**HW4**

n=1+0+5+3+0+3+0+6+1;

a = fix(n/10);

b = rem(n,10);

**所以a = 1 b = 9**

**兩根時**

pole = [-a+b\*j –a+b\*j ]

可得 **(s + 1 - 9i)\*(s + 1 + 9i) = s^2 + 2s + 82**

**三根時**

pole = [-a+b\*j –a+b\*j -a]

可得 **(s + 1 - 9i)\*(s + 1 + 9i)\*(s + 1) = s^3 + 3s^2 + 84s + 82**

**甲:**

x1\_dot = x1^2 + x2

x2\_dot = -x1 + u

(a)

y = x1

z1 = y = x1, z2 = z1\_dot = y\_dot

所以: (y\_dot2表示微分兩次)

y\_dot = x1\_dot = x1^2 + x2

y\_dot2 = x1\_dot2 = 2x1\*x1\_dot + x2\_dot

= 2x1^3 + 2x1x2 – x1\*u

在y\_dot2可以找到對u的關係(只需要兩個eigenvalue)，因此可得:

z1\_dot = y\_dot = z2

z2\_dot = y\_dot2 = 2x1^3 + 2x1x2 – x1 + u

為使z2\_dot = α(x) + β(x)u

所以:

α(x) = 2x1^3 + 2x1x2

β(x) = –x1

(b)

由於

z1\_dot = y\_dot = z2

z2\_dot = y\_dot2 = 2x1^3 + 2x1x2 – x1 + u

為使z的狀態方程式可以線性化達到線性回饋目的

所以需要把z2\_dot的非線性部分去除掉:

希望為: z2\_dot = -k1\*z1 - k2\*z2

所以設定u = -(2x1^3 + 2x1x2 - x1 + k1\*z1 + k2\*z2)

此時的z狀態方程式為:

z1\_dot = z2

z2\_dot = -k1\*z1 - k2\*z2

所以z狀態的A矩陣為[0 1;-k1 -k2]

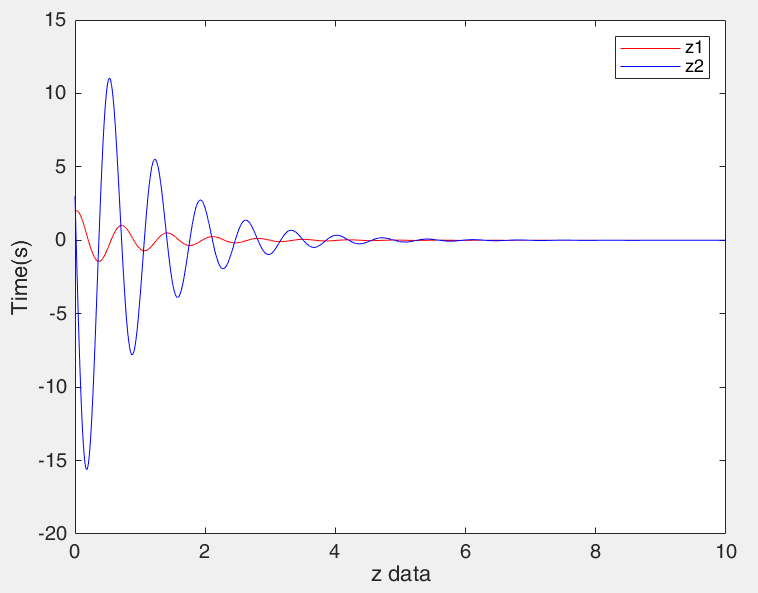
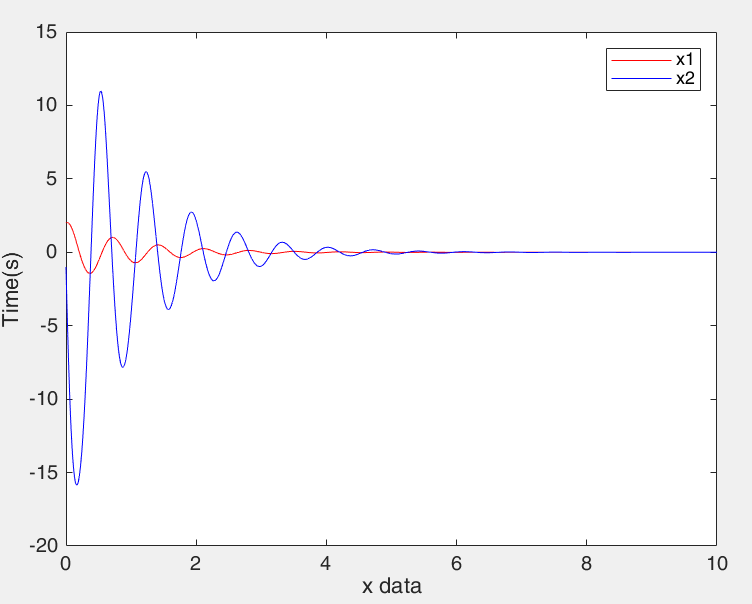
配合題目的eigenvalue設計:

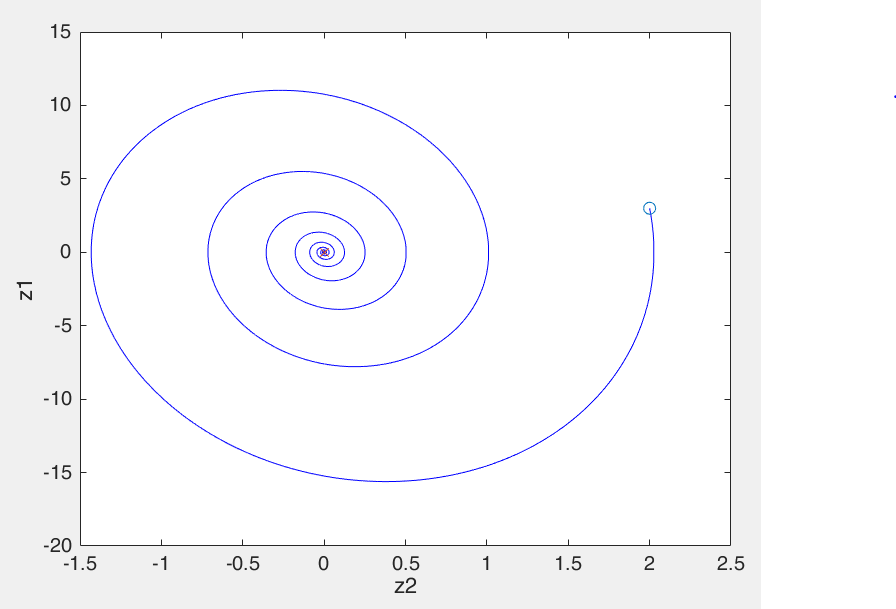
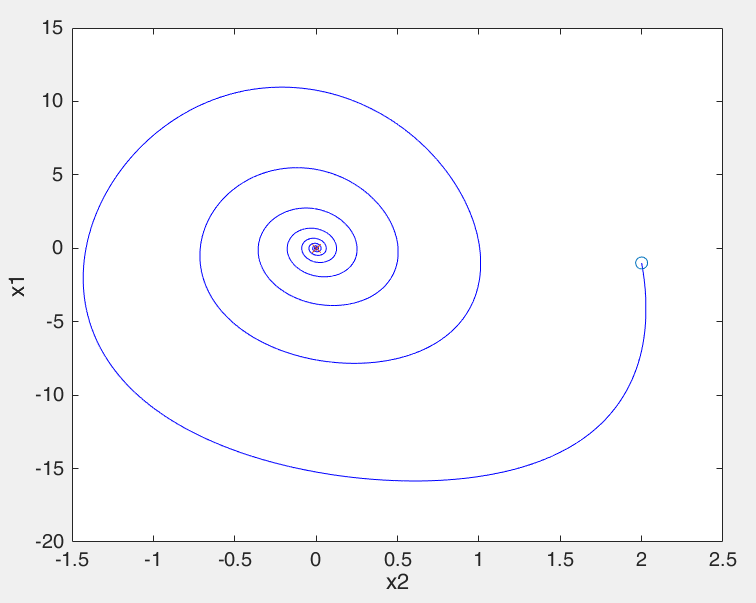
det(A) = s^2 + k2\*s + k1 = s^2 + 2s + 82

所以

k1 = 82; k2 = 2

(c)作圖如下





程式碼:主函式

%% main\_code

delt = 0.0001;

totTime = 10 ;

totalStep = totTime/delt ;

tarr= [0 : 1 : totalStep]\*delt;

x = [2 -1]'

xarr1 = [0,0,0,0]';xarr2 = xarr1;

xarr1(1) = x(1); xarr2(1) = x(2);

y = x(1)

zarr1 = [0,0,0,0]';zarr2 = zarr1;

zarr1(1) = x(1); zarr2(1) = x(1)^2+x(2);

x\_1 = xarr1(1); x\_2 = xarr2(1);

z\_1 = zarr1(1); z\_2 = zarr2(1);

