ΤΕΛΙΚΗ ΕΡΓΑΣΙΑ ΣΥΣ411

Παυλόπουλος Χρήστος 2018030139

Α' Μέρος

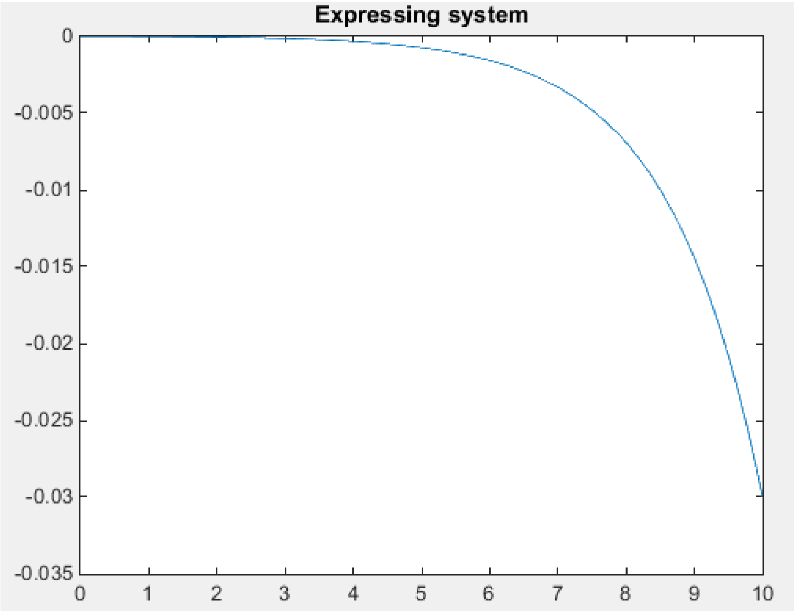
Α) Αρχικά, δημιουργήθηκαν οι πίνακες που μοντελοποιούν το σύστημα που μελετάμε και υπολογίστηκαν οι ιδιοτιμές και τα ιδιοδιανύσματα του πίνακα Α.

```
%printing eigenvalues with eig()
eigenvalues = eig(A)

%creating the eigenvectors
[V, D] = eig(A);

%each collumn of matrix V is an eigenvector 4
eigenvector1=V(:,1)
eigenvector2=V(:,2)
eigenvector3=V(:,3)
```

```
eigenvalues =

  -1.7036 + 0.0000i
   0.7310 + 0.0000i
  -0.0438 + 0.2066i
  -0.0438 - 0.2066i

eigenvector1 =

    0.9943
    0.0633
   -0.0740
    0.0434

eigenvector2 =

    0.9995
   -0.0142
   -0.0165
   -0.0226

eigenvector3 =

   1.0000 + 0.0000i
   0.0005 + 0.0003i
   0.0013 + 0.0002i
  -0.0003 - 0.0064i
```

Στην συνέχεια εκφράστηκε και σχεδιάστηκε η χρονική απόκριση του συστήματος για τις ζητούμενες αρχικές συνθήκες.

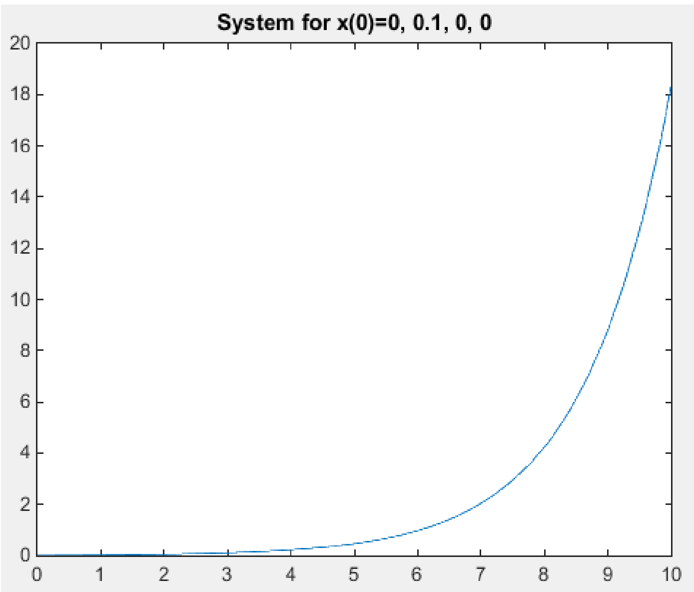
```
t = 0:0.01:10;          %time space
sys = ss(A,B,C,0);      %state space model
x0=[0.01 0 0 0];
u=zeros(size(t));
[Y,t,x] = lsim(sys,u,t,x0); %signal creation
figure()
plot(t,Y)
title('Expressing system')
```



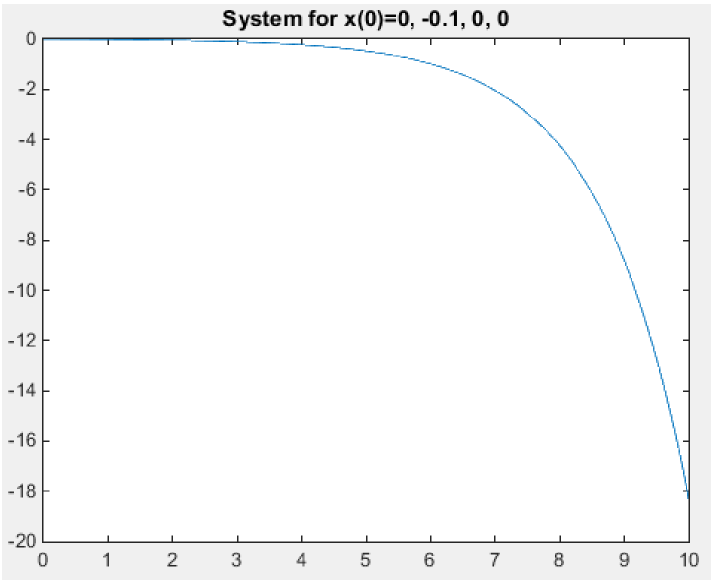
Δημιουργήθηκαν διανύσματα αρχικών συνθηκών x0() για τις δύο επιπλέον περιπτώσεις που ζητήθηκαν, βρέθηκε και σχεδιάστηκε η χρονική απόκριση του συστήματος για κάθε περίπτωση με την χρήση της lsim

```
t = 0:0.01:10;            %time space
sys = ss(A,B,C,0);        %state space model
x0_a=[0 0.01 0 0];
u=zeros(size(t));
[Y,t,x] = lsim(sys,u,t,x0_a); %signal creation
figure()
plot(t,Y)
title('System for x(0)=0, 0.1, 0, 0')

t = 0:0.01:10;            %time space
sys = ss(A,B,C,0);        %state space model
x0_b=[0 -0.01 0 0];
u=zeros(size(t));
[Y,t,x] = lsim(sys,u,t,x0_b); %signal creation
figure()
plot(t,Y)
title('System for x(0)=0, -0.1, 0, 0')
```

