現代控制理論報告

104303206 黃筱晴

[新增內容]

1. Q2 Lyap.控制器

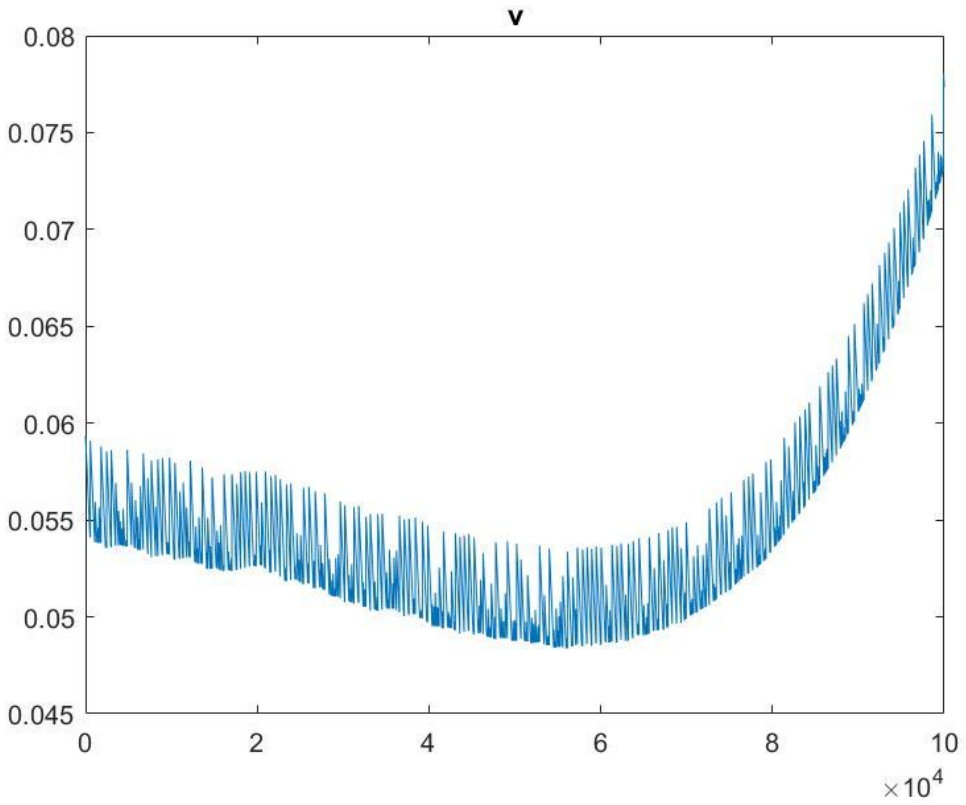
重新設計 $v(x)=0.5*(x_1^2+x_2^2+x_3^2)$，恆≧0。

則 $v'=x_1\,x_1'+x_2\,x_2'+x_3\,x_3'\equiv\alpha(x)+\beta(x)*u$

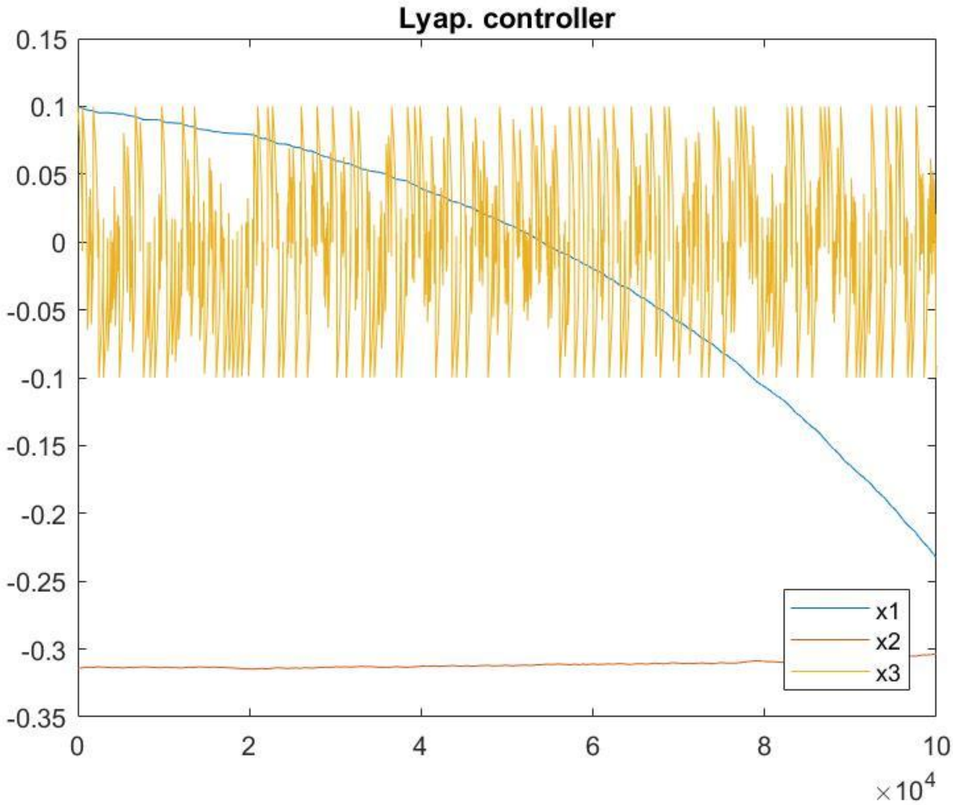
設計 $u=(\alpha(x)+1)/\beta(x)$使 v'恆等於-1。

因為當 u(x)分母趨近零時會導致發散，所以加入判斷式讓 u 達飽和，也因此造成圖中的不連續的部分。

```
if abs(u(i))>10000
    u(i)=10000*sign(u(i));
    fprintf('DANGER !!! i=%f\n',i);
end
```



前半段在鬆開 u 後還勉強控的回來，v 大致以固定斜率遞減，但是後來鬆開太多次後就控不下來了，v 越來越大，最後系統還是發散。



Lyap. controller
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
-0.25
-0.3
-0.35
0
2
4
6
8
10
×10^4
x1
x2
x3

Q2 FB linearization 控制器

$$\text{設計 } Z'=\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ a1 & a2 & a3 \end{bmatrix} Z$$

```
pole1=[1 8];pole2=[1 6];pole3=[1 7];
char_poly=conv(pole1,conv(pole2,pole3));%(s+p1)*(s+p2)*(s+p3)=0
```

設計 pole 位置:-6,-7,-8 ，

由程式算出 a1=- char_poly (4)、a2= - char_poly (3)、a3=- char_poly (2)

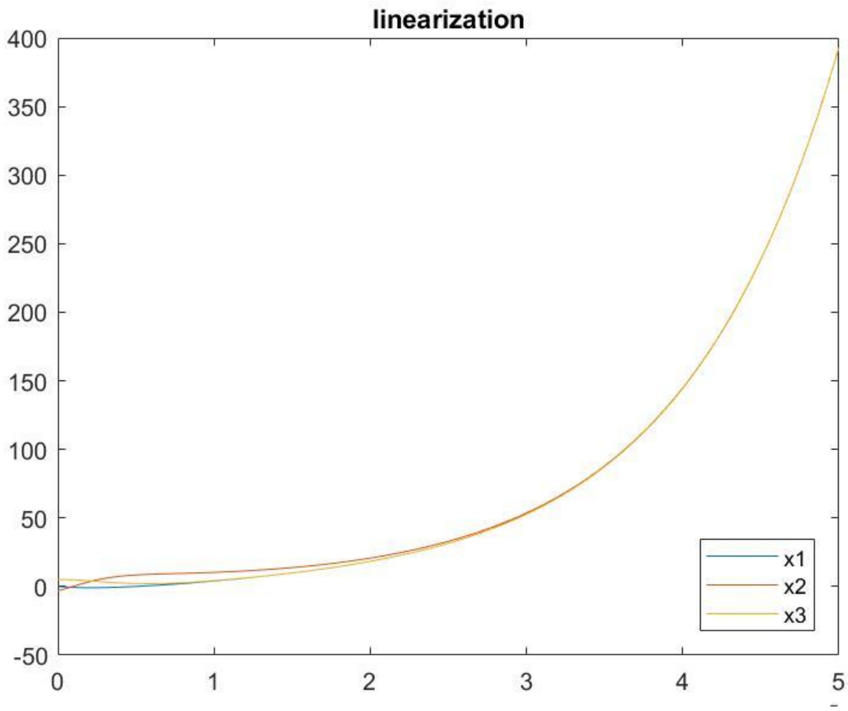
令 $z_1=x_1-x_3$；$z_2=z_1'$；$z_3=z_2'$

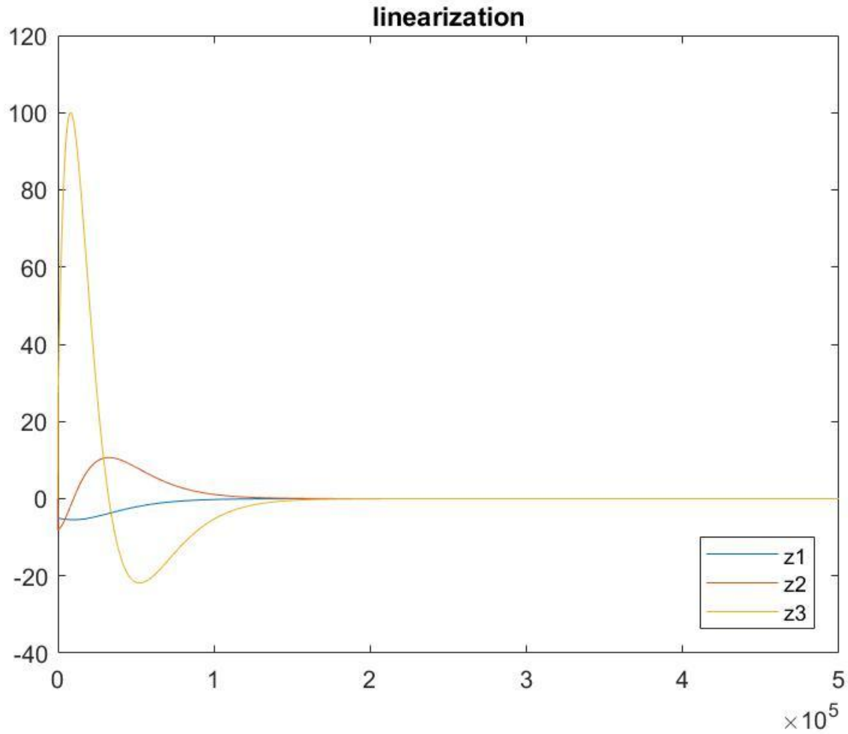
使用 z3'來設計控制器，將題目給的系統帶入計算後得

$z_3'=\sin(x_1-x_3)+u+2z_1z_2+z_3-u''$，設計 u''使 $z_3'=a_1z_1+a_2z_2+a_3z_3$

```
tmp=-char_poly(4)*z1(i)-char_poly(3)*z2(i)-char_poly(2)*z3(i);
u_dot_dot(i)=sin(x1-x3)+u(i-2)+2*z1(i)*z2(i)+z3(i)-tmp;
```

這裡步數好像會有點問題(u_dot_dot(i), u_dot(i-1), u(i-2)?)，所以把數值逼近的delta調到很小(delta=0.00001)讓這個問題不要造成太大影響。





z 成功以線性系統的方式收斂下來。但是因為 z 設計的關係，最後 x 是跑到 (x1-x3)=0 的平面上，然後 x2 沒有控到...