Bx = [0;1]

for i=1:totalStep

Ax = [ x\_1 1;

-1 0

]

ux = -1\*(2\*x\_1^3 + 2\*x\_1\*x\_2 - x\_1 + 82\*z\_1 + 2\*z\_2);

xN = myFindNextPos2\_u(Ax,x,delt,Bx,ux);

xarr1(i+1) = xN(1);xarr2(i+1) = xN(2);

x = xN;

x\_1 = xN(1);x\_2 = xN(2);

z\_1 = x\_1; z\_2 = x\_1^2+x\_2;

zarr1(i+1) = z\_1;zarr2(i+1) = z\_2;

end

figure();plot(tarr,xarr1,'-r',tarr,xarr2,'-b');legend('x1','x2');ylabel('Time(s)');xlabel('x data');

figure();plot(tarr,zarr1,'-r',tarr,zarr2,'-b');legend('z1','z2');ylabel('Time(s)');xlabel('z data');

figure()

plot(xarr1,xarr2,'-b');hold on;

plot(xarr1(1),xarr2(1),'o',xarr1(i+1),xarr2(i+1),'x');

ylabel('x1');xlabel('x2');

figure()

plot(zarr1,zarr2,'-b');hold on;

plot(zarr1(1),zarr2(1),'o',zarr1(i+1),zarr2(i+1),'x');

ylabel('z1');xlabel('z2');

程式碼:副函式

function xNew = myFindNextPos2\_u(A,x,delt,B,u)

x\_dot = zeros(size(x));

xNew = x\_dot;

ut = B\*u;

x\_dot = A\*x+ut

xNew = x + x\_dot\*delt;

return;

end

function xNew = myFindNextPos2(A,x,delt)

x\_dot = zeros(size(x));

xNew = x\_dot;

x\_dot = A\*x+ut;

xNew = x + x\_dot\*delt;

return;

end

**乙:**

x1\_dot = x1^3 + x2

x2\_dot = x3

x3\_dot = u

讓output y=x1 & (z1=y,z2=y\_dot,z3=y\_dot2)，設計FB control 使系統收斂:

運用FB linearization如下:

z1 = y = x1

z1\_dot = z2 = x1\_dot = x1^3 + x2

z2\_dot = z3 = x1\_dot2 = 3x1^5 + 3x1^2x2 + x3

z3\_dot = x1\_dot3 = 15x1^7 + 21x1^4x2 + 6x1x2^2 + 3x1^2x3 + u

於z3\_dot產生對u的關係式(產生三個根)，為使回饋線性化變為:

z1\_dot = z2

z2\_dot = z3

z3\_dot = -k1z1 –k2z2 –k3z3

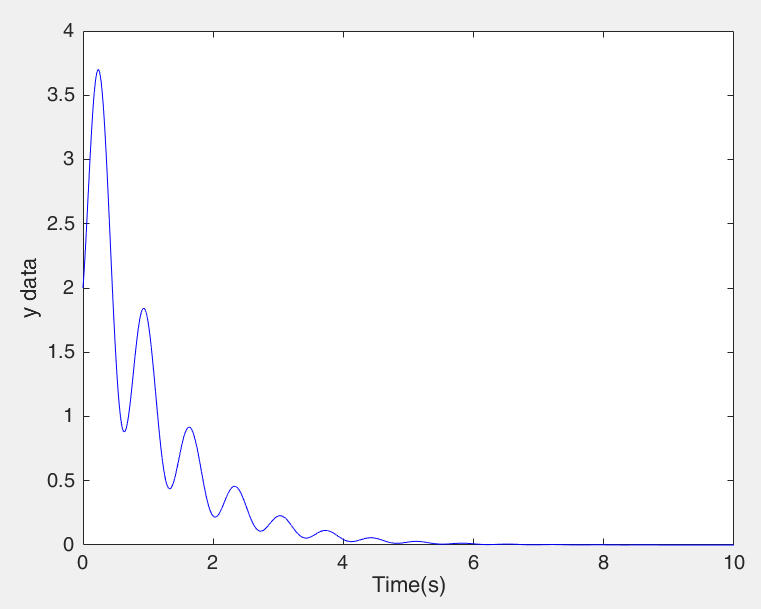
其中z之狀態A矩陣為 = [0 1 0;0 0 1;-k1 –k2 –k3]

