**HW5**

**a)**

x1\_dot = x2

x2\_dot = a\*x1^3\*y + b\*x1\*x2^3 + (x1^2+x2^2)\*u + d(t)

d(t) = 0

**使用線性回饋的sliding mode control 修正**

假設 s = A\*x2 + B\*x1 (A,B可以決定狀態彎曲程度)

,

本處另 B = 2 ( 固定A = 1 )

當 s\_dot >0

u = -1\*((A\*(a\*x1^3\*x2 + b\*x1\*x2^3) + B\*x2) + k)/(x1^2+x2^2);

當 s\_dot <0

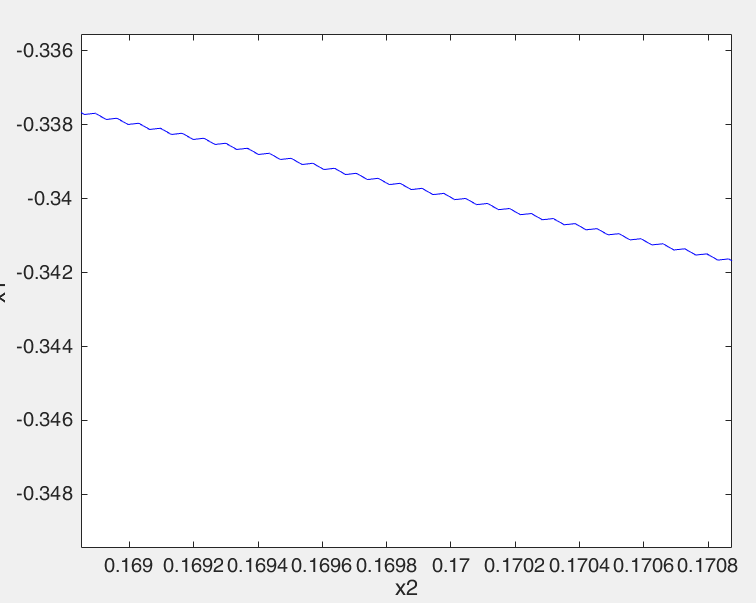
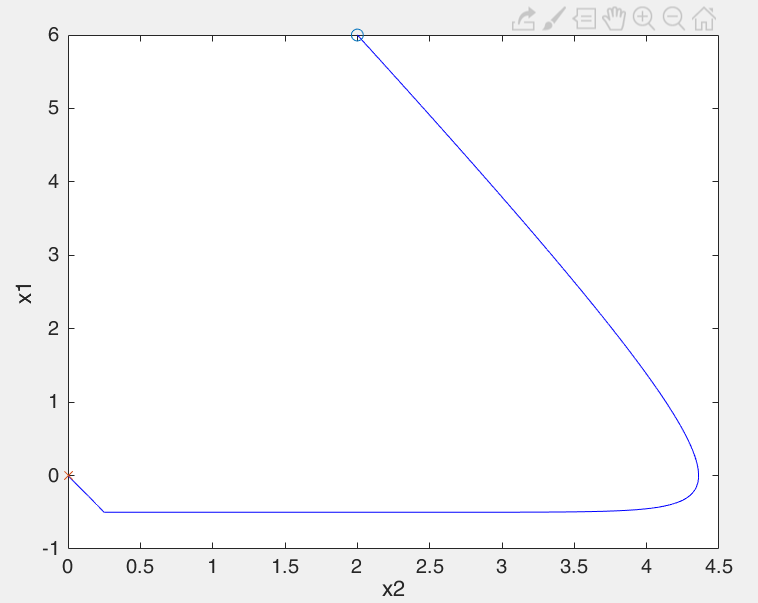
u = -1\*((A\*(a\*x1^3\*x2 + b\*x1\*x2^3) + B\*x2) - k)/(x1^2+x2^2);