程式碼

```
%linearization
clc;clear;
delta=0.00001;
totalTime=5;
totalStep=totalTime/delta;

pole1=[1 8];pole2=[1 6];pole3=[1 7];
char_poly=conv(pole1,conv(pole2,pole3));%(s+p1)*(s+p2)*(s+p3)=0

x1array=[1:totalStep]*0;x2array=x1array;x3array=x1array;
u=x1array;u_dot=x1array;u_dot_dot=x1array;
z1=x1array;z2=x1array;z3=x1array;
x1_dot=x1array;x2_dot=x1array;x2_dot=x1array;
x1array(1)=0;x2array(1)=-pi;x3array(1)=5;%init condition
x1array(2)=0;x2array(2)=-pi;x3array(2)=5;%init condition
```

```matlab
x1array(3)=0;x2array(3)=-pi;x3array(3)=5;%init condition
for i=3:totalStep
    x1=x1array(i);x2=x2array(i);x3=x3array(i);
    x1_dot(i)=x2+x1-x3+sin(x1-x3);
    x2_dot(i)=x3+(x1-x3)^2;
    x3_dot(i)=sin(x1-x3)+u(i-2);

    z1(i)=x1-x3;
    z2(i)=x1_dot(i)-x3_dot(i);
    z3(i)=x3+z1(i)^2+z2(i)-u_dot(i-1);

    tmp=-char_poly(4)*z1(i)-char_poly(3)*z2(i)-char_poly(2)*z3(i);
    u_dot_dot(i)=sin(x1-x3)+u(i-2)+2*z1(i)*z2(i)+z3(i)-tmp;

    x1array(i+1)=x1+x1_dot(i)*delta;
    x2array(i+1)=x2+x2_dot(i)*delta;
    x3array(i+1)=x3+x3_dot(i)*delta;
    u_dot(i)=u_dot(i-1)+u_dot_dot(i)*delta;
    u(i-1)=u(i-2)+u_dot(i-1)*delta;
end

figure(1);
plot(x1array);hold on;
plot(x2array);hold on;
plot(x3array);legend('x1','x2','x3','location','southeast');
title('linearization');
figure(2);
plot(z1);hold on;
plot(z2);hold on;
plot(z3);hold on;legend('z1','z2','z3','location','southeast');
title('linearization');
figure(3);
plot(x1array-x3array);hold on;legend('x1-x3','location','southeast');
title('linearization');

figure(4);
time=1:totalStep-2;
xt=z1(time);yt=z2(time);zt=z3(time);
```

```
plot3(xt,yt,zt);hold on;
grid on;title('z1 z2 z3 phase portrait');
```

Q3 MRAC

因為MRAC控制器快追到modle之後過久了還是會發散，所以嘗試：

1. 500筆後就不再重新設計u。結果它很快就發散了。

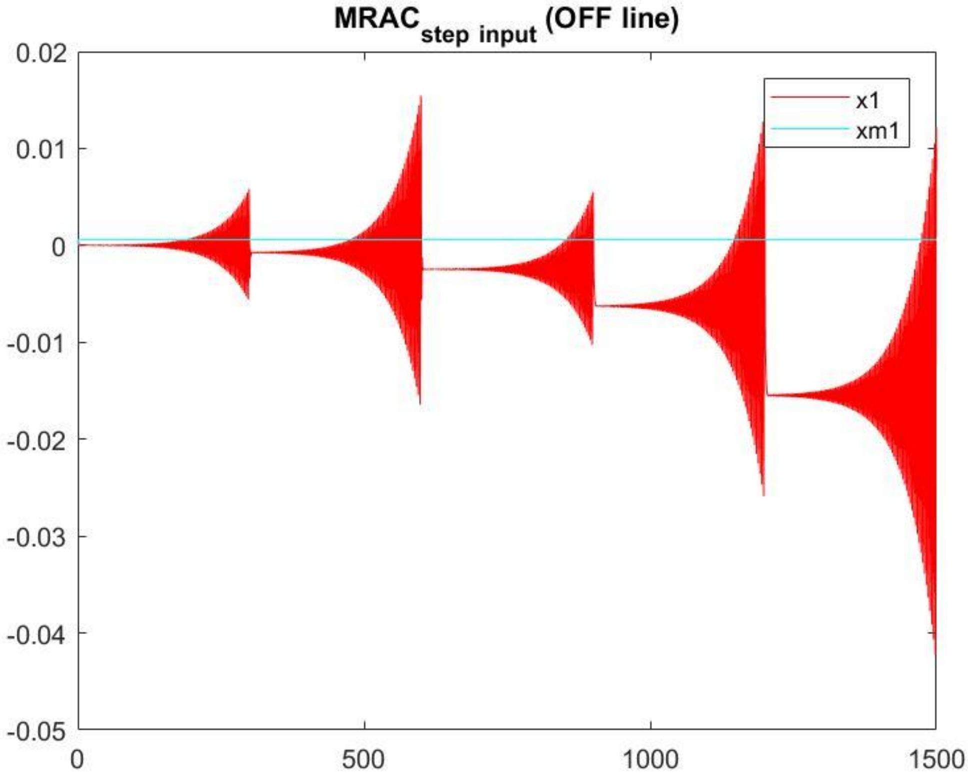
2.Offline操作，第500步資料後，每30000步中只有前500步更新控制器。

```
if k<=500
    u(k)=theta0(k)*r(k)+theta1(k)*x1(k)+theta2(k)*x2(k)+theta3(k)*x3(
k);
end
if k>500
    if mod(k,30000)<500
    u(k)=theta0(k)*r(k)+theta1(k)*x1(k)+theta2(k)*x2(k)+theta3(k)*x3(
k);
    else
        u(k)=u(k-1);
    end
end
```

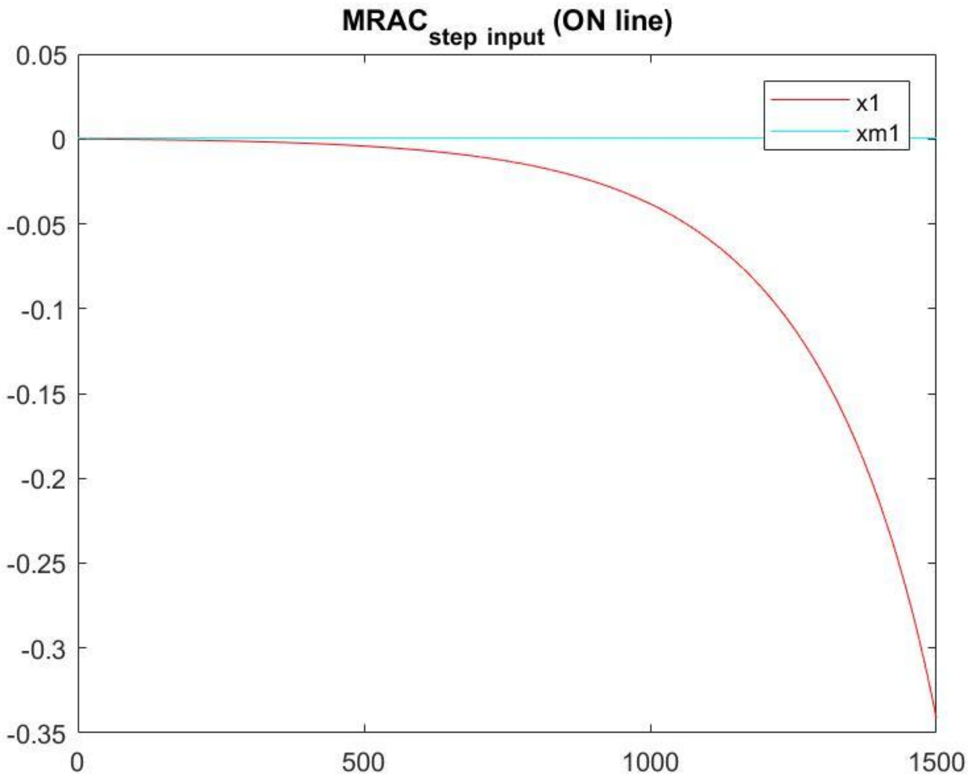
totaltime=1500;delta=0.01;

一直 Online 的 MRAC 控制器，1500 秒時，發散程度大概 |0.03|。

Offline 的大概|0.004|。性能有比較好一點。



MRAC$_{\text{step input}}$ (OFF line)
x1
xm1
0.02
0.01
0
-0.01
-0.02
-0.03
-0.04
-0.05
0
500
1000
1500



MRAC$_{\text{step input}}$ (ON line)
x1
xm1
0.05
0
-0.05
-0.1
-0.15
-0.2
-0.25
-0.3
-0.35
0
500
1000
1500

1.