配合題目的eigenvalue設計:

det(A) = s^3 + k3\*s^2 + k2\*s + k1 = s^3 + 3s^2 + 84s + 82

所以 k1 = 82; k2 = 84; k3 = 3

作圖如下



程式碼:主函式

delt = 0.0001;

totTime = 10 ;

totalStep = totTime/delt ;

tarr= [0 : 1 : totalStep]\*delt;

x = [2 -1 -3]'

xarr1 = [0,0,0,0]';xarr2 = xarr1;xarr3 = xarr1;

xarr1(1) = x(1); xarr2(1) = x(2); xarr3(1) = x(3);

y(1) = x(1);

zarr1 = [0,0,0,0]';zarr2 = zarr1;zarr3 = zarr1;

zarr1(1) = x(1); zarr2(1) = x(1)^3+x(2); zarr3(1) = 3\*x(1)^5+3\*x(1)^2\*x(2)+x(3);

x\_1 = xarr1(1); x\_2 = xarr2(1); x\_3 = xarr3(1);

z\_1 = zarr1(1); z\_2 = zarr2(1); z\_3 = zarr3(1);

z = [z\_1 z\_2 z\_3]';

Bx = [0;0;1];

for i=1:totalStep

Ax = [ x\_1^2 1 0;

0 0 1;

0 0 0

];

ux = -1\*(15\*x\_1^7 + 21\*x\_1^4\*x\_2 + 6\*x\_1\*x\_2^2 + 3\*x\_1^2\*x\_3 + 82\*z\_1 + 84\*z\_2 + 3\*z\_3);

xN = myFindNextPos2\_u(Ax,x,delt,Bx,ux);

xarr1(i+1) = xN(1);xarr2(i+1) = xN(2);xarr3(i+1) = xN(3);

x = xN;

x\_1 = xN(1);x\_2 = xN(2);x\_3 = xN(3);

z\_1 = x\_1; z\_2 = x\_1^3+x\_2; z\_3 = 3\*x\_1^5+3\*x\_1^2\*x\_2+x\_3;

zarr1(i+1) = z\_1;zarr2(i+1) = z\_2;zarr3(i+1) = z\_3;

y(i+1) = z\_1;

end

figure()

plot(tarr,y,'-b');xlabel('Time(s)');ylabel('y data');