Για x(0) = [0 0.1 0 0]

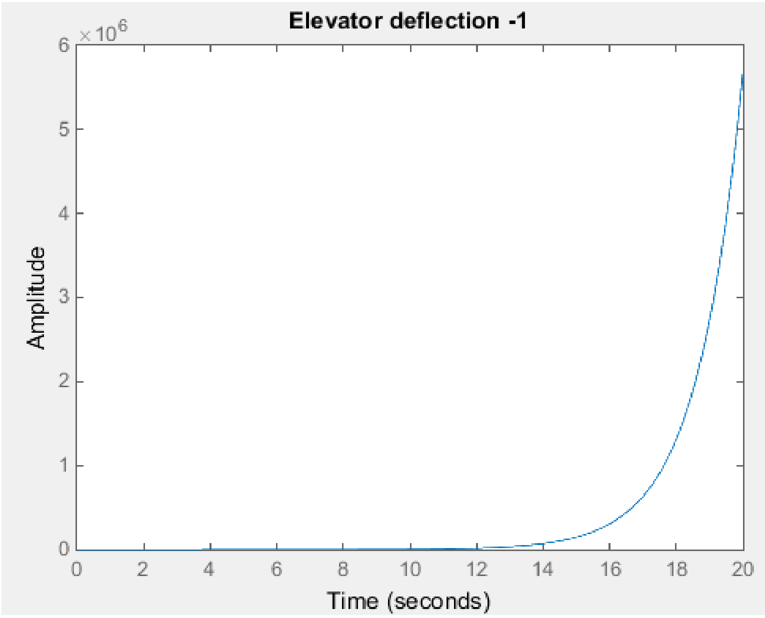


Για x(0) = [0   -0.1   0   0]



Β) Στο δεύτερο ερώτημα μας ζητήθηκε η αλλαγή του elevator deflection σε -1. Αυτή η αλλαγή προσομοιώνει την χειροκίνητη αλλαγή ύψους του αεροπλάνου από τον πιλότο. Αυτό το καταφέρνουμε με την μεταποίηση της εισόδου κατά την βηματική απόκριση ώστε να δείχνει την αλλαγή που προσπαθούμε να απεικονίσουμε.

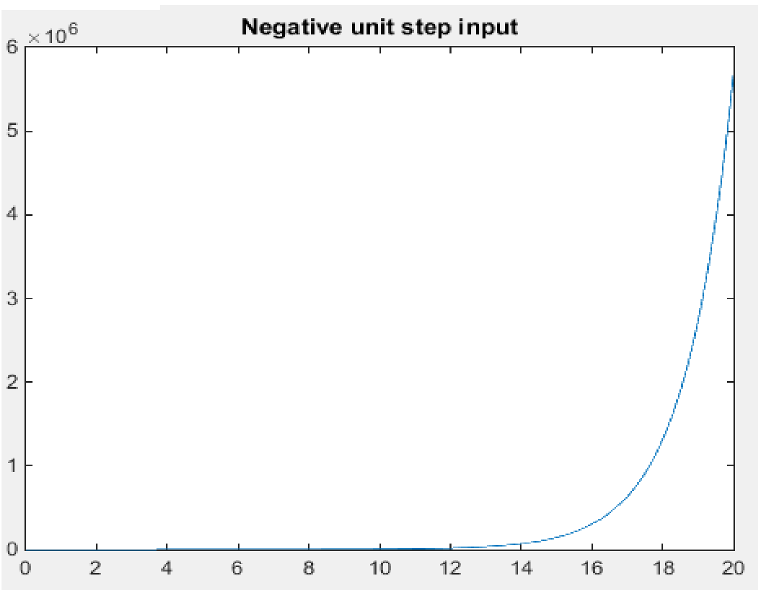
```
t = 0:0.01:20;          %time
sys = ss(A,B,C,0);      %state
figure()
step(-1*sys,t)          %addin
title('Elevator deflection -1'
```



Γ) Σε αυτό το ερώτημα απεικονίστηκε το σύστημα για μηδενικές αρχικές συνθήκες και αρνητική μοναδιαία είσοδο. Έτσι προσομοιώνεται η λειτουργία του elevator στην προσπάθεια ανύψωσης της μύτης του αεροπλάνου για την αύξηση του ύψους του.

```
t = 0:0.01:20;           %time space
u = -1.*ones(size(t));   %negative unit input
sys = ss(A,B,C,0);       %state space model
[Y,t,x] = lsim(sys,u,t); %signal creation
figure()
plot(t,Y)
title('Negative unit step input')
```

Παρατηρούμε μια μικρή διαφορά στην κλίση μεταξύ των δύο τελευταίων παραστάσεων. Η επανεκκίνηση του elevator (σ.σ. negative unit input) οδηγεί σε πιο γρήγορη μεταβολή.