考慮三階系統 $T(s)=\frac{Y(s)}{U(s)}=(s^3+a_1s^2+a_2s+a_3)^{-1}$，極點為使$(s^3+a_1s^2+a_2s+a_3)=0$ 的 s。

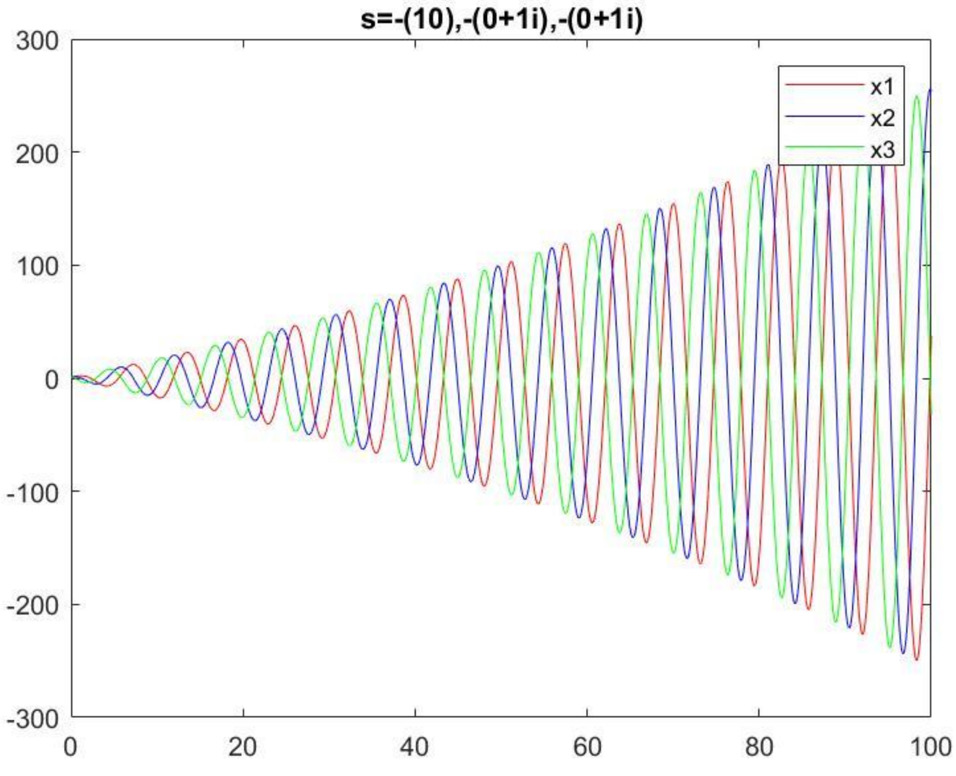
則此系統的微分方程式為：$y'''+a_1y''+a_2y'+a_3y=u$

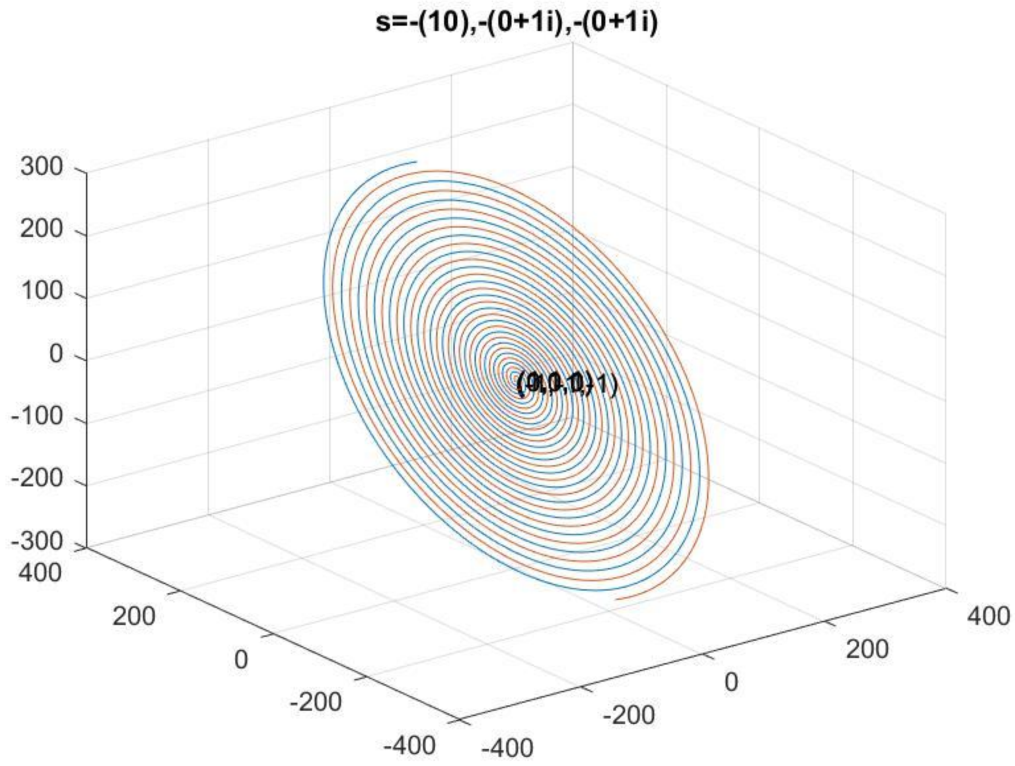
令 $x_1=y$；$x_2=x_1'=y'$；$x_3=x_2'=y''$；

則 $x_3'=y'''=-a_1y''-a_2y'-a_3y+u=-a_3x_1-a_2x_2-a_1x_3+u$

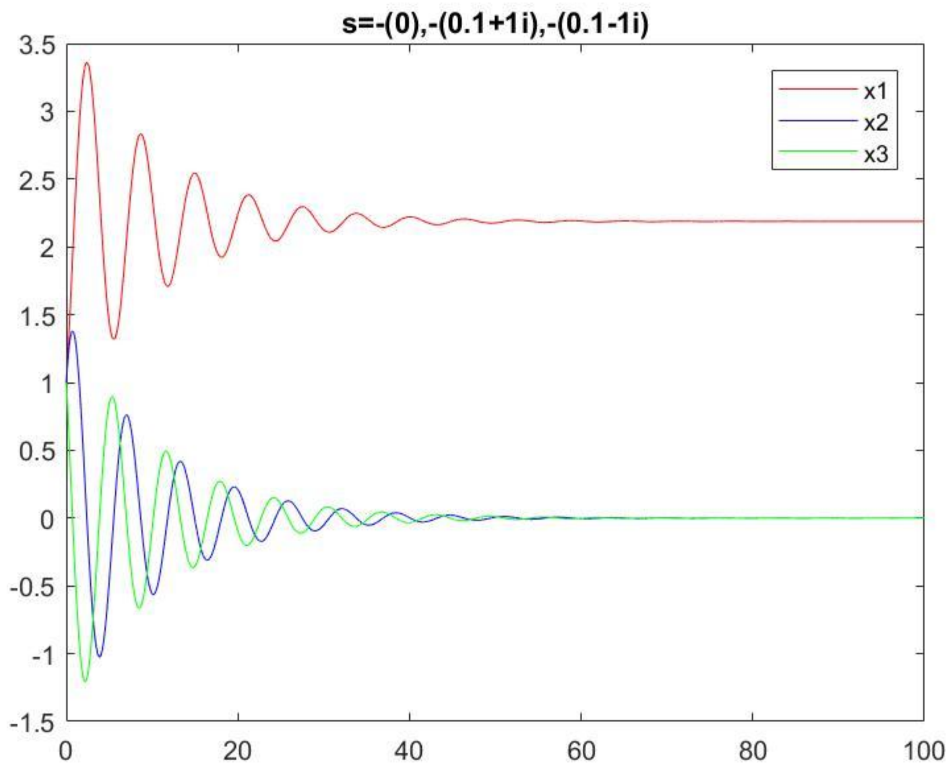
u=0 則有 $\begin{bmatrix} x1' \\ x2' \\ x3' \end{bmatrix}=\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a3 & -a2 & -a1 \end{bmatrix}\begin{bmatrix} x1 \\ x2 \\ x3 \end{bmatrix}$

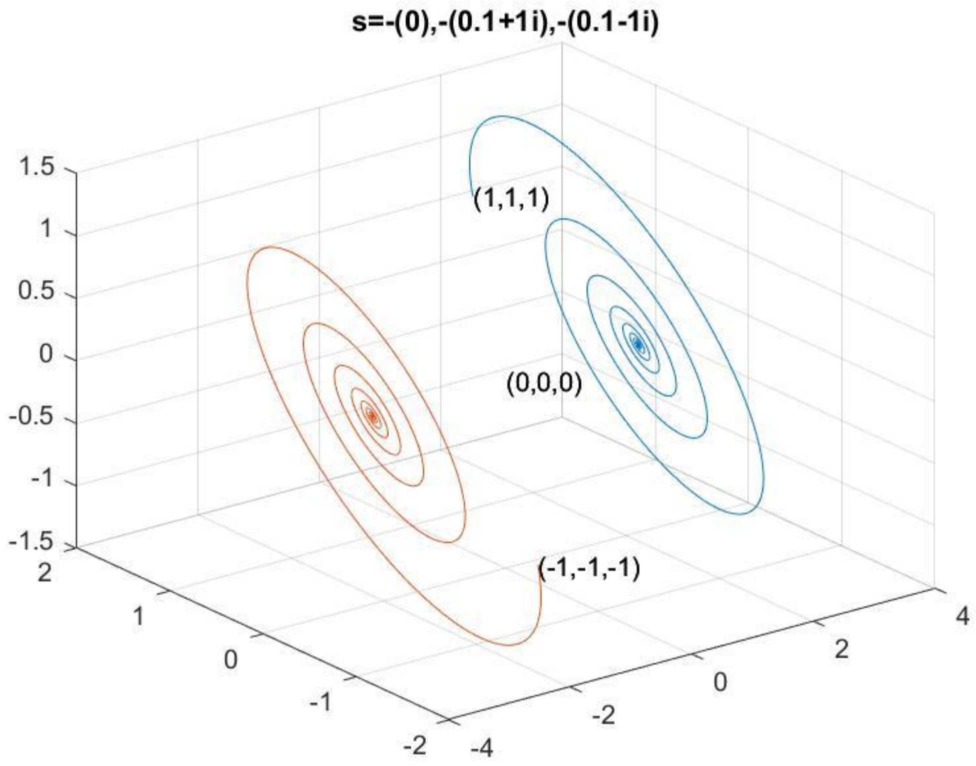
(1)虛軸上有≧2 極點：系統發散



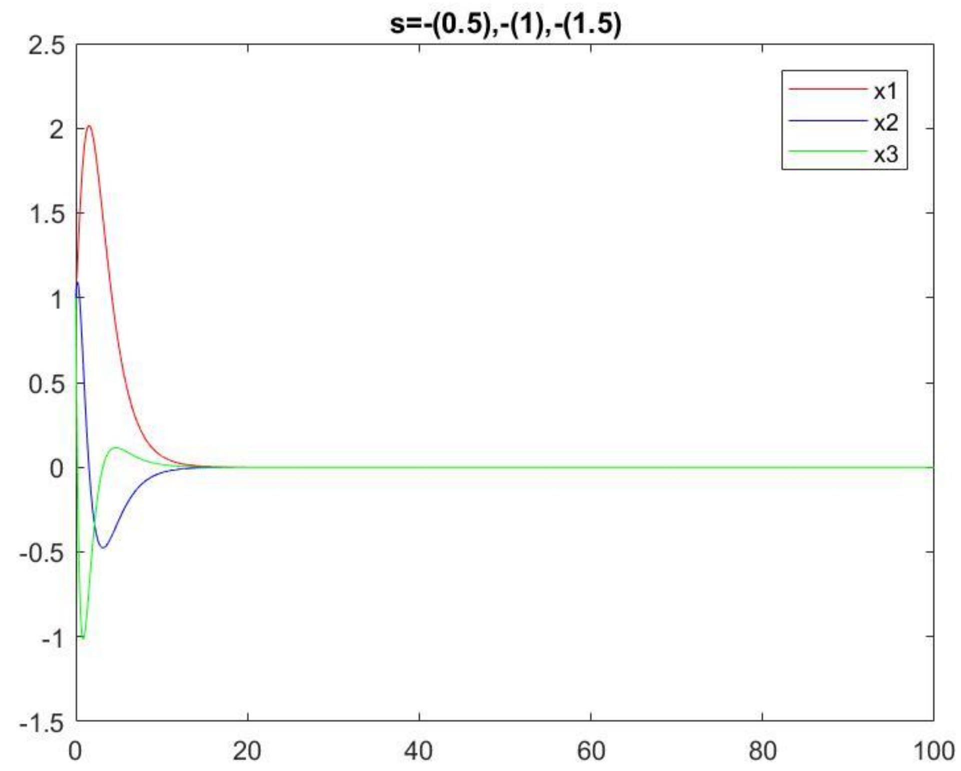

s=-(10),-(0+1i),-(0+1i)
300
200
100
0
-100
-200
-300
400
200
0
-200
-400
-400
-200
0
200
400
(0,0,0,1)