**丙:**

x1\_dot = 0.5x1^3 + θ1x2

x2\_dot = θ2x1x2 + u

z1 = x1

z1\_dot = x1\_dot = 0.5x1^3 + θ1x2

z2\_dot = x1\_dot2 = 0.75\*x1^5 + 1.5θ1x1^2x2 + θ1θ2x1x2 + θ1u

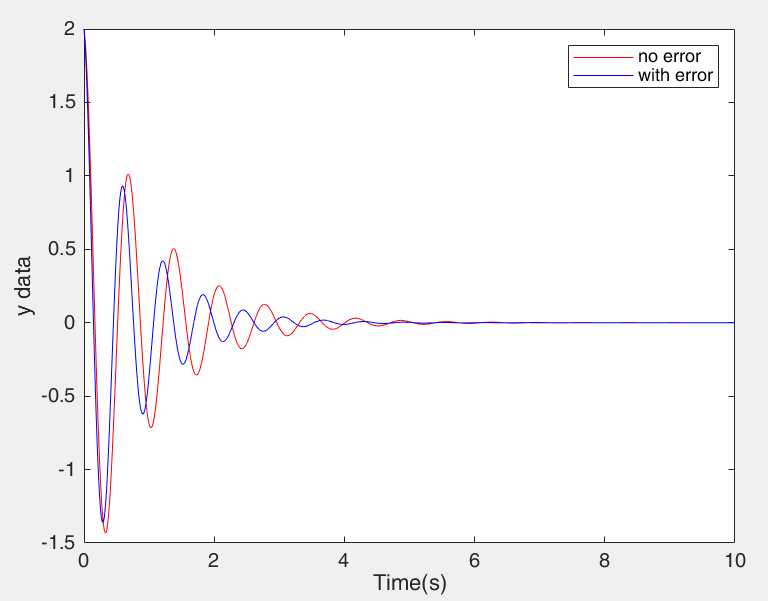
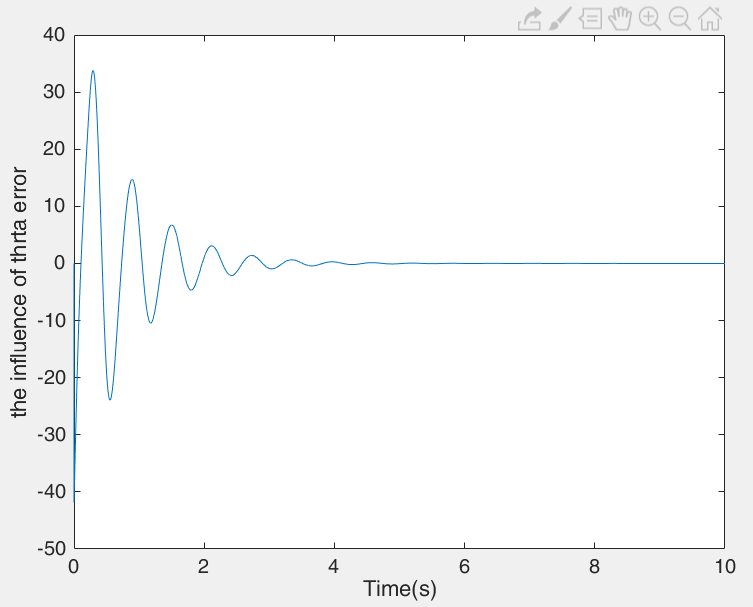
由於系統θ1、θ2會有變動，可能使系統更穩定或更不穩定，所以在考量有誤差時z2\_dot的量值時，可以把z2\_dot - θ1u進行觀察，z2\_dot - θ1u<0會更穩定，反之更不穩定。

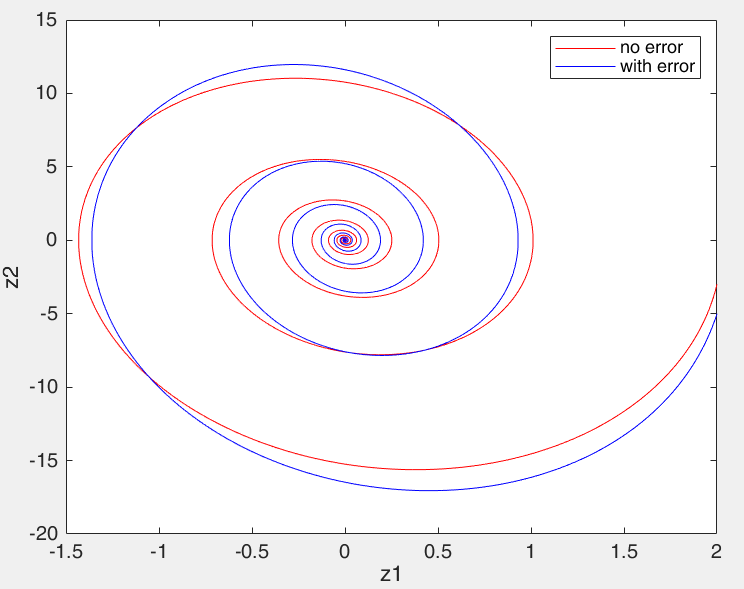
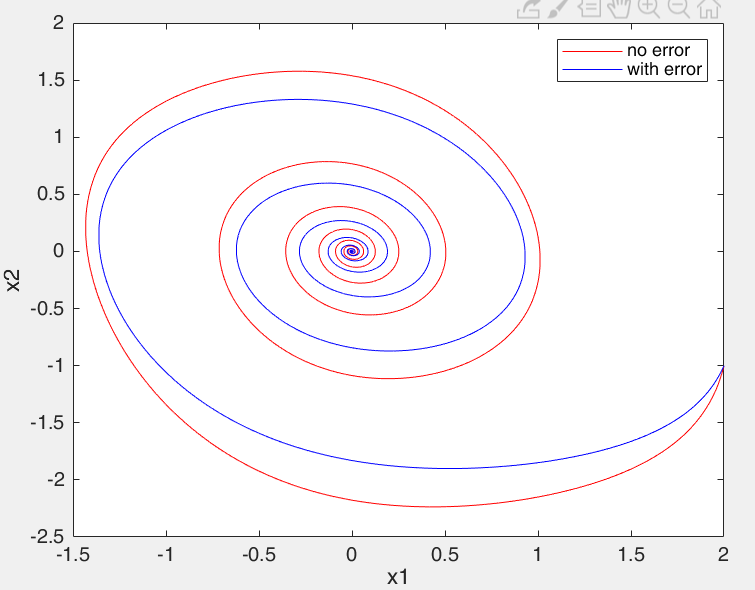
這主要是因為