Δ) Μελετήθηκε το σύστημα ως προς την ευστάθειά του. Αν ένα σύστημα έχει έστω και έναν πόλο στο δεξί ημιεπίπεδο των πραγματικών αριθμών δεν είναι ευσταθές. Παρατηρούμε πως ο πίνακας A έχει μια ιδιοτιμή με θετικό πραγματικό μέρος και οδηγούμαστε στο συμπέρασμα πως το σύστημά μας δεν είναι ευσταθές. Μεγάλη και διαρκής μεταβολή στο ύψος του αεροπλάνου θα το οδηγήσει σε ασταθή λειτουργία.

```
%%%%%%%
%%%D%%%
%%%%%%%

poles = eig(A)
poles =

  -1.7036 + 0.0000i
   0.7310 + 0.0000i
  -0.0438 + 0.2066i
  -0.0438 - 0.2066i
```

E) Για την ελεγξιμότητα ελέγχουμε τον βαθμό του πίνακα ελεγξιμότητας. Αν ο βαθμός αυτός είναι ίσος με τον βαθμό του πίνακα A τότε το σύστημα είναι ελέγξιμο. Οι μη-ελέγξιμες καταστάσεις του συστήματος ισούνται με μηδέν, άρα συμπεραίνουμε πως το σύστημα είναι ελέγξιμο.

```
%%%%%%%
%%%E%%%
%%%%%%%

ctrb(A,B)
Co=ctrb(sys);
unco=rank(A)-rank(Co)

unco =

     0
```

Z) Για την παρατηρησιμότητα ελέγχουμε τον βαθμό του πίνακα παρατηρησιμότητας. Αν ο βαθμός αυτός είναι ίσος με τον βαθμό του πίνακα A τότε το σύστημα είναι παρατηρήσιμο. Οι μη-παρατηρήσιμες καταστάσεις του συστήματος ισούνται με μηδέν, έτσι συμπεραίνουμε πως το σύστημα είναι παρατηρήσιμο.

```
%%%%%%%
%%%Z%%%
%%%%%%%

obsv(A,C)
Ob = obsv(sys);
unob=rank(A)-rank(Ob)
unob =

     0
```

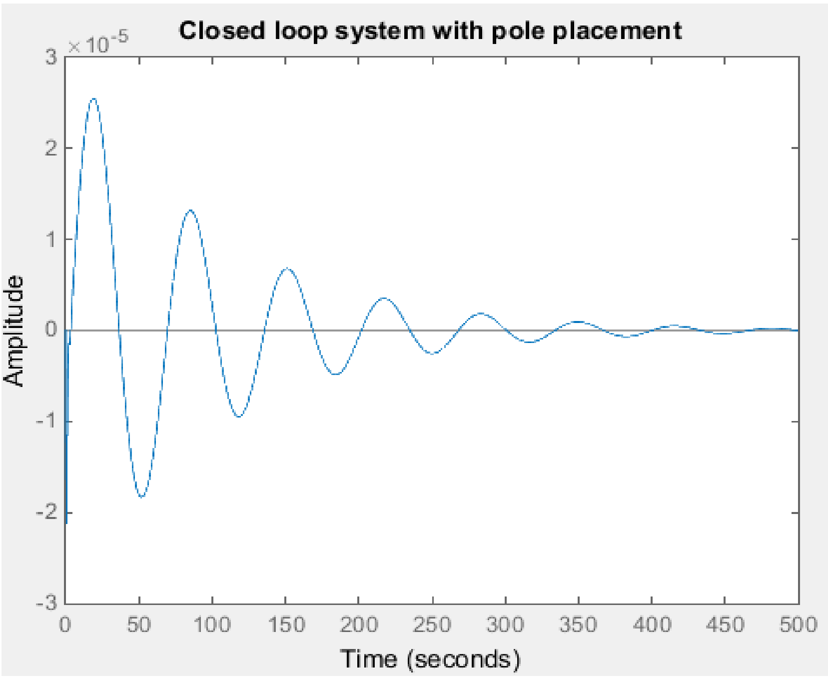
Β' Μέρος

Α) Στο πρώτο ερώτημα μας ζητήθηκε η σχεδίαση ενός ανατροφοδοτή ώστε οι ιδιοτιμές του κλειστού βρόγχου να είναι -1.25 ± j2.2651 και -0.01 ± j0.095. Με την εντολή place υπολογίσαμε το παρακάτω K

```
p1 = -1.25 + 2.2651i;
p2 = -1.25 - 2.2651i;
p3 = -0.01 + 0.095i;
p4 = -0.01 - 0.095i;

K = place(A,B,[p1 p2 p3 p4])
sys_cl = ss(A-B*K,B,C,0);
```

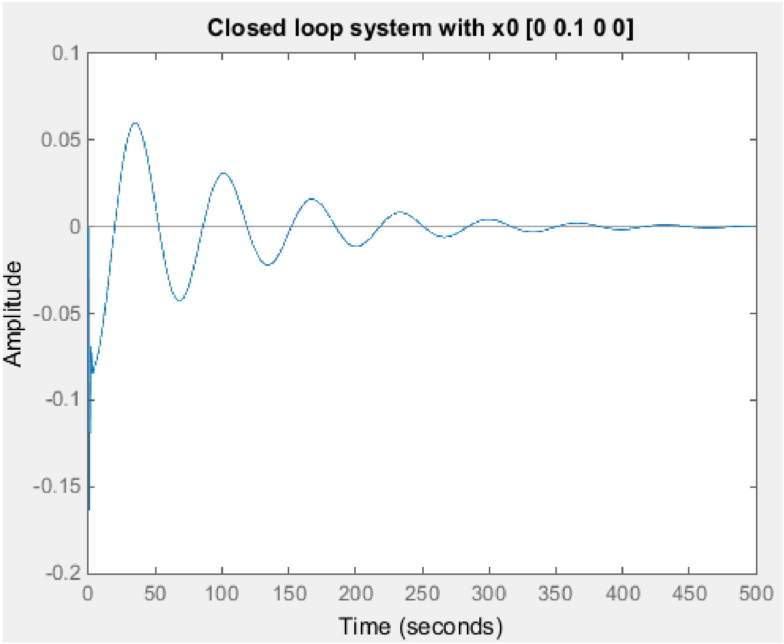
```
K =

   -0.0047   -4.1260   -0.7075   -0.1149
```