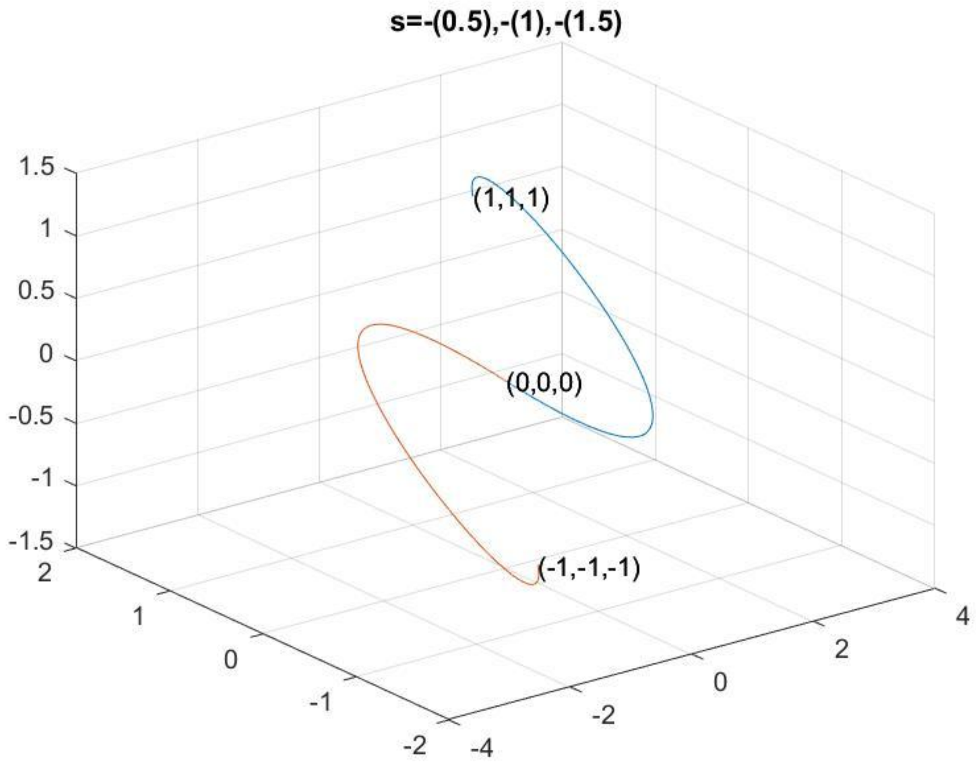
(2)虛軸上有 1 極點


s=-(0),-(0.1+1i),-(0.1-1i)
x1
x2
x3
3.5
3
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
0
20
40
60
80
100