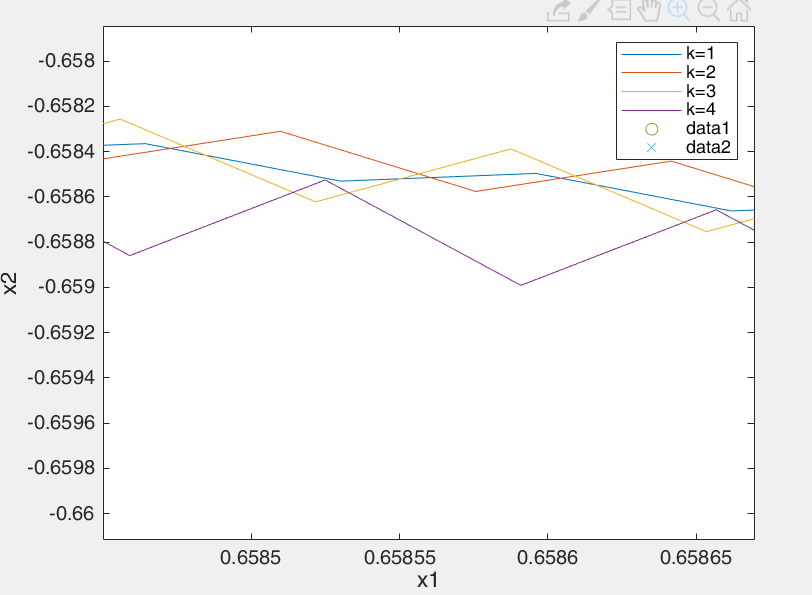
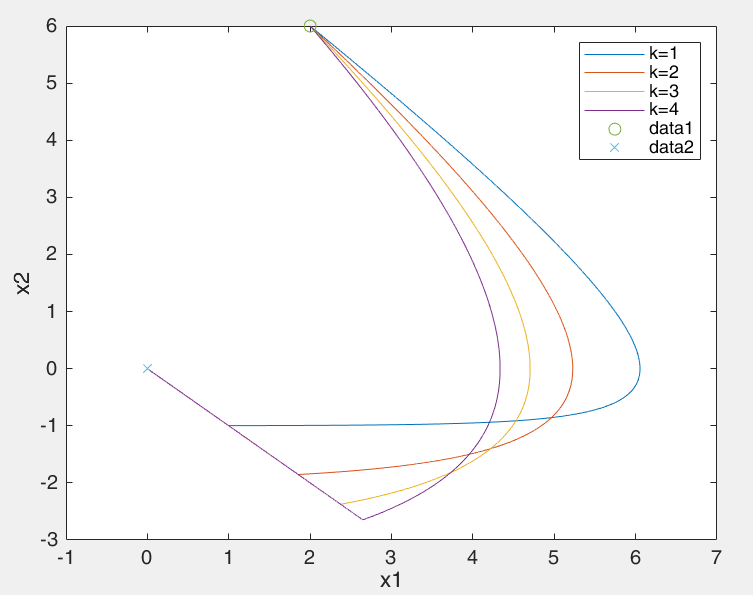
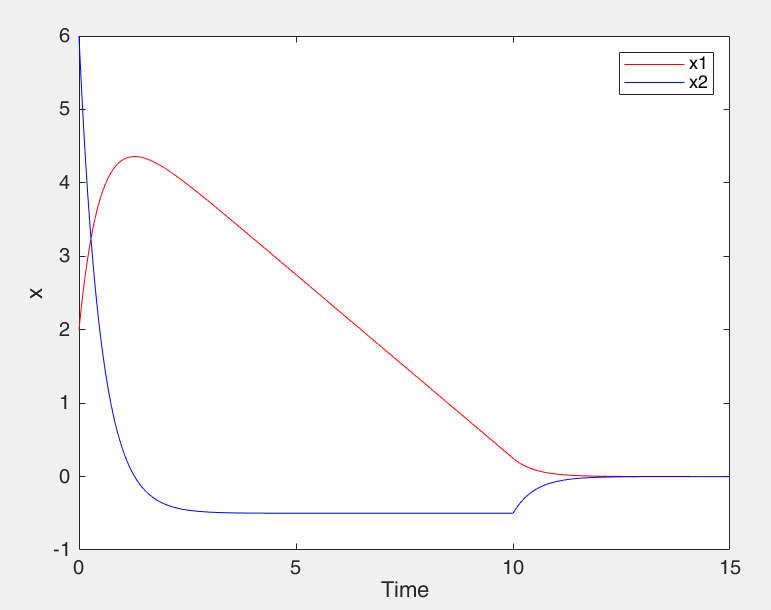
可以得到以下幾張圖。在左上圖可以看出在x1大約為0.25處有一個明顯轉折，此時已經控制到了slide mode的軌道上，x2產生非線性轉折，開始收斂。此外，跟上slide mode的軌道，斜率大約是 -0.5 ，這跟自行調整的AB有關，這是因為經由此控制器u控制上軌道後，又A為1

x1\_dot = x2

x2\_dot = (1-A)\* (a\*x1^3\*x2 + b\*x1\*x2^3) - B\*x2 = - B\*x2 (k值由於來回震盪所以不需考慮)

x2和x1的移動速度比約為 -B:1 因此可以調整上升斜率。此外在軌道上後會因為過衝產生反向控制產生顫振現象，如右上圖。而從右下圖可以觀測到k值影響系統跟上軌道速度，k較大的話不會在軌道外面轉太久，而顫振更明顯