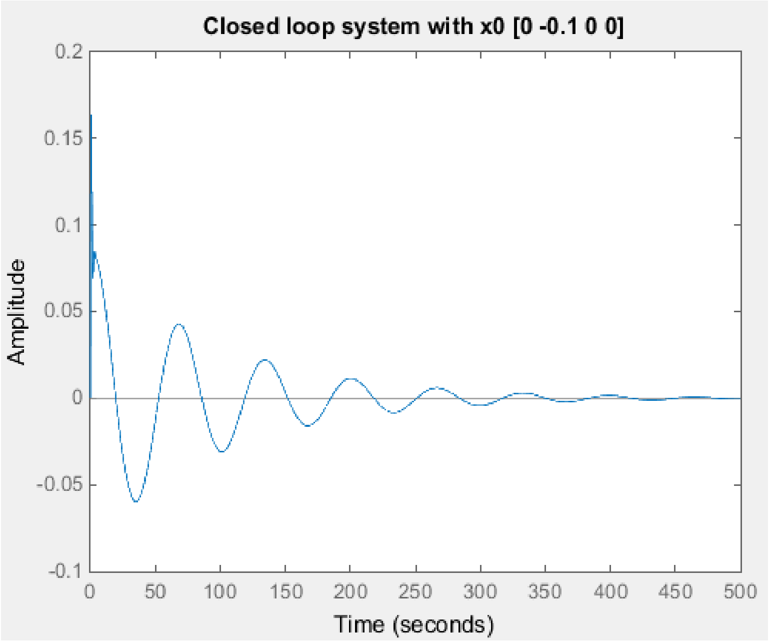
Παρακάτω απεικονίζεται η συμπεριφορά του συστήματος για τους δοσμένους πόλους



Β) Βρέθηκε και σχεδιάστηκε η χρονική απόκριση του κλειστού συστήματος για αρχικές συνθήκες x(0) = [0, 0.1, 0, 0] και x(0) = [0, -0.1, 0, 0]. Όπως και στην προηγούμενη άσκηση δημιουργήθηκαν δύο διανύσματα αρχικών συνθηκών και έγινε η σχεδίαση του κλειστού συστήματος.

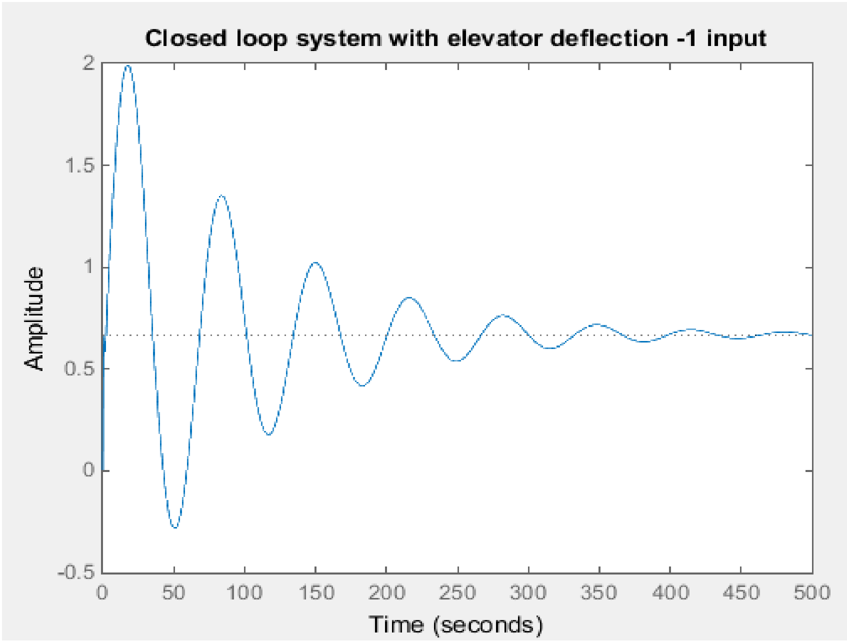
```
%% B

x0_a=[0 0.1 0 0];
r=zeros(size(t));
figure()
lsim(sys_cl,r,t,x0_a);
title('Closed loop system with x0 [0 0.1 0 0]');

x0_b=[0 -0.1 0 0];
r=zeros(size(t));
figure()
lsim(sys_cl,r,t,x0_b);
title('Closed loop system with x0 [0 -0.1 0 0]');
```



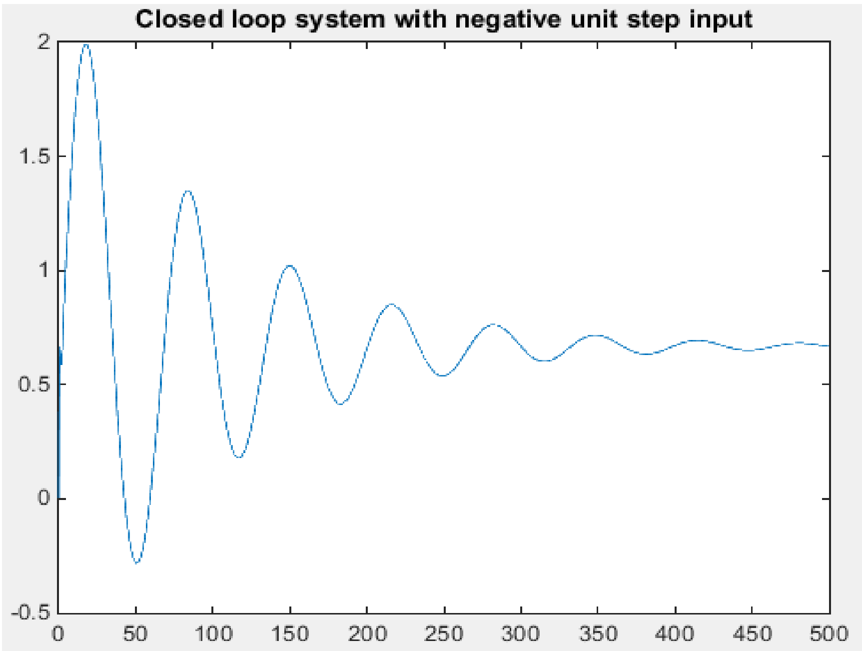


Γ) Βρέθηκε και σχεδιάστηκε η χρονική απόκριση του συστήματος για μηδενικές αρχικές συνθήκες και είσοδο elevator deflection -1 και αρνητική μοναδιαία είσοδο με παρόμοιο τρόπο όπως στην προηγούμενη άσκηση.

```
step(-1*sys_cl,t)
```