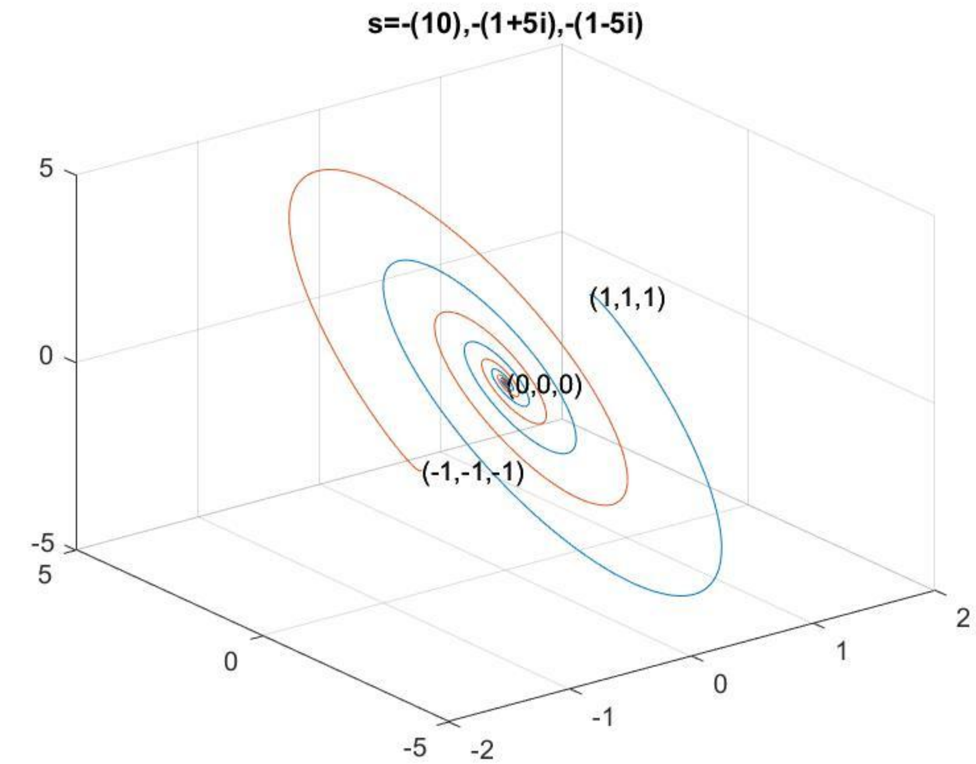


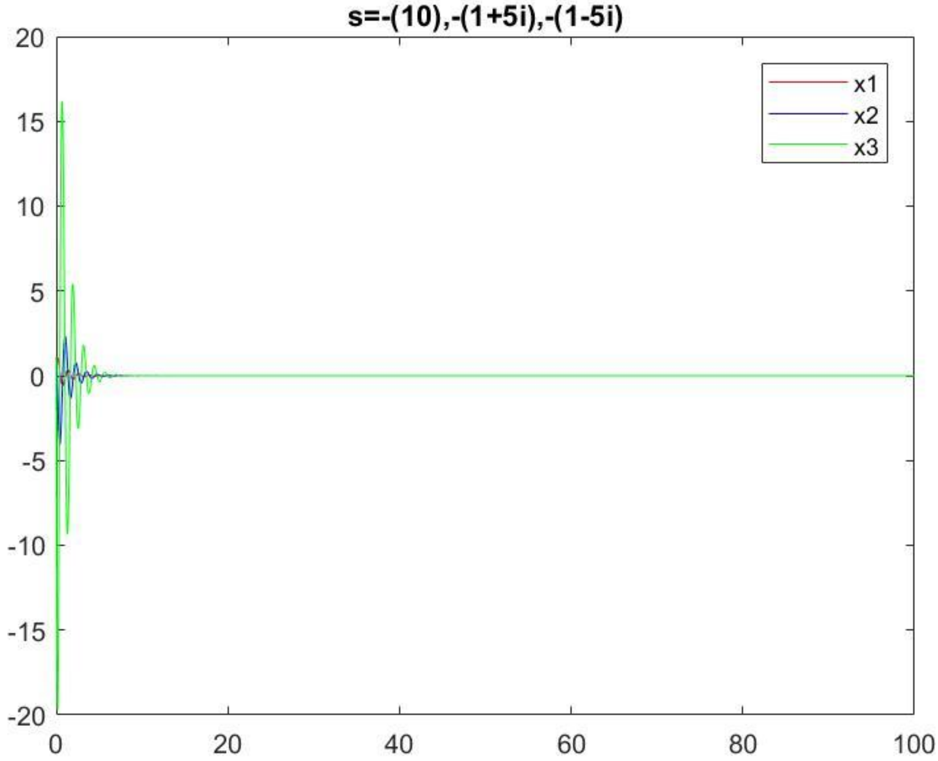
(3)3 實根



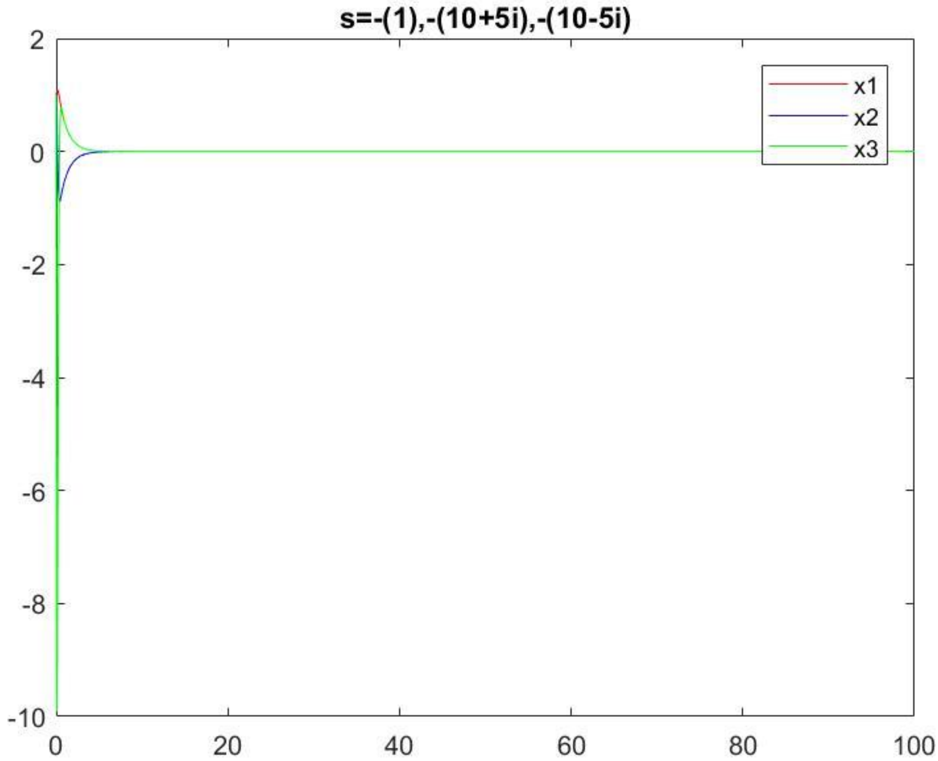


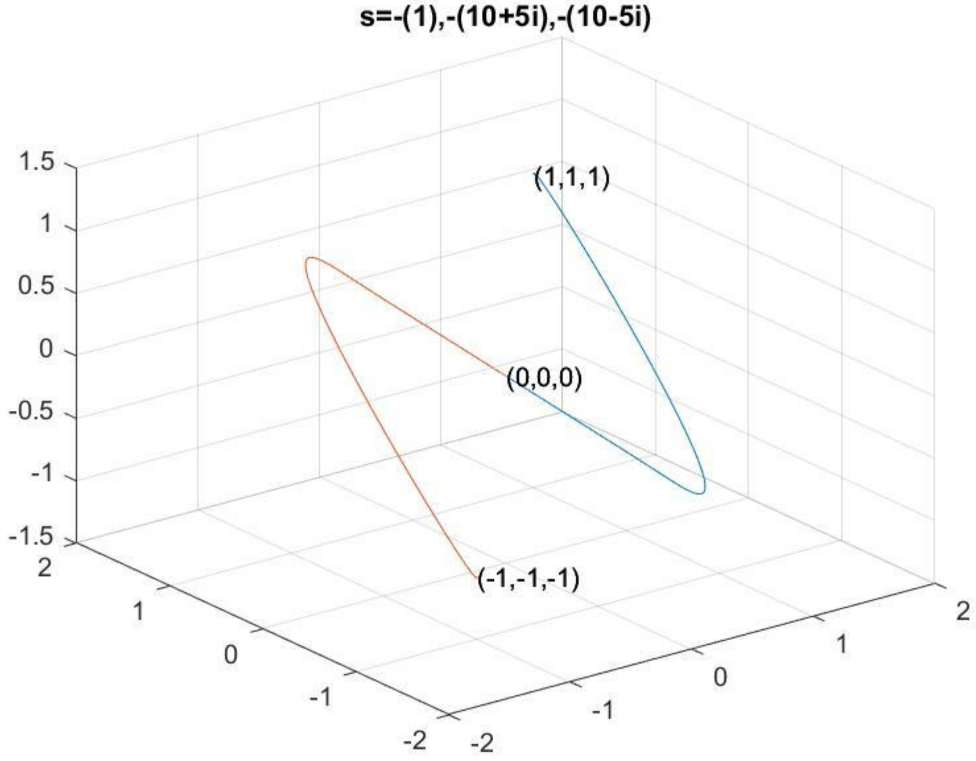
(4)共軛虛根為主極點




s=-(10),-(1+5i),-(1-5i)
x1
x2
x3
20
15
10
5
0
-5
-10
-15
-20
0
20
40
60
80
100

(5)共軛虛根不是主極點


s=-(1),-(10+5i),-(10-5i)
x1
x2
x3
2
0
-2
-4
-6
-8
-10
0
20
40
60
80
100



程式碼

```
clear;clc;
totaltime=100;
delta=0.01;
totalstep=totaltime/delta;

pole1=[1 0];pole2=[1 0.1-2i];pole3=[1 0.1+2i];
char_poly=conv(pole1,conv(pole2,pole3));%(s+p1)*(s+p2)*(s+p3)=0
A=[0 1 0;0 0 1;-char_poly(4) -char_poly(3) -char_poly(2)];%x_dot=A*x

IC=[1,1,1;-1,-1,-1];%initial condition
for i=1:2
    x1=[1:totalstep]*0;x2=x1;x3=x1;
    x1_dot=x1;x2_dot=x1;x3_dot=x1;
    x1(1)=IC(i,1);
    x2(1)=IC(i,2);
    x3(1)=IC(i,3);
```

```
    fprintf('init. condidtion:(%d,%d,%d)\n',x1(1),x2(1),x3(1));
    for k=1:totalstep
        x1_dot(k)=x2(k);
        x2_dot(k)=x3(k);
        x3_dot(k)=A(3,1)*x1(k)+A(3,2)*x2(k)+A(3,3)*x3(k);
        x1(k+1)=x1(k)+x1_dot(k)*delta;
        x2(k+1)=x2(k)+x2_dot(k)*delta;
        x3(k+1)=x3(k)+x3_dot(k)*delta;
    end

    if mod(i,2)==1
        figure(1);
        plot([0:1:totalstep]*delta,x1,'r');hold on;
        plot([0:1:totalstep]*delta,x2,'b');hold on;
        plot([0:1:totalstep]*delta,x3,'g');hold on;
        legend('x1','x2','x3');

str1=num2str(pole1(2));str2=num2str(pole2(2));str3=num2str(pole3(2));
        str=['s=-(' str1 '),-(' str2 '),-(' str3 ')'];
        title(str);
    end

    figure(2);
    time=1:totalstep-1;
    xt=x1(time);yt=x2(time);zt=x2(time);
    plot3(xt,yt,zt);hold on;

text(0,0,0,'(0,0,0)');text(1,1,1,'(1,1,1)');text(-1,-1,-1,'(-1,-1,-1)
');
    grid on;
end

title(str);
fprintf(str);
```

2.

