     -1.9694    -0.362
    11     -16.202    14.926
    12     -22.795    23.314
    13     -11.935    13.353
    14     -12.658    13.961
    15     -8.1518    9.8859
    16      -25.16    23.785
```