若是不希望產生顫振問題可以用Lyapunov Func.設計方式解決問題

令V = 0.5s^2

V\_dot = s\_dot\*s

設定 s\_dot = -ks 可得 V\_dot = -k\*s^2

當 s\_dot >0

k = s

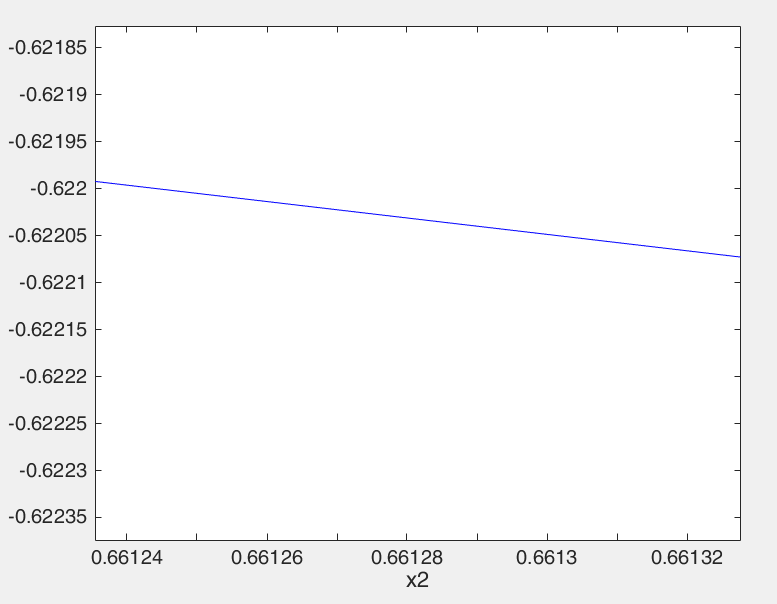
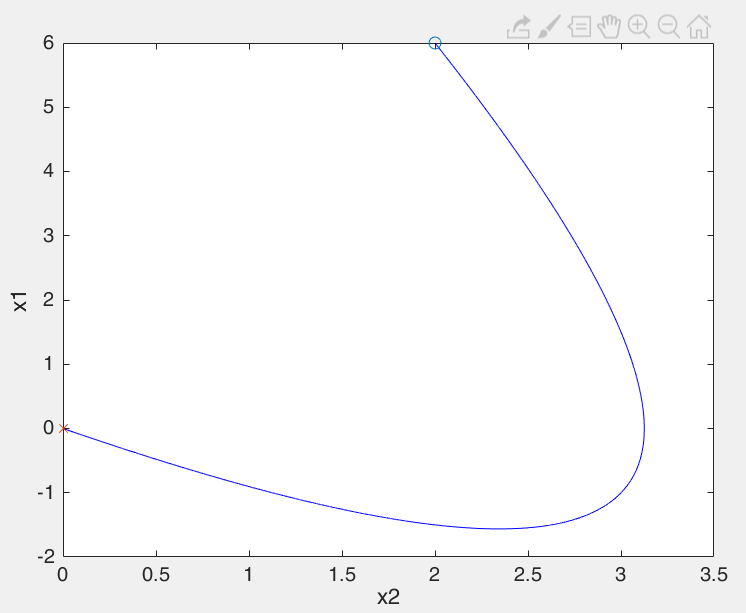
u = -1\*((A\*(a\*x1^3\*x2 + b\*x1\*x2^3) + B\*x2) + k)/(x1^2+x2^2);

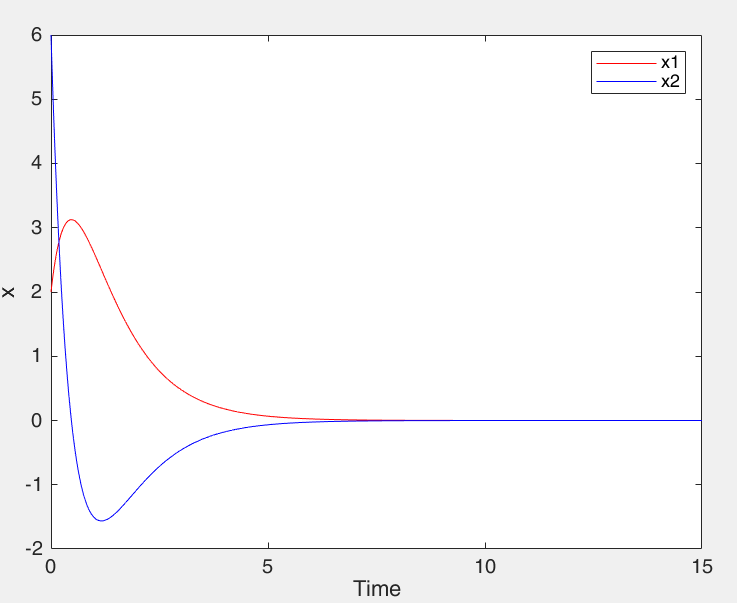
當 s\_dot <0

k = -1\*s

u = -1\*((A\*(a\*x1^3\*x2 + b\*x1\*x2^3) + B\*x2) - k)/(x1^2+x2^2);

此時會得到的軌跡會有s\_dot = -ks產生的衰退效果，所以只會漸進slide mode的軌道，減緩過衝後回復的顫振問題





**主程式:**

%% Student ID

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

a = fix(n/10);

b = rem(n,10);

%k = 1

delt = 0.0001;

totTime = 15 ;

totalStep = totTime/delt ;

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

for k=1:4

x = [2 6]'

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

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

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

for i=1:totalStep

Bx = [0;(x\_1^2+x\_2^2)];

Ax = [ 0 1;

b\*x\_2^3 a\*x\_1^3

];

A = 1;

B = 1;

s = A\*x\_2+B\*x\_1;

if s>0

%k = s; %解開註解做k=s控制，註解為k=k\*sign(s)

ux = -1\*((A\*(a\*x\_1^3\*x\_2 + b\*x\_1\*x\_2^3) + B\*x\_2) + k)/(x\_1^2+x\_2^2);

else

%k = -s; %解開註解做k=s控制，註解為k=k\*sign(s)

ux = -1\*((A\*(a\*x\_1^3\*x\_2 + b\*x\_1\*x\_2^3) + B\*x\_2) - k)/(x\_1^2+x\_2^2);

end

xNew = zeros(size(x));

ut = Bx\*ux;

x\_dot = Ax\*x+ut;

xNew = x + x\_dot\*delt;

xN = xNew;

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

x = xN;

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

end

figure(2)

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

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

end

legend('k=1','k=2','k=3','k=4')