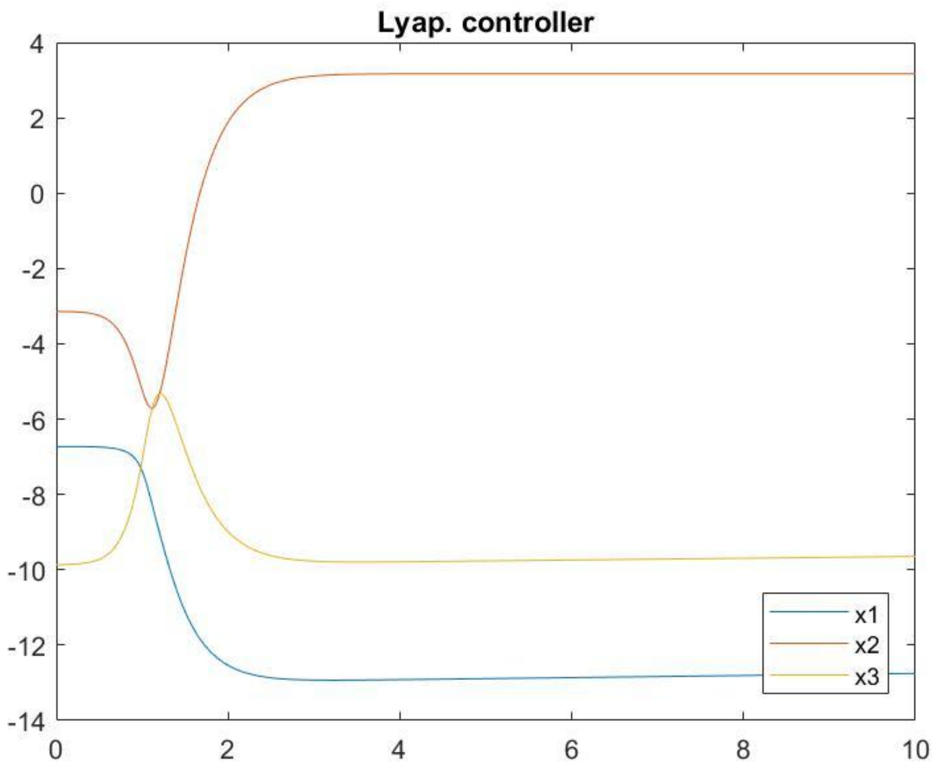
(1)Lyap. controller

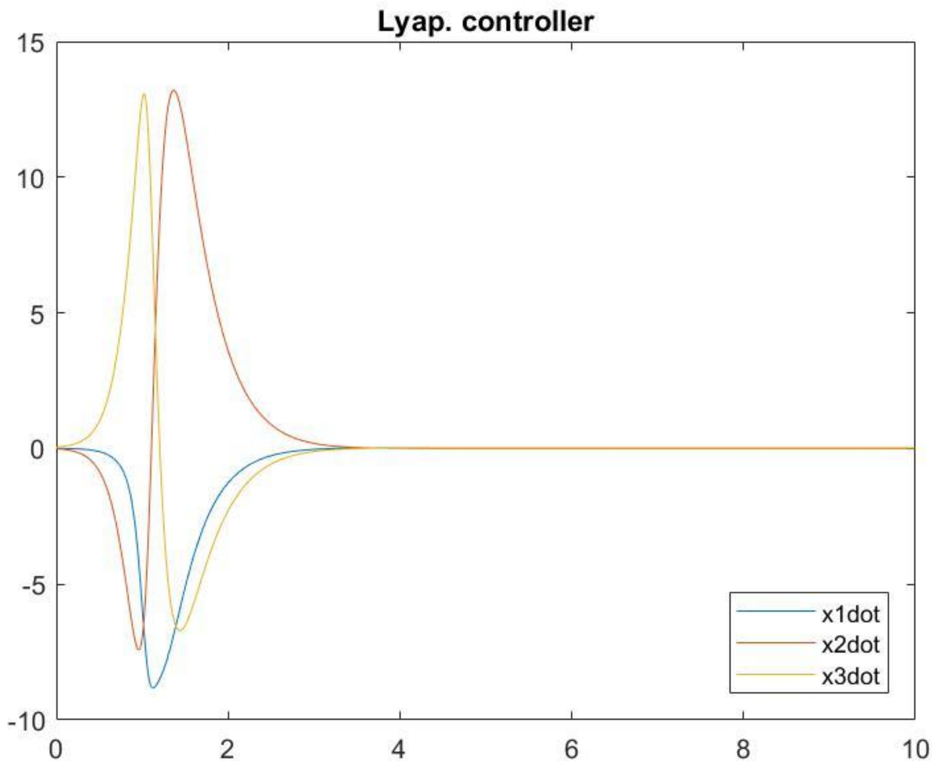
令 $v(x)=0.5*(x_1+x_2+x_3)^2$，恆≧0。

則 $v'=(x_1+x_2+x_3)(x_1'+x_2'+x_3')\equiv\alpha(x)+\beta(x)*u$

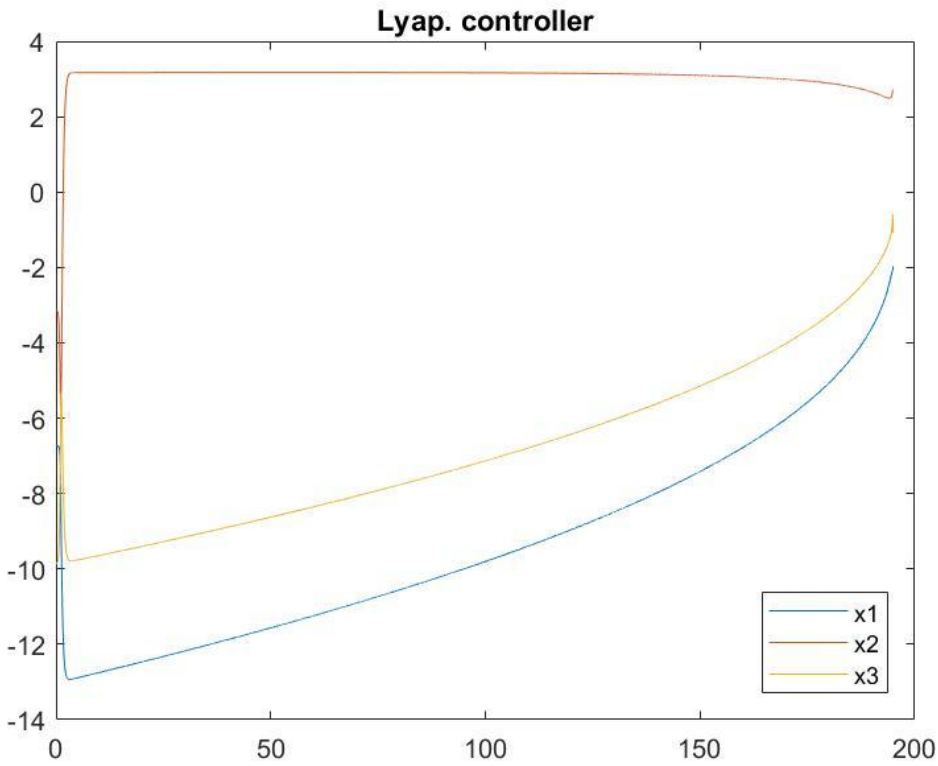
設計 $u=(\alpha(x)+1)/\beta(x)$使 v'恆等於-1。

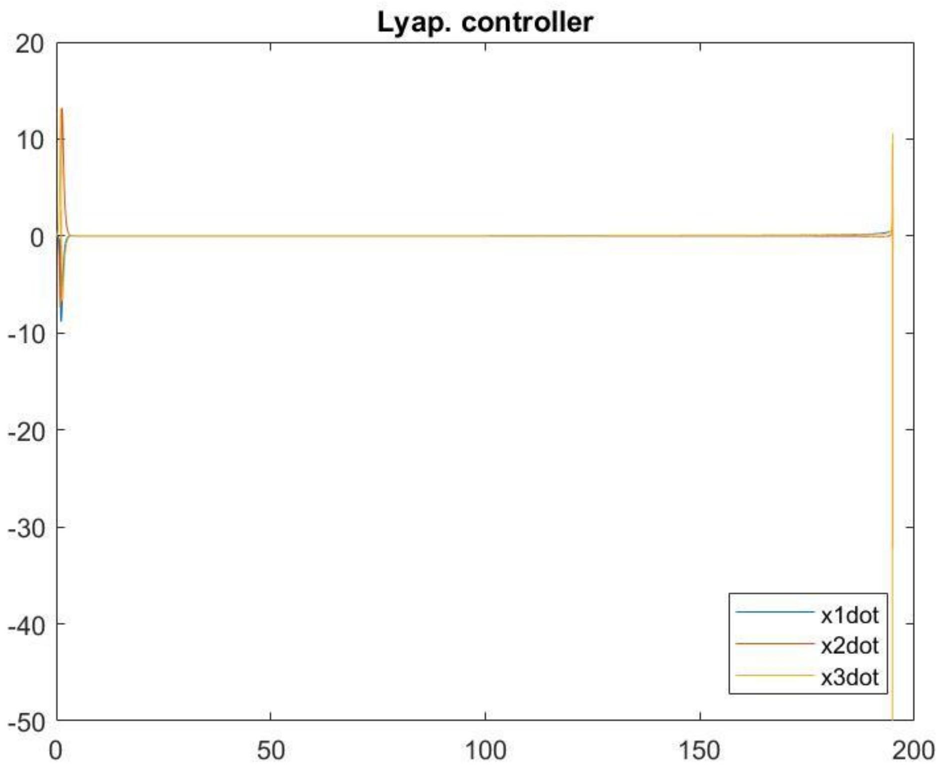
這題超困難的，怎麼做都發散。後來嘗試先算出系統的其中一個平衡點 $(x1,x2,x3)=(\pi-\pi^2,-\pi,-\pi^2)$，直接將初值設在這裡。原本預期它會一直停在這裡，但過了一陣子它又跑向另一個平衡點了。





它在下一個平衡過了一陣子後，開始衝向原點。但是因為 u 的設計在原點會發散，所以加入了一些判斷式適時將 u 鬆開，結果鬆開幾次系統就再也控不下來了(大概在 196 秒發散)


Lyap. controller
4
2
0
-2
-4
-6
-8
-10
-12
-14
0
50
100
150
200
x1
x2
x3


Lyap. controller
20
10
0
-10
-20
-30
-40
-50
0
50
100
150
200
x1dot
x2dot
x3dot

程式碼

```
%lyap.
clc;clear;
delta=0.01;
totalTime=200;
totalStep=totalTime/delta;

x1array=[1:totalStep]*0;x2array=x1array;x3array=x1array;
x1_dot=x1array;x2_dot=x1array;x2_dot=x1array;

x1array(1)=pi-(pi)^2;x2array(1)=-pi;x3array(1)=-pi^2;%init condition
for i=1:totalStep
   x1=x1array(i);x2=x2array(i);x3=x3array(i);
   v(i)=0.5*(x1+x2+x3)^2;
   u(i)=(-(x2+x1+2*sin(x1-x3)+(x1-x3)^2)-1/(x1+x2+x3));
   v_dot(i)=(x1+x2+x3)*(x2+x1+2*(sin(x1-x3))+(x1-x3)^2+u(i));
   fprintf('i=%d v=%f v_dot=%f u=%f\n',i,v(i),v_dot(i),u(i));
   if v(i)<0.001
     u(i)=0;
     fprintf('DANGER1 !!!\n');
   end
   if abs(u(i))>10^10
     u(i)=10^10*sign(u(i));
     fprintf('DANGER2 !!!\n');
   end




   x1_dot(i)=x2+x1-x3+sin(x1-x3);
   x2_dot(i)=x3+(x1-x3)^2;
   x3_dot(i)=sin(x1-x3)+u(i);

   x1array(i+1)=x1+x1_dot(i)*delta;
   x2array(i+1)=x2+x2_dot(i)*delta;
   x3array(i+1)=x3+x3_dot(i)*delta;
end

figure(1);
```

```
plot([0:1:totalStep]*delta,x1array);hold on;
plot([0:1:totalStep]*delta,x2array);hold on;
plot([0:1:totalStep]*delta,x3array);legend('x1','x2','x3','location',
'southeast');
title('Lyap. controller');
figure(2);
plot([0:1:totalStep-1]*delta,x1_dot);hold on;
plot([0:1:totalStep-1]*delta,x2_dot);hold on;
plot([0:1:totalStep-1]*delta,x3_dot);legend('x1dot','x2dot','x3dot','
location','southeast');
title('Lyap. controller');
```