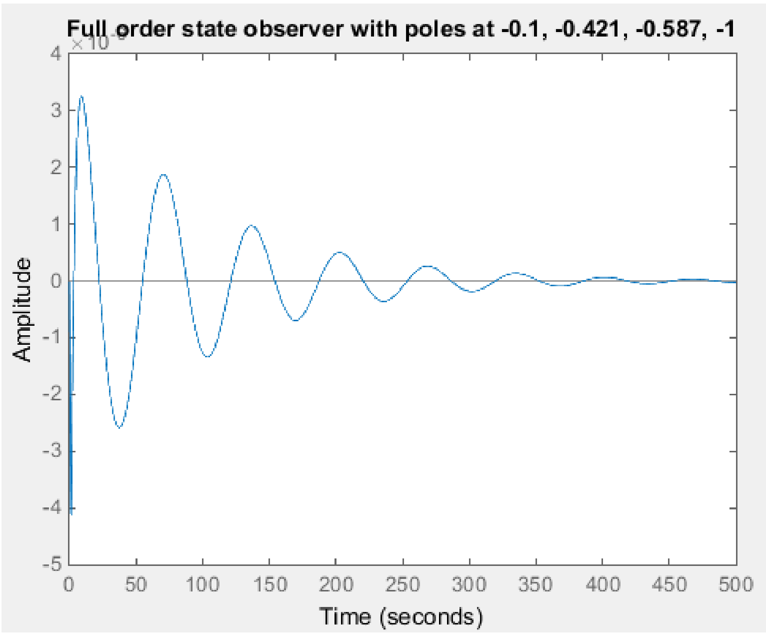

```
u2 = -1.*ones(size(t));    %negative unit input
[Y,t,x] = lsim(sys_cl,u2,t); %signal creation
```



Δ) Ο πίνακας C τέθηκε C = [0 0 1 0] και σχεδιάστηκε ένας παρατηρητής με επιθυμητούς πόλους τα σημεία −0.1, −0.421, −0.587, −1. Βρέθηκε το κατάλληλο L με το διάνυσμα των επιθυμητών πόλων και υπολογίστηκαν οι πίνακες A,B,C του παρατηρητή. Δημιουργήθηκε το σύστημα και προσομοιώθηκε.

```
L=place(A',C',[op1 op2 op3 op4])'

L =

 -179.7758
    1.3792
    1.0477
    1.4451
```

```
At = [ A-B*K             B*K
       zeros(size(A))    A-L*C ]

Bt = [    B*Nbar
       zeros(size(B)) ]

Ct = [ C     zeros(size(C)) ]

sys = ss(At,Bt,Ct,0);
```



Ε) Στην συνέχεια υπολογίστηκε και σχεδιάστηκε το σφάλμα εκτίμησης σύμφωνα με τον τύπο από την θεωρία για τις παρακάτω αρχικές συνθήκες:

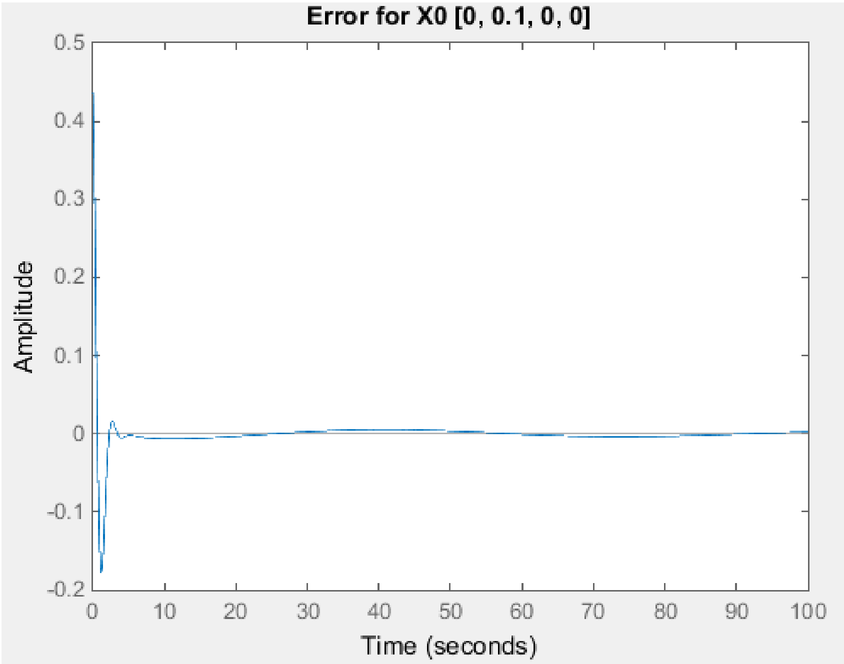
Therefore, the state estimate error dynamics are described by

$$\dot{\mathbf{e}} = (A - LC)\mathbf{e}$$

Αρχικά πραγματοποιήθηκε η αφαίρεση των δοσμένων αρχικών συνθηκών με των αναμενόμενων αρχικών συνθηκών. Ύστερα υπολογίστηκε το $\dot{e}$ και προσομοιώθηκε δείχνοντας την δυναμική του σφάλματος εκτίμησης.

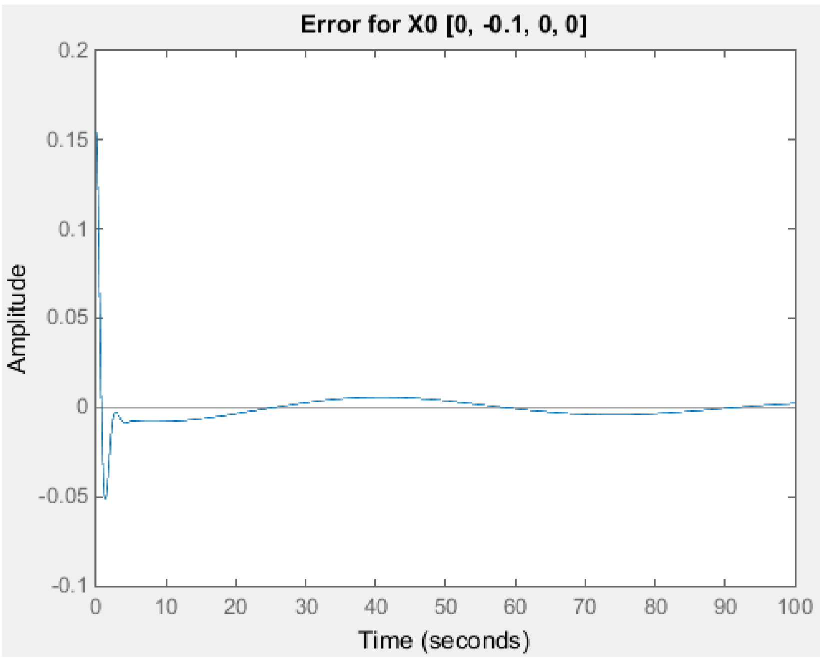
Για x(0) = (0, 0.1, 0, 0) και x^(0) = (0.2, −0.1, 0.1, −0.1)

```
x0_bar = [0.2 -0.1 0.1 -0.1];

et_a=(A-L*C)*e_a'


et_a =

  -21.9596
   -0.0651
    0.4375
    0.0445
```