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

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

% figure(1);plot(tarr,xarr1,'-r',tarr,xarr2,'-b');legend('x1','x2');

% ylabel('x');xlabel('Time');

% figure(2)

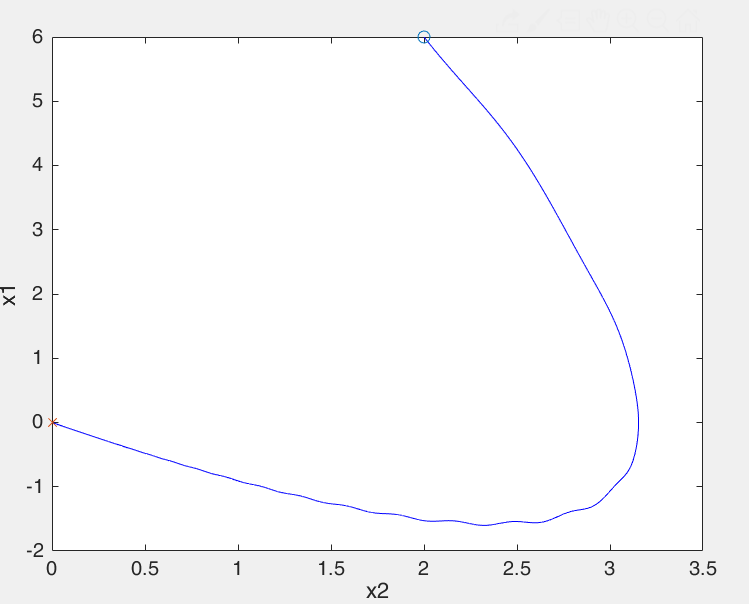
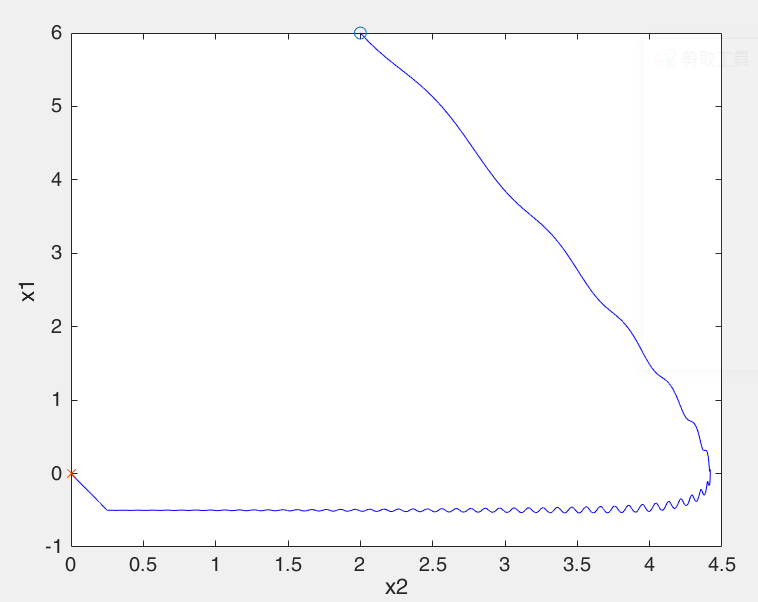
% plot(xarr1,xarr2,'-b');hold on;

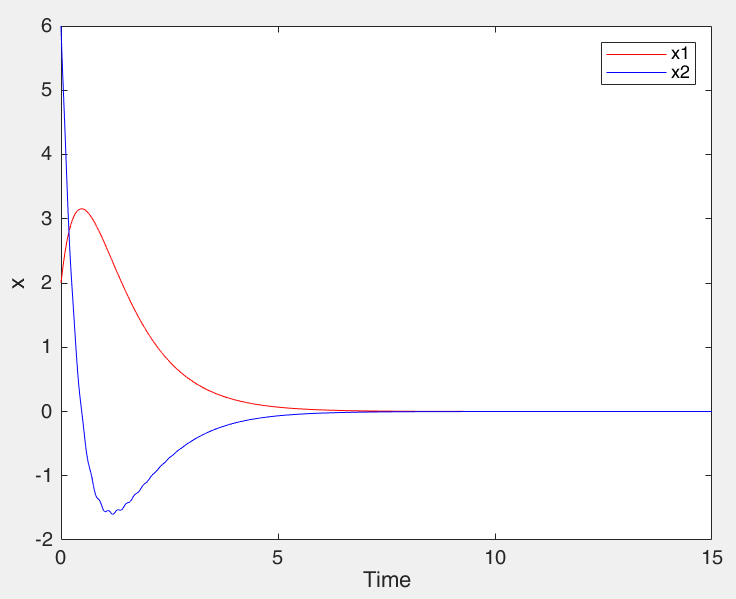
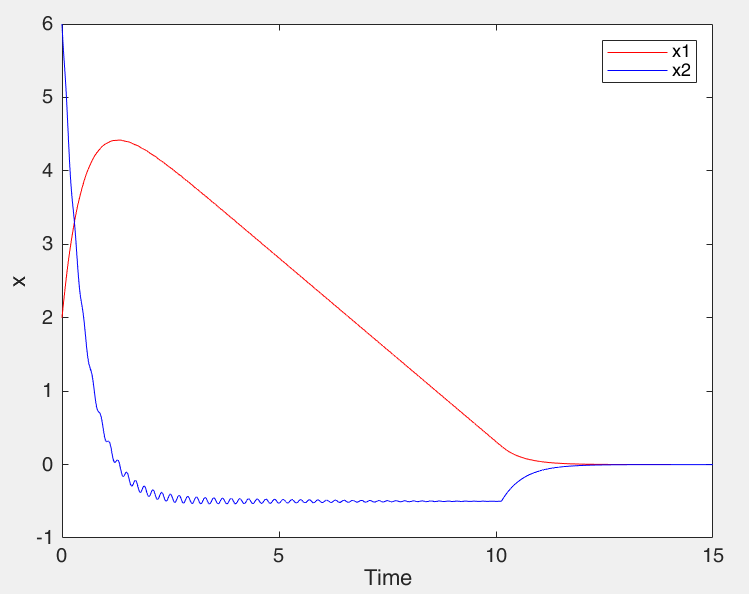
% plot(xarr1(1),xarr2(1),'o',xarr1(i+1),xarr2(i+1),'x');