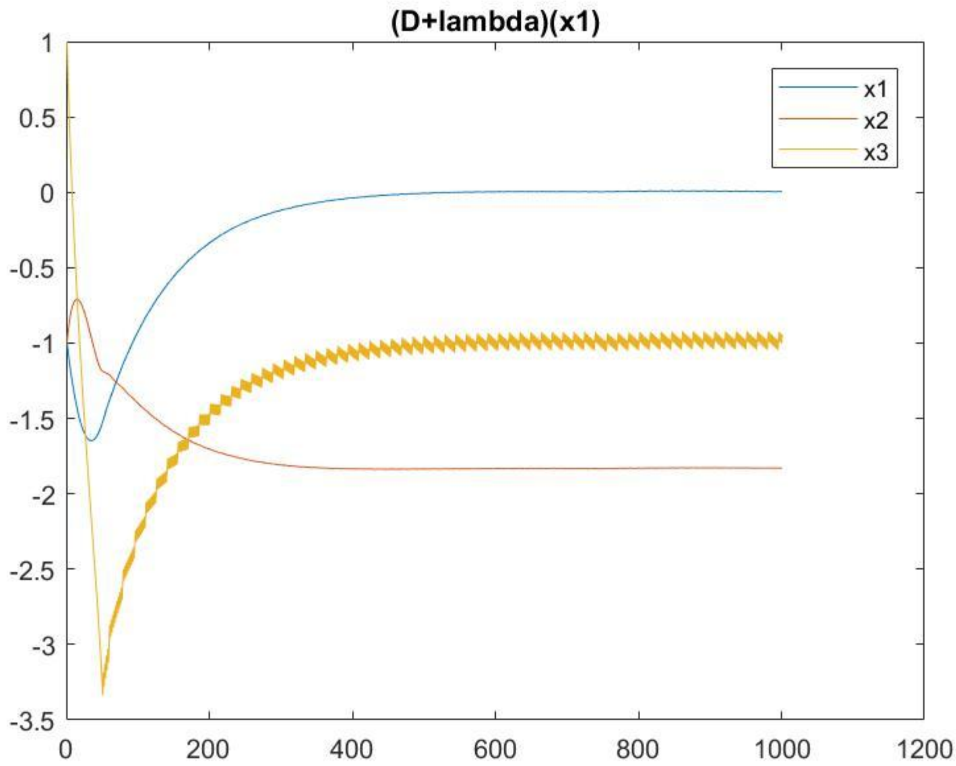
(2)sliding control

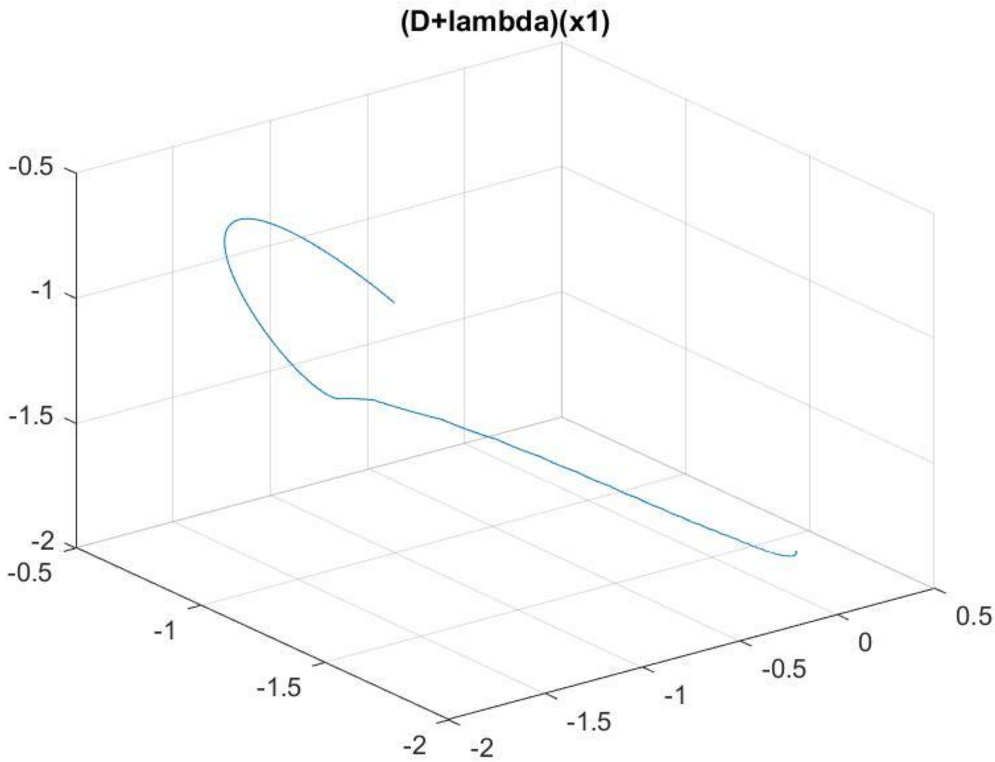
<1>

令$f=(D+\lambda)(x_1)=x_1'+\lambda x_1=x_2+x_1-x_3+\sin(x_1-x_3)+\lambda x_1$;

則$f'=x_2'+x_1'-x_3'+(\sin(x_1-x_3))'+\lambda x_1'\equiv\alpha(x)+\beta(x)*u$;

設計u使$f'=-K*\mathrm{sign}(f)$，讓f以$e^{-\lambda t}$收斂。最後收斂到$x_1=0$。



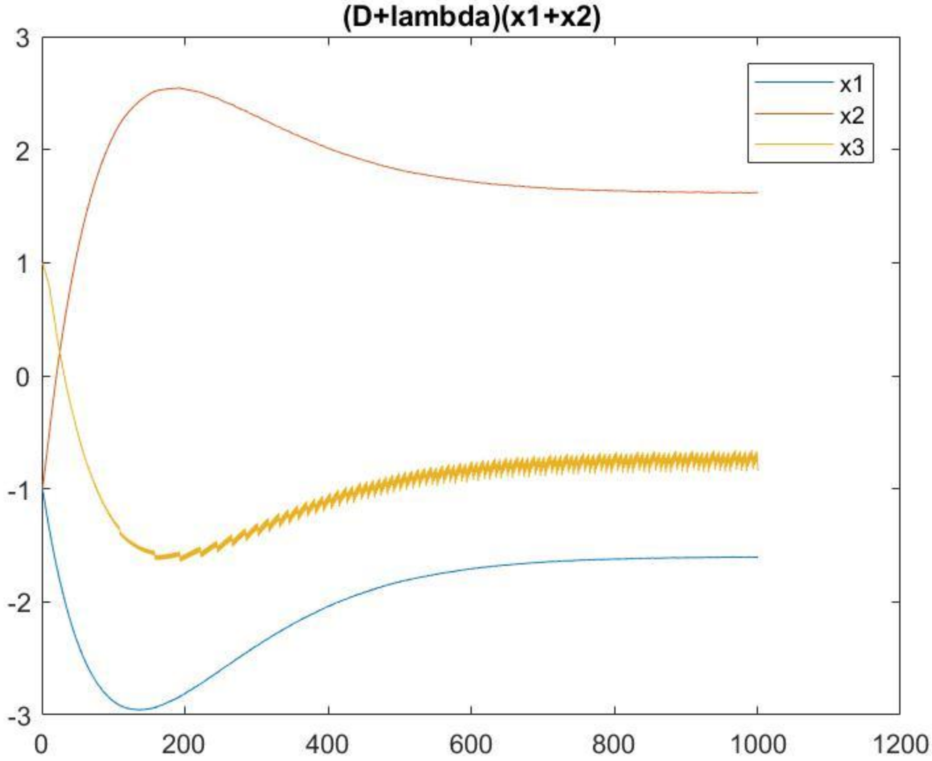


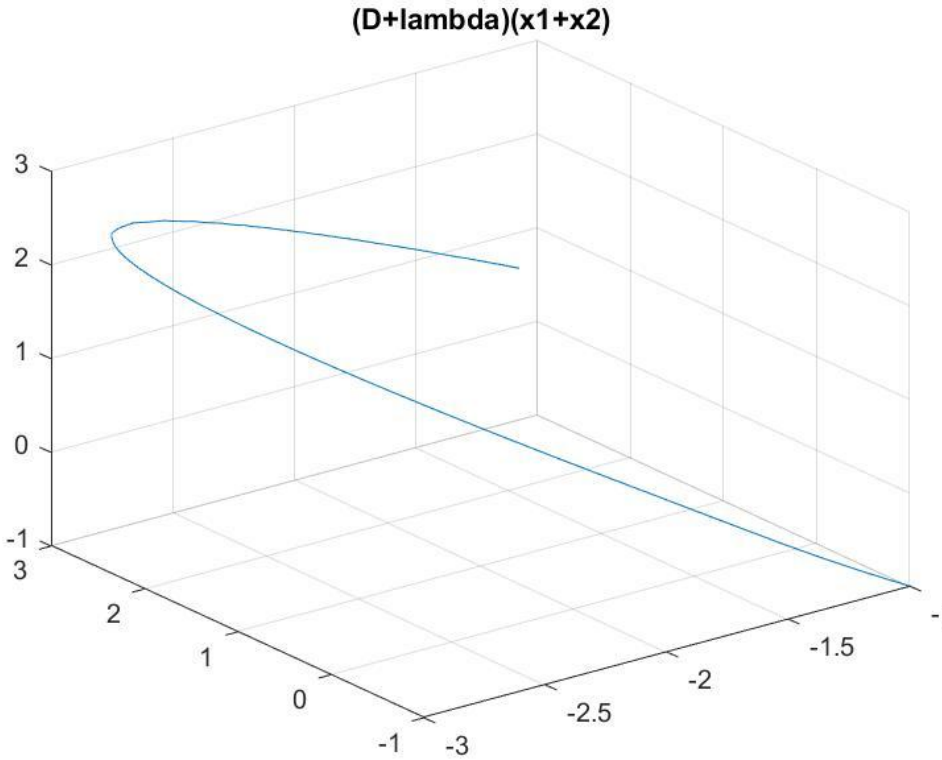
<2>

令$f=(D+\lambda)(x_1+x_2)=x_1+x_2+\sin(x_1-x_3)+(x_1-x_3)^2+\lambda(x_1+x_2)$;

則$f'= x_1'+x_2'+(\sin(x_1-x_3))'+2(x_1-x_3)(x_1'-x_3')+\lambda(x_1'+x_2')$;$\equiv\alpha(x)+\beta(x)*u$;

設計u使$f'=-K*\mathrm{sign}(f)$，讓f以$e^{-\lambda t}$收斂。最後收斂到$x_1+x_2=0$。


(D+lambda)(x1+x2)
x1
x2
x3
3
2
1
0
-1
-2
-3
0
200
400
600
800
1000
1200


(D+lambda)(x1+x2)
3
2
1
0
-1
3
2
1
0
-1
-3
-2.5
-2
-1.5
-1

程式碼

```
%f=(D+lambda)(x1+x2)
clc;clear;

lambda=1;
K=10;
delta=0.01;
totalTime=10;
totalStep=totalTime/delta;
x1array=[1:totalStep]*0;x2array=x1array;x3array=x1array;
x1_dot=x1array;x2_dot=x1array;x2_dot=x1array;
x1array(1)=-1;x2array(1)=-1;x3array(1)=1;%init condition
for i=1:totalStep
    x1=x1array(i);x2=x2array(i);x3=x3array(i);

    f(i)=x1+x2+sin(x1-x3)+(x1-x3)^2+lambda*(x1+x2);

u(i)=(K*sign(f(i))+(1+lambda)*((x2+x1-x3+sin(x1-x3))+(x3+(x1-x3)^2))+
(cos(x1-x3)+2*(x1-x3))*(x2+x1-x3))/(cos(x1-x3)+2*(x1-x3));

    x1_dot(i)=x2+x1-x3+sin(x1-x3);
    x2_dot(i)=x3+(x1-x3)^2;
    x3_dot(i)=sin(x1-x3)+u(i);

    x1array(i+1)=x1+x1_dot(i)*delta;
    x2array(i+1)=x2+x2_dot(i)*delta;
    x3array(i+1)=x3+x3_dot(i)*delta;
end

figure(1);
plot(x1array);hold on;
plot(x2array);hold on;
plot(x3array);legend('x1','x2','x3');
title('(D+lambda)(x1+x2)')

figure(2);
time=1:totalStep-1;
xt=x1array(time);
```

```
yt=x2array(time);
zt=x2array(time);
plot3(xt,yt,zt);
grid on;title('(D+lambda)(x1+x2)')
```

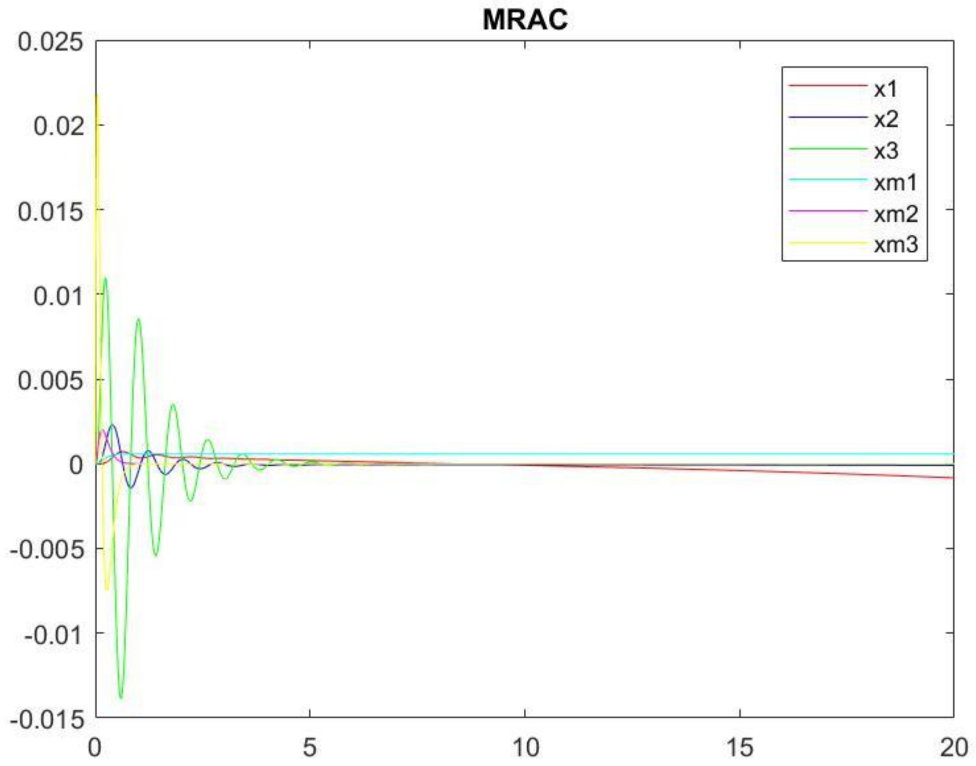
3.