z2\_dot - θ1u = 1.5\*Δ1x1^2x2 + (Δ1θ2-Δ2θ1-Δ1Δ2)x1x2 + Δ1u

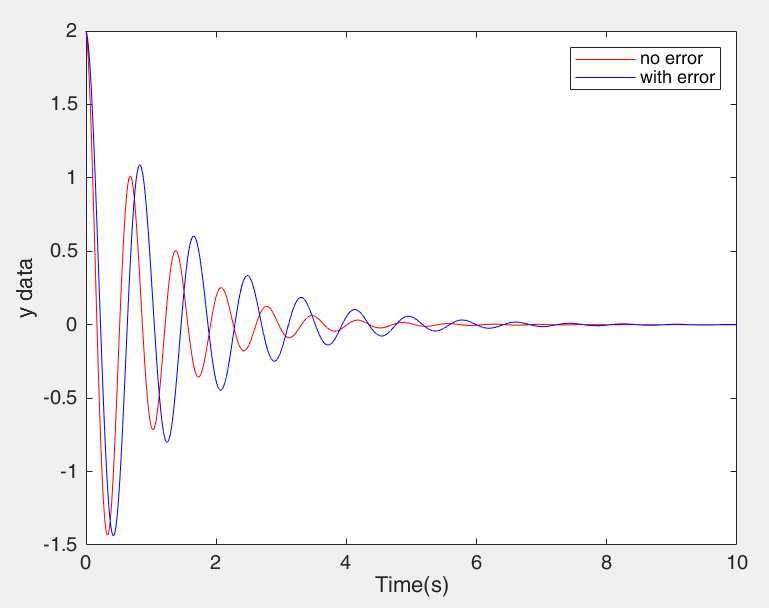
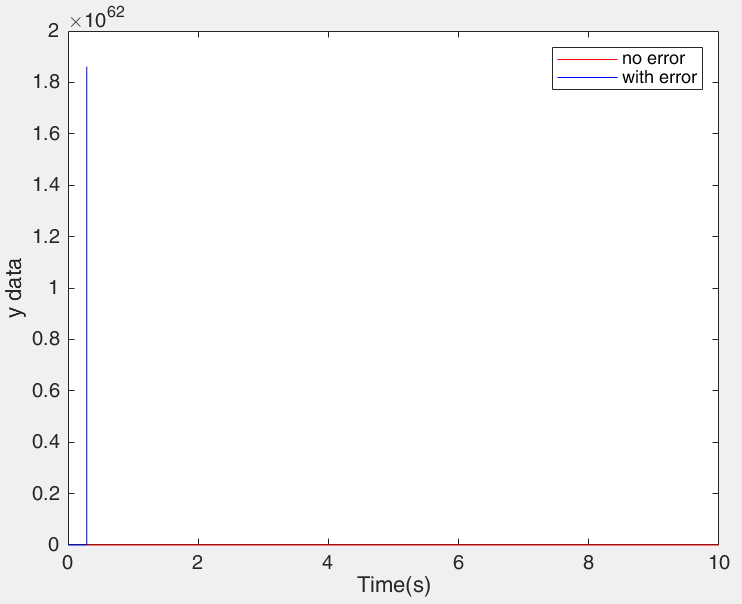
其中主要影響來自於Δ1u，所以Δ1的正負值影響很重要。若是沒有考慮Δ的情形，u受線性化影響所以會小於零，如果Δ1是正的，則z2\_dot - θ1u會變得更小，系統更穩定；如果Δ1是負的，則z2\_dot - θ1u會變得更大，系統更不穩定。

z2\_dot - θ1u作為誤差對系統影響並單獨跑出圖進行觀測，由下圖可知系統初期皆小於0，因此更穩定:





最後附上一張當Δ1=-2Δ2=-0.5的輸出結果

(x1=2 ,x2=-1)|t=0 (x1=3.2 ,x2=-1)|t=0

程式碼:主程式

dth1 = 2;dth2 = -0.5;

totTime = 10 ;

x = [2 -1]';

delt = 0.0001;

totalStep = totTime/delt ;

tarr= [0 : 1 : totalStep]\*delt;

theta1 = 7;theta2 = 2;

theta\_1 = theta1;theta\_2 = theta2;

% no error

xarr1 = [0,0,0,0]';xarr2 = xarr1;

xarr1(1) = x(1); xarr2(1) = x(2);

y(1) = x(1);

zarr1 = [0,0,0,0]';zarr2 = zarr1;

zarr1(1) = x(1); zarr2(1) = 0.5\*x(1)^3+(theta\_1)\*x(2);

x\_1 = xarr1(1); x\_2 = xarr2(1);

z\_1 = zarr1(1); z\_2 = zarr2(1);

z = [z\_1 z\_2]';

Bx = [0;1];

for i=1:totalStep

Ax = [ 0.5\*x\_1^2 (theta\_1);

0 (theta\_2)\*x\_1

];

ux = -(0.75\*x\_1^5 + 1.5\*theta1\*x\_1^2\*x\_2+theta1\*theta2\*x\_1\*x\_2+82\*z\_1 + 2\*z\_2)/(theta1);

xN = myFindNextPos2\_u(Ax,x,delt,Bx,ux);

xarr1(i+1) = xN(1);xarr2(i+1) = xN(2);

x = xN;

x\_1 = xN(1);x\_2 = xN(2);

z\_1 = x\_1; z\_2 = 0.5\*x\_1^3+(theta\_1)\*x\_2;

zarr1(i+1) = z\_1;zarr2(i+1) = z\_2;

y(i+1) = z\_1;

end

theta\_1 = theta1+dth1;theta\_2 = theta2+dth2;

x\_e = [2 -1]';

% with error

xarr1\_e = [0,0,0,0]';xarr2\_e = xarr1\_e;

xarr1\_e(1) = x\_e(1); xarr2\_e(1) = x\_e(2);

y\_e(1) = x\_e(1);

zarr1\_e = [0,0,0,0]';zarr2\_e = zarr1\_e;

zarr1\_e(1) = x\_e(1); zarr2\_e(1) = 0.5\*x\_e(1)^3+(theta\_1)\*x\_e(2);

x\_1\_e = xarr1\_e(1); x\_2\_e = xarr2\_e(1);

z\_1\_e = zarr1\_e(1); z\_2\_e = zarr2\_e(1);

z = [z\_1\_e z\_2\_e]';

Bx = [0;1];

for i=1:totalStep

Ax = [ 0.5\*x\_1\_e^2 (theta\_1);

0 (theta\_2)\*x\_1\_e

];

ux = -(0.75\*x\_1\_e^5+1.5\*theta1\*x\_1\_e^2\*x\_2\_e+theta1\*theta2\*x\_1\_e\*x\_2\_e+82\*z\_1\_e + 2\*z\_2\_e)/(theta1);

% 檢驗系統z2\_dot受到u控制後還剩下的error量值

uu2(i+1) = 1.5\*dth1\*x\_1\_e^2\*x\_2\_e + (dth1\*theta2 - dth2\*theta1 -dth1\*dth2)\*x\_1\_e\*x\_2\_e + dth1\*ux;

xN = myFindNextPos2\_u(Ax,x\_e,delt,Bx,ux);

xarr1\_e(i+1) = xN(1);xarr2\_e(i+1) = xN(2);

x\_e = xN;

x\_1\_e = xN(1);x\_2\_e = xN(2);

z\_1\_e = x\_1\_e; z\_2\_e = 0.5\*x\_1\_e^3 + (theta\_1)\*x\_2\_e;

zarr1\_e(i+1) = z\_1\_e;zarr2\_e(i+1) = z\_2\_e;

y\_e(i+1) = z\_1\_e;

end

figure()

plot(tarr,uu2);xlabel('Time(s)');ylabel('the influence of thrta error');

figure()

plot(tarr,y,'-r',tarr,y\_e,'-b');xlabel('Time(s)');ylabel('y data');

figure()

plot(xarr1,xarr2,'-r',xarr1\_e,xarr2\_e,'-b');xlabel('x1');ylabel('x2');

figure()

plot(zarr1,zarr2,'-r',zarr1\_e,zarr2\_e,'-b');xlabel('z1');ylabel('z2');