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

**b)**

在資料疊代處的x2\_dot加上 0.1\*sin((a+b)\*pi\*t)在x\_2速率上產生震動。左下圖以及左上圖為k=1控制；右下圖以及右上圖是k=-s控制。由下圖可以觀測可以發現，經由k=s的穩定控制後震盪變的微弱。





**主程式:**

%% Student ID

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

a = fix(n/10);

b = rem(n,10);

k = 1

delt = 0.0001;

totTime = 15 ;

totalStep = totTime/delt ;

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

x = [2 6]'

%(xarra1,xarra2) : ­ì¥»ªº¨t²Î

xarr1 = [0,0,0,0]';xarr2 = xarr1; %x®y¼Ð¦ì¸mªì©l %y®y¼Ð¦ì¸mªì©l

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

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

for i=1:totalStep

Bx = [0;(x\_1^2+x\_2^2)];

Ax = [ 0 1;

b\*x\_2^3 a\*x\_1^3

];

A = 1;

B = 2;

s = A\*x\_2+B\*x\_1;

if s>0

%k = 1\*s;%解開註解做k=s控制，註解為k=1

ux = -1\*((A\*(a\*x\_1^3\*x\_2 + b\*x\_1\*x\_2^3) + B\*x\_2) + k)/(x\_1^2+x\_2^2);

else

%k = -1\*s;%解開註解做k=s控制，註解為k=1

ux = -1\*((A\*(a\*x\_1^3\*x\_2 + b\*x\_1\*x\_2^3) + B\*x\_2) - k)/(x\_1^2+x\_2^2);

end

xNew = zeros(size(x));

ut = Bx\*ux;

dt = Bx\*(0.1\*sin((a+b)\*pi\*tarr(i)));

x\_dot = Ax\*x+ut+dt;

xNew = x + x\_dot\*delt;

xN = xNew;

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

x = xN;

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

end

figure(1);plot(tarr,xarr1,'-r',tarr,xarr2,'-b');legend('x1','x2');

ylabel('x');xlabel('Time');

figure(2)

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

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

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

**c)**

學號之ab新假設

a\_h = a b\_h = b

a = a\_h + a\_delta b = b\_h + b\_delta

與a)題相仿

當 s\_dot >0

u = -1\*((A\*(a\_h\*x1^3\*x2 + b\_h\*x1\*x2^3) + B\*x2) + k)/(x1^2+x2^2);

當 s\_dot <0

u = -1\*((A\*(a\_h\*x1^3\*x2 + b\_h\*x1\*x2^3) + B\*x2) - k)/(x1^2+x2^2);