Για x(0) = (0, -0.1, 0, 0) και x^(0) = (0.2, −0.1, 0.1, −0.1)

```
x0_bar = [0.2 -0.1 0.1 -0.1];
e_b=x0_b-x0_bar
et_b=(A-L*C)*e_b'


et_b =

  -21.1874
    0.0382
    0.1541
    0.0445
```



Z) Η ταχύτητα σύγκλισης καθορίζεται από τους πόλους του εκτιμητή σφάλματος. Επειδή θέλουμε να χρησιμοποιήσουμε τον εκτιμητή σφάλματος ως είσοδο στον ελεγκτή θέλουμε η κατάσταση να συγκλίνει γρηγορότερα από το κλειστό σύστημα. Έτσι θέλουμε οι πόλοι του παρατηρητή να είναι γρηγορότεροι από του ελεγκτή. Για να το κάνουμε αυτό στην δική μας περίπτωση θα πάρουμε τον μικρότερο πόλο (-0.1) και θα τον πολλαπλασιάσουμε με 6 (μπορούσαμε και με μεγαλύτερο αριθμό αλλά δεν θέλουμε φαινόμενα θορύβου). Αυτό και σε συνδυασμό με την εντολή place που μας απαγορεύει την τοποθέτηση ίδιων πόλων θα μας οδηγήσει την εντολή για τους ακόλουθους πόλους.

```
poles =

   -1.0000
   -0.5870
   -0.4210
   -0.1000
```

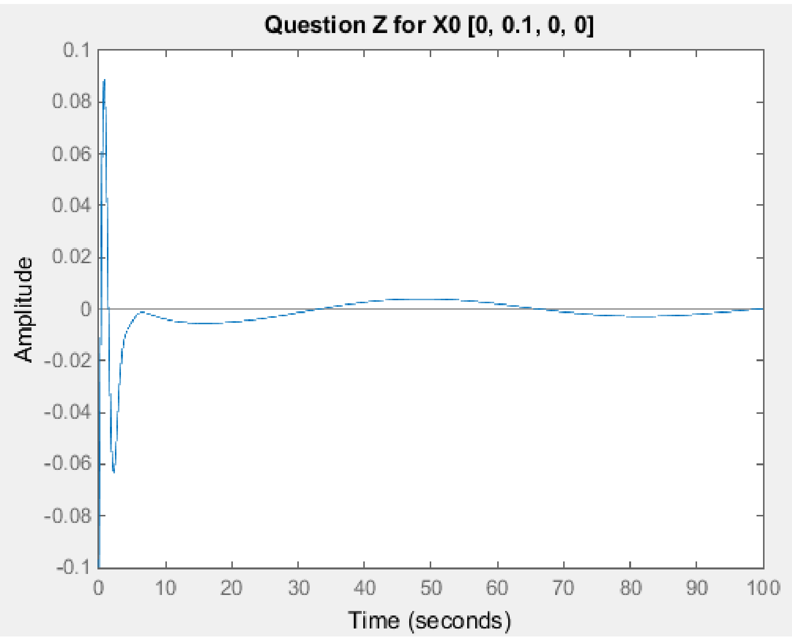
```
P =[-0.6 -0.7 -0.8 -0.9];
```

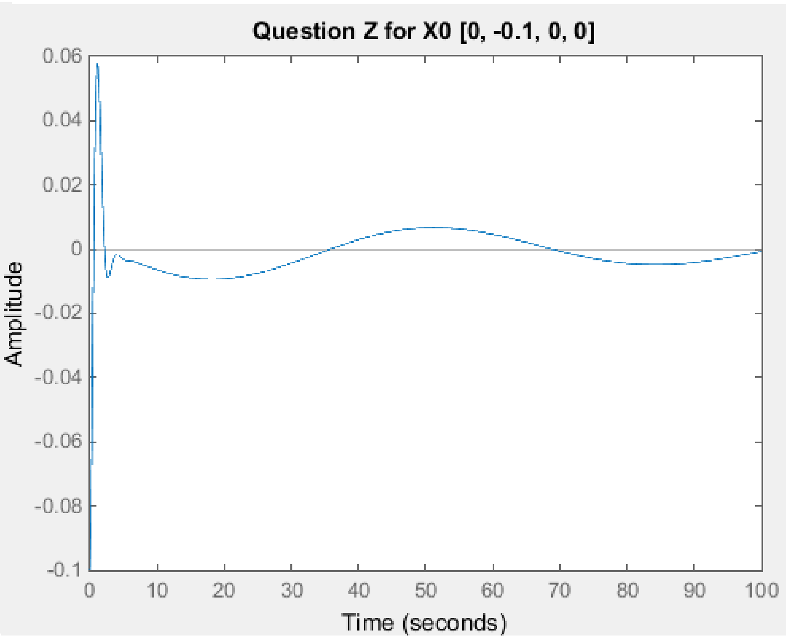
Υπολογίζουμε το νέο L (στοιχείο του παρατηρητή) και τους νέους πίνακες A,B,C για να σχεδιάσουμε τον ολοκληρωμένο ελεγκτή βασισμένο σε παρατηρητή. Αρχικές συνθήκες θέτονται οι διαφορές αρχικών συνθηκών με τις αναμενόμενες αρχικές συνθήκες.

```
L=place(A',C',P)'

 L =

  -858.8232
     2.2972
     1.9397
     6.4465
```

```
At2 = [ A-B*K              B*K
        zeros(size(A))     A-L*C ]

Bt2 = [     B*Nbar
        zeros(size(B)) ]

Ct2 = [ C     zeros(size(C)) ]

sys2 = ss(At2,Bt2,Ct2,0);

figure()
lsim(sys2,zeros(size(t)),t,[e_a e_a]);
title('Question Z for X0 [0, 0.1, 0, 0]')

figure()
lsim(sys2,zeros(size(t)),t,[e_b e_b]);
title('Question Z for X0 [0, -0.1, 0, 0]')
```