設計 modle 的極點為 s=-11,s=-12,s=-13，要離虛軸夠遠系統才不會發散。

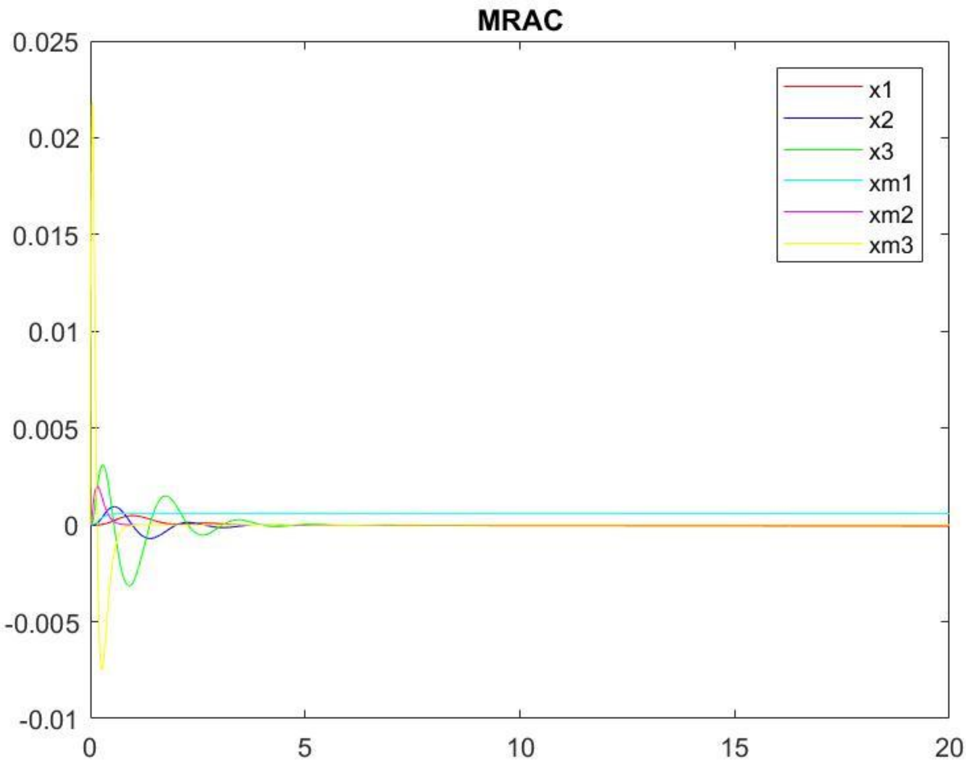
經計算得 Am=$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -14 & -4 & -3 \end{bmatrix}$

(1)r=unit step

一開始先預設 Q 為單位矩陣，γ 皆為 1。

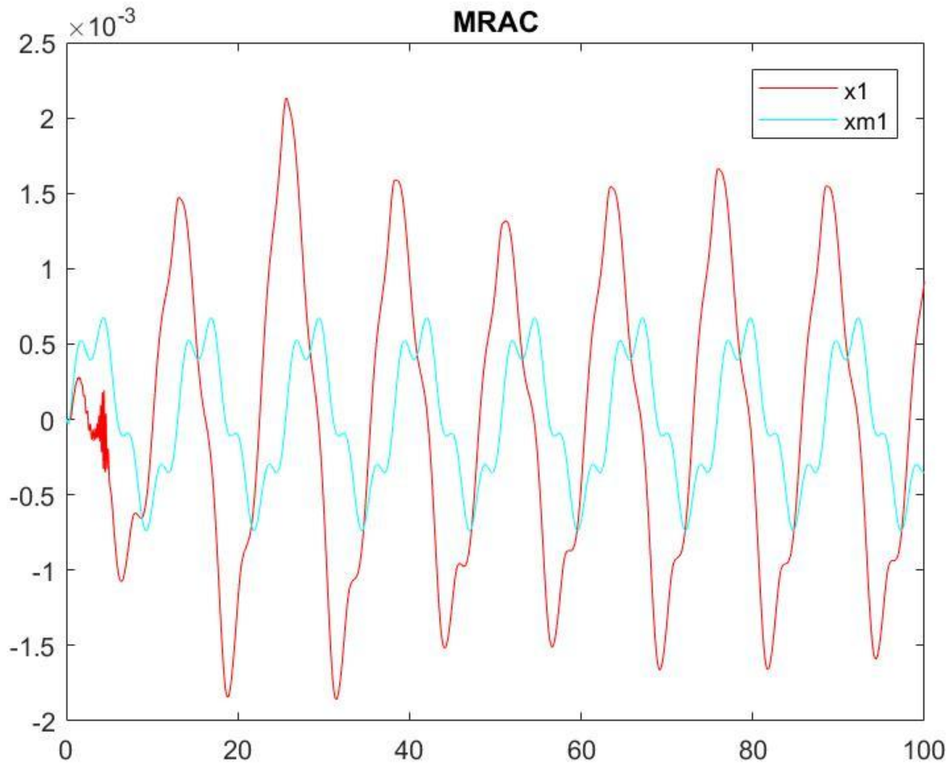


發現它快追上之後又會垂下來，其中 x_1 的誤差最嚴重而且越來越大，最後導致發散。故調整權重，改成 Q=$\begin{bmatrix} 0.0001 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1000 \end{bmatrix}$，也試過調 γ 但是效果不彰。



(2)r=弦波的合成

$$再改成\ Q=\begin{bmatrix} 0.00001 & 0 & 0 \\ 0 & 0.00001 & 0 \\ 0 & 0 & 100000 \end{bmatrix}，\gamma\ 仍為\ 1。$$



程式碼

```
%MRAC
clear;clc;
totaltime=100;
delta=0.01;
totalstep=totaltime/delta;
%select para.
Q=[0.00001 0 0;0 0.00001 0;0 0 100000];
pole=conv([1 13],conv([1 12],[1 11]));
Am=[0 1 0;0 0 1;-pole(4) -pole(3) -pole(2)];bm=1;%model
A=[0 1 0;0 0 1;-12 -4 -3];b=1;%real sys.
P=lyap(Am,Q);
```

```
gamma0=1;gamma1=1;gamma2=1;gamma3=1;

%model
xm1(1)=0;xm2(1)=0;xm3(1)=0;
for k=1:totalstep
%    r(k)=1;
    r(k)=sin(0.5*k*delta)+0.3*cos(2*k*delta+4);
    xm1_dot(k)=xm2(k);
    xm2_dot(k)=xm3(k);
    xm3_dot(k)=Am(3,1)*xm1(k)+Am(3,2)*xm2(k)+Am(3,3)*xm3(k)+bm*r(k);
    xm1(k+1)=xm1(k)+xm1_dot(k)*delta;
    xm2(k+1)=xm2(k)+xm2_dot(k)*delta;
    xm3(k+1)=xm3(k)+xm3_dot(k)*delta;
end

%real sys.
theta0(1)=0;theta1(1)=0;theta2(1)=0;theta3(1)=0;
x1(1)=0;x2(1)=0;x3(1)=0;
for k=1:totalstep


u(k)=theta0(k)*r(k)+theta1(k)*x1(k)+theta2(k)*x2(k)+theta3(k)*x3(k);

    x1_dot(k)=x2(k);
    x2_dot(k)=x3(k);
    x3_dot(k)=A(3,1)*x1(k)+A(3,2)*x2(k)+A(3,3)*x3(k)+1*u(k);
    x1(k+1)=x1(k)+x1_dot(k)*delta;
    x2(k+1)=x2(k)+x2_dot(k)*delta;
    x3(k+1)=x3(k)+x3_dot(k)*delta;



    e1(k)=xm1(k)-x1(k);
    e2(k)=xm2(k)-x2(k);
    e3(k)=xm3(k)-x3(k);
    zeta(k)=0.5*(P(1,3)*e1(k)+P(2,3)*e2(k)+P(3,3)*e3(k));
    theta0_dot(k)=zeta(k)*r(k)/(b*gamma0);
    theta1_dot(k)=zeta(k)*x1(k)/(b*gamma1);
    theta2_dot(k)=zeta(k)*x2(k)/(b*gamma2);
```

```
    theta3_dot(k)=zeta(k)*x3(k)/(b*gamma3);

    theta0(k+1)=theta0(k)+theta0_dot(k)*delta;
    theta1(k+1)=theta1(k)+theta1_dot(k)*delta;
    theta2(k+1)=theta2(k)+theta2_dot(k)*delta;
    theta3(k+1)=theta3(k)+theta3_dot(k)*delta;

end


plot([0:1:totalstep]*delta,x1,'r');hold on;
% plot([0:1:totalstep]*delta,x2,'b');hold on;
% plot([0:1:totalstep]*delta,x3,'g');hold on;
plot([0:1:totalstep]*delta,xm1,'c');hold on;
% plot([0:1:totalstep]*delta,xm2,'m');hold on;
% plot([0:1:totalstep]*delta,xm3,'y');hold on;
legend('x1','xm1');
title('MRAC');
```

4.

$C_1(s)=\frac{s+1}{s+100}$、$C_2(s)=\frac{s+50}{s+100}$。

看起來 C_2 抗雜訊功能較佳，只是前期震盪也較嚴重。