所以會殘留部分delta產生的沒有被消除，所以需要調整k值進行強制調控，使系統穩定。

可得x2\_dot = (a\_delta\*x1^3\*x2 + b\_delta \*x1\*x2^3) - B\*x2 - k

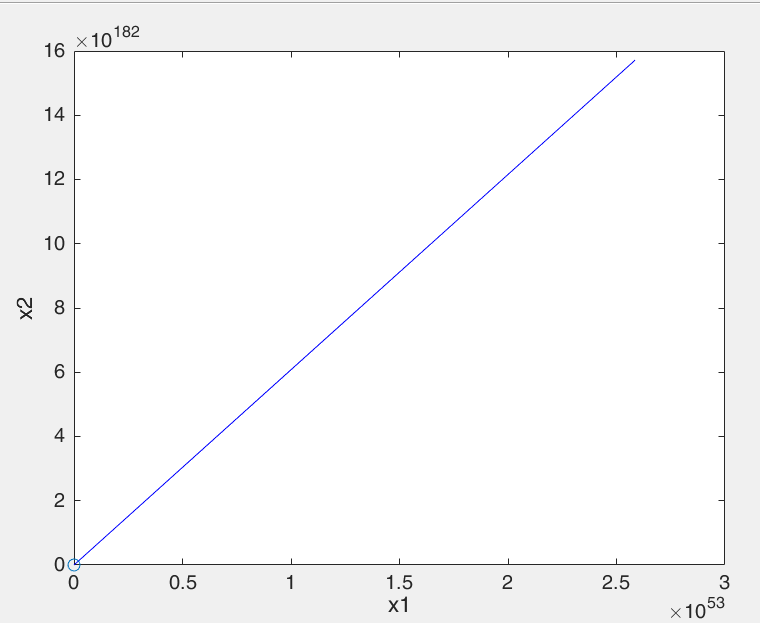
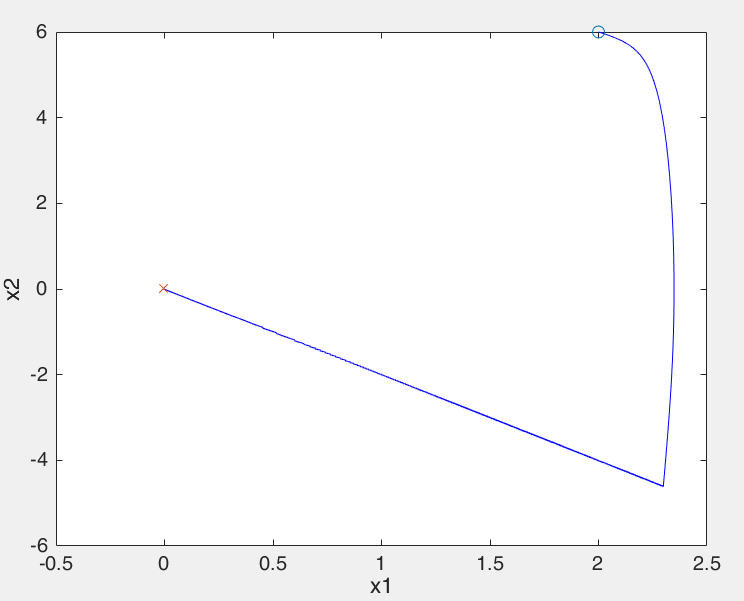
為使系統穩定，所以需控制至x2\_dot再任意時間都小於零，而B\*x2為欲加上去穩定用系統，要完整保留，所以不考慮進去。

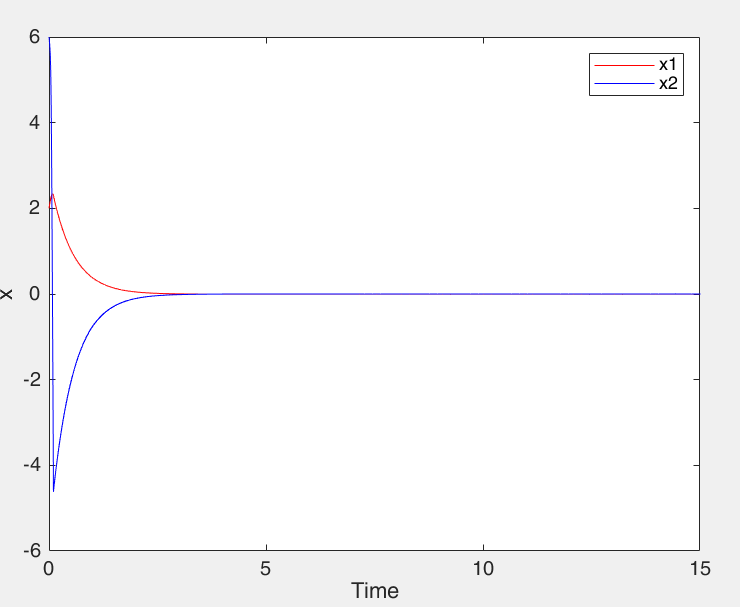
k > (a\_delta\*x1^3\*x2 + b\_delta \*x1\*x2^3)

由於初始值的x2最大，可得k的最大值，此圖初始座標為(2,6)，所以:

k > k0 = 0.2\*6\*2^3 + 0.4\*2\*6^3 =182.4

可得下左圖為k=185以及下右圖為k=178的狀態圖





用Lyapunov Func.設計方式解決問題減少顫振

令V = 0.5s^2

V\_dot = s\_dot\*s

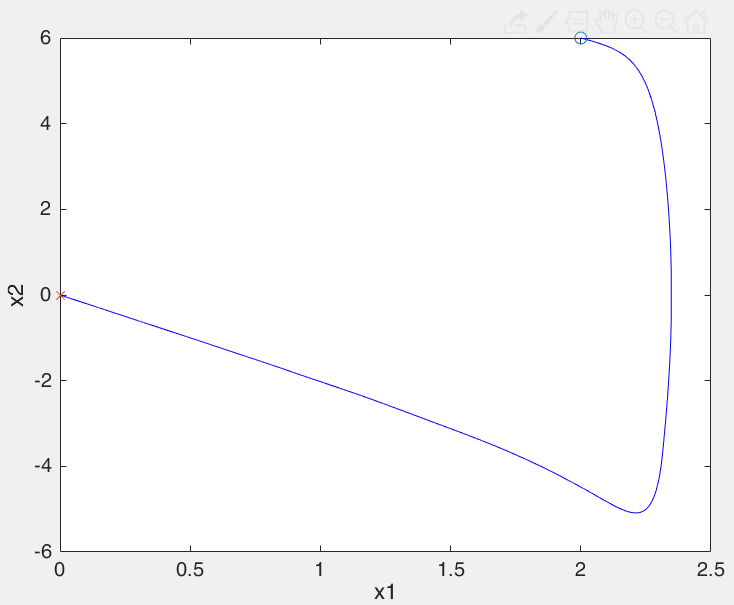
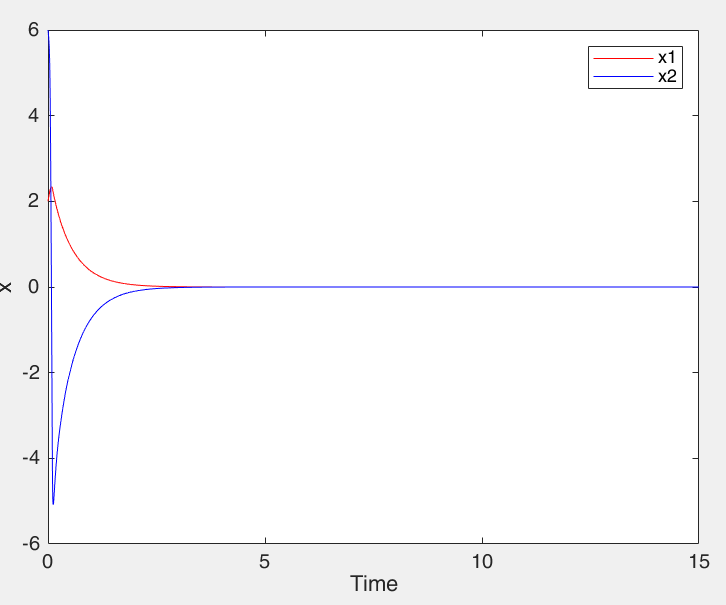
設定 s\_dot = -ks 可得 V\_dot = -k\*s^2

u = -1\*((A\*(a\*x1^3\*x2 + b\*x1\*x2^3) + B\*x2) + k\***sat(s)**)/(x1^2+x2^2);

**sat(s) =** { 1 , s >= 1

s , -1 < s <1

-1 , s <= -1 }

**程式碼**

%% Student ID

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

a\_h = fix(n/10);

b\_h = rem(n,10);

K = 180;

a\_delt = 0.2;

b\_delt = 0.4;

a = a\_h + a\_delt;

b = b\_h + b\_delt;

delt = 0.0001;

totTime = 15 ;

totalStep = totTime/delt ;

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

x = [2 6]'

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

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

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

for i=1:totalStep

Bx = [0;(x\_1^2+x\_2^2)];

Ax = [ 0 1;

b\*x\_2^3 a\*x\_1^3

];

A = 1;

B = 2;

s = A\*x\_2 + B\*x\_1;

if s>0

if s>=1

k = K;

% else %註解為k=k 解掉註解為k=K\*s

% k = K\*s;

end

ux = -1\*((A\*(a\_h\*x\_1^3\*x\_2 + b\_h\*x\_1\*x\_2^3) + B\*x\_2) + k)/(x\_1^2+x\_2^2);

else

if s<=-1

k = K;

% else %註解為k=k 解掉註解為k=K\*s

% k = -1\*K\*s;

end

ux = -1\*((A\*(a\_h\*x\_1^3\*x\_2 + b\_h\*x\_1\*x\_2^3) + B\*x\_2) - k)/(x\_1^2+x\_2^2);

end

xNew = zeros(size(x));

ut = Bx\*ux;

dt = Bx\*(0.1\*sin((a+b)\*pi\*tarr(i)));

x\_dot = Ax\*x+ut+dt;

xNew = x + x\_dot\*delt;

xN = xNew;

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

x = xN;

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

end

figure(1);plot(tarr,xarr1,'-r',tarr,xarr2,'-b');legend('x1','x2');

xlabel('Time');ylabel('x');

figure(2)

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

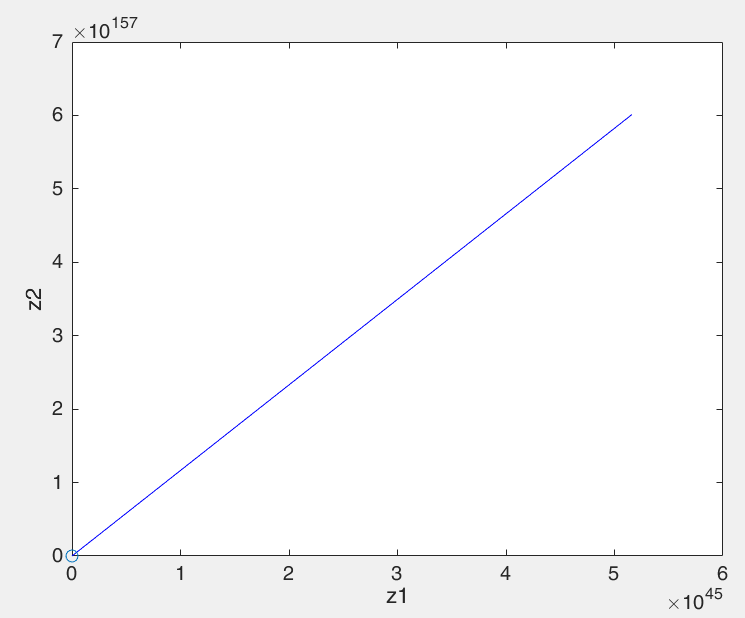
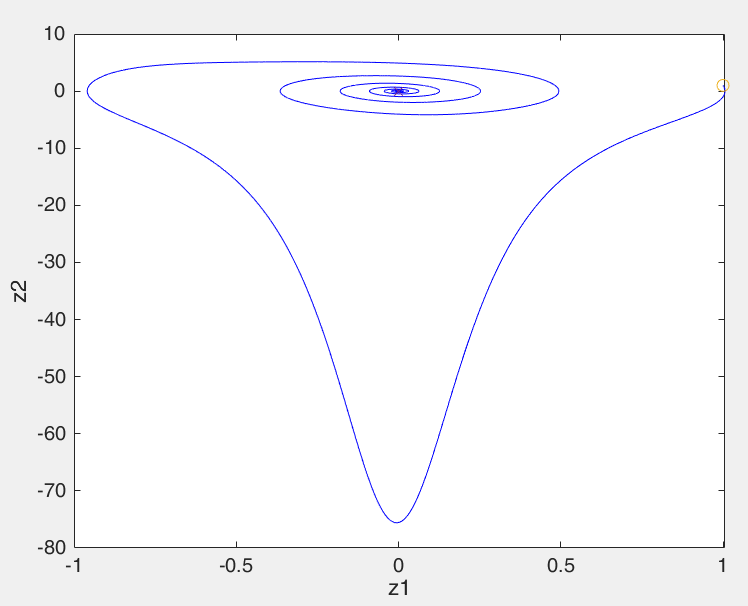
plot(xarr1(1),xarr2(1),'o',xarr1(i+1),xarr2(i+1),'x');