Παίρνουμε τις εξής χρονικές αποκρίσεις:





Η) Για την συνάρτηση μεταφοράς χρησιμοποιούμε τη σχέση από το βιβλίο του Antsaklis **y(s) = T(s)r(s) = (C + DF) * [sI - (A + BF)] -1 * B + D*r(s)** όπου y(s) η Laplace μετατροπή του y(t) και r(s) η Laplace μετατροπή του r(t). Ουσιαστικά πρόκειται για μια μετατροπή του γνωστού τύπου **y(s) = H(s) = C * [sI - A] -1 * B + D** συμπεριλαμβάνοντας τις μεταβλητές του κέρδους και reference input. Στην περίπτωσή μας το reference input δεν αλλάζει την τιμή της συνάρτησης μεταφοράς καθότι ο πίνακας D είναι ίσος με μηδέν. Ωστόσο αν ήταν διάφορος του μηδενός θα μπορούσαμε επηρεάζοντας το r(t) να αλλάξουμε την συνάρτηση μεταφοράς

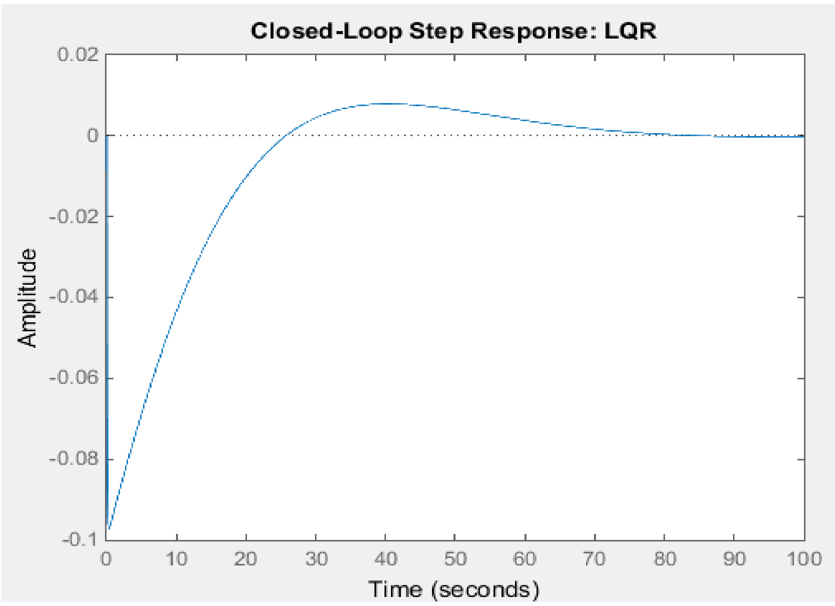
```
%% H
s = tf('s');
sia=(s*eye(8,8))-(At);
F=inv(sia);
tf=Ct*F*Bt
```

```
tf =

  2.461 s^7 + 7.749 s^6 + 20.58 s^5 + 10.99 s^4 + 0.8052 s^3
                            + 0.1041 s^2 + 0.00373 s - 4.323e-19
  -------------------------------------------------------------
  s^8 + 5.04 s^7 + 19.86 s^6 + 34.34 s^5 + 46.51 s^4
                  + 2.424 s^3 + 0.8493 s^2 + 0.01914 s + 0.00373
```

Θ) Ζητήθηκε ο σχεδιασμός του κέρδους με την τεχνική LQR. Έχουμε παραμέτρους state cost weighted matrix Q και control weighted matrix R. Τέθηκε για διευκόλυνση ο πίνακας R=1.

Ο πίνακας Q = p*C'*C περιέχει την μεταβλητή p η οποία και μεταβλήθηκε κατά την διάρκεια των δοκιμών. Για μικρές τιμές του p το σύστημα είχε μια πολύ αργή ανύψωση στην κυματομορφή. Κατέληξα στην τιμή p=100 καθώς το αποτέλεσμα για το pitch correction είναι επιθυμητό. Ωστόσο παραμένει ένα αργό σύστημα.

```
K =

    0.0010   -0.8827   -9.7962   -0.6662
```



Τέλος βρίσκουμε τους πόλους του κλειστού συστήματος.

Διαπιστώνουμε πως οι 3 από τους 4 πόλους στο ερώτημα αυτό έχουν μεγαλύτερο αρνητικό πραγματικό μέρος από τους πόλους του ερωτήματος Α. Θεωρούμε έτσι την συμπεριφορά του συστήματος με την μέθοδο LQR πιο ευσταθή.

```
P=pole(sys lqr)     p =

                     -16.5412 + 0.0000i
                      -0.5884 + 0.0000i
                      -0.0543 + 0.0525i
                      -0.0543 - 0.0525i
```

ΤΕΛΟΣ ΕΡΓΑΣΙΑΣ

Ακολουθεί ο κώδικας MATLAB

Άσκηση Α

```
clear all; close all; clc;

%creating matrices A,B,C,D
A = [-0.0507 -3.861 0 -32.2; -0.00117 -0.5164 1 0;
    -0.000129 1.4168 -0.4932 0; 0 0 1 0]
B = [0; -0.0717; -1.645; 0]
C = [0 0 1 1]
D = 0;

%%%%%%%
%%%A%%%
%%%%%%%

%printing eigenvalues with eig()
eigenvalues = eig(A)

%creating the eigenvectors
[V, D] = eig(A);

%each collumn of matrix V is an eigenvector 4
eigenvector1=V(:,1)
eigenvector2=V(:,2)
eigenvector3=V(:,3)


t = 0:0.01:10;          %time space
sys = ss(A,B,C,0);      %state space model
x0=[0.01 0 0 0];
u=zeros(size(t));
[Y,t,x] = lsim(sys,u,t,x0); %signal creation
figure()
plot(t,Y)
title('Expressing system')

t = 0:0.01:10;          %time space
sys = ss(A,B,C,0);      %state space model
x0_a=[0 0.01 0 0];
u=zeros(size(t));
[Y,t,x] = lsim(sys,u,t,x0_a); %signal creation
figure()
plot(t,Y)
title('System for x(0)=0, 0.1, 0, 0')

t = 0:0.01:10;          %time space
sys = ss(A,B,C,0);      %state space model
x0_b=[0 -0.01 0 0];
u=zeros(size(t));
[Y,t,x] = lsim(sys,u,t,x0_b); %signal creation
figure()
plot(t,Y)
title('System for x(0)=0, -0.1, 0, 0')
```

```
%%%%%%%
%%%B%%%
%%%%%%%

t = 0:0.01:20;          %time space
sys = ss(A,B,C,0);      %state space model
figure()
step(-1*sys,t)          %adding -1 elevator deflection as a feedback on
system
title('Elevator deflection -1')

%%%%%%%
%%%C%%%
%%%%%%%

t = 0:0.01:20;           %time space
u = -1.*ones(size(t));   %negative unit input
sys = ss(A,B,C,0);       %state space model
[Y,t,x] = lsim(sys,u,t); %signal creation
figure()
plot(t,Y)
title('Negative unit step input')

%%%%%%%
%%%D%%%
%%%%%%%

poles = eig(A)

%%%%%%%
%%%E%%%
%%%%%%%

ctrb(A,B)
Co=ctrb(sys);
unco=rank(A)-rank(Co)


%%%%%%%
%%%Z%%%
%%%%%%%

obsv(A,C)
Ob = obsv(sys);
unob=rank(A)-rank(Ob)
```

Άσκηση Β

```
clear all; close all; clc;

%creating matrices A,B,C,D
A = [-0.0507 -3.861 0 -32.2; -0.00117 -0.5164 1 0;
    -0.000129 1.4168 -0.4932 0; 0 0 1 0]
B = [0; -0.0717; -1.645; 0]
C = [0 0 1 1]
D = 0;

sys = ss(A,B,C,0);

co = ctrb(A,B);

Controllability = rank(co)


%% A
t = 0:0.01:500;          %time space
x0=[0.01 0 0 0];
r=zeros(size(t));

p1 = -1.25 + 2.2651i;
p2 = -1.25 - 2.2651i;
p3 = -0.01 + 0.095i;
p4 = -0.01 - 0.095i;

K = place(A,B,[p1 p2 p3 p4])
sys_cl = ss(A-B*K,B,C,0);

Nbar=rscale(sys,K)

figure()
lsim(sys_cl,r,t,x0);
title('Closed loop system with pole placement');

%% B

x0_a=[0 0.1 0 0];
r=zeros(size(t));
figure()
lsim(sys_cl,r,t,x0_a);
title('Closed loop system with x0 [0 0.1 0 0]');

x0_b=[0 -0.1 0 0];
r=zeros(size(t));
figure()
lsim(sys_cl,r,t,x0_b);
title('Closed loop system with x0 [0 -0.1 0 0]');

%% Ca

t = 0:0.01:500;          %time space
figure()
step(-1*sys_cl,t)          %adding -1 elevator deflection as a feedback on
system
title('Closed loop system with elevator deflection -1 input')
```

```matlab
%% Cb

t = 0:0.01:500;            %time space
u2 = -1.*ones(size(t));    %negative unit input
[Y,t,x] = lsim(sys_cl,u2,t); %signal creation
figure()
plot(t,Y)
title('Closed loop system with negative unit step input')

%% D

C = [0 0 1 0]

%desired poles
op1=-0.1;
op2=-0.421;
op3=-0.587;
op4=-1;

%calculation of L
L=place(A',C',[op1 op2 op3 op4])'

At = [ A-B*K              B*K
       zeros(size(A))     A-L*C ]

Bt = [    B*Nbar
       zeros(size(B)) ]

Ct = [ C    zeros(size(C)) ]

sys = ss(At,Bt,Ct,0);

figure()
lsim(sys,zeros(size(t)),t,[x0 x0]);
title('Full order state observer with poles at -0.1, -0.421, -0.587, -1')

%% E

poles=eig(A-L*C)
x0_bar = [0.2 -0.1 0.1 -0.1];
e_a=x0_a-x0_bar
e_b=x0_b-x0_bar

t = 0:0.01:100;           %time space
r=zeros(size(t));

et_a=(A-L*C)*e_a'
et_b=(A-L*C)*e_b'

figure()
lsim(sys,r,t,[et_a' et_a']);
title('Error for X0 [0, 0.1, 0, 0]');

figure()
lsim(sys,r,t,[et_b' et_b']);
title('Error for X0 [0, -0.1, 0, 0]');
```

```
%% Z

P =[-0.6 -0.7 -0.8 -0.9];

L=place(A',C',P)'

At2 = [ A-B*K              B*K
       zeros(size(A))    A-L*C ]

Bt2 = [    B*Nbar
       zeros(size(B)) ]

Ct2 = [ C     zeros(size(C)) ]

sys2 = ss(At2,Bt2,Ct2,0);

figure()
lsim(sys2,zeros(size(t)),t,[e_a' e_a']);
title('Question Z for X0 [0, 0.1, 0, 0]')

figure()
lsim(sys2,zeros(size(t)),t,[e_b' e_b']);
title('Question Z for X0 [0, -0.1, 0, 0]')

%% H
s = tf('s');
sia=(s*eye(8,8))-(At);
F=inv(sia);
tf=Ct*F*Bt


%% Th

p=100;
Q=(p*C')*C
R=1;
[K]=lqr(A,B,Q,R)

sys_lqr=ss(A-B*K, B, C, D);

figure()
step(sys_lqr)
title('Closed-Loop Step Response: LQR');

P=pole(sys_lqr)
```