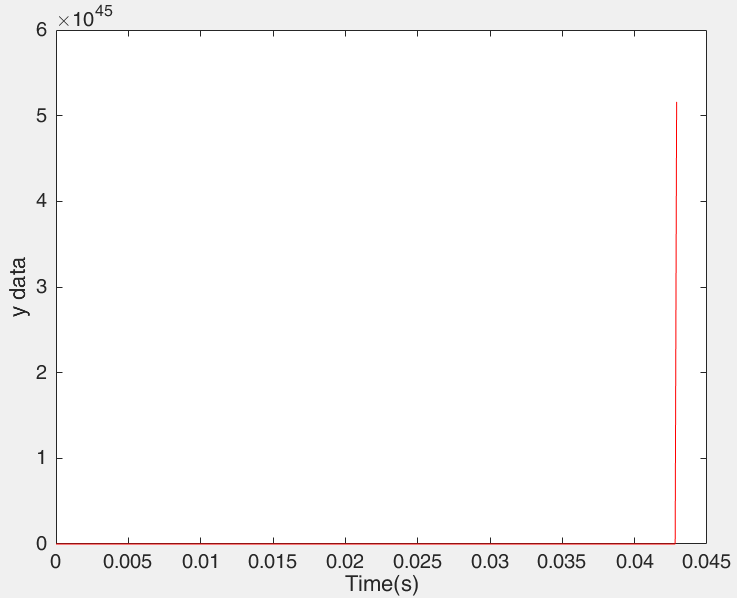
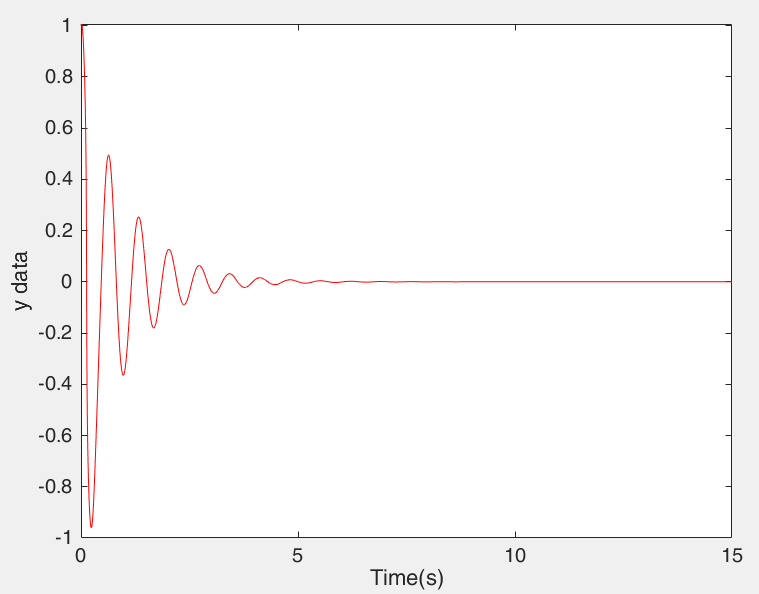
xlabel('x1');ylabel('x2');

**FB Linearization**

此題的狀態x = z剛好相等，所以指出Z圖以及y圖，由下圖可知道FB linearization 在微小誤差變化影響甚巨

**X0 = [1;1] X0 = [2;6]**





%% 3\_FB linearization

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

a\_h = fix(n/10);

b\_h = rem(n,10);

a\_delt = 0.2;

b\_delt = 0.4;

a = a\_h + a\_delt;

b = b\_h + b\_delt;

delt = 0.0001;

totTime = 15 ;

totalStep = totTime/delt ;

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

x = [1 1]'

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) = x(2);

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

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

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

for i=1:totalStep

Bx = [0;(x\_1^2+x\_2^2)];

Ax = [ 0 1;

b\*x\_2^3 a\*x\_1^3

];

A = 1;

B = 2;

ux = -1\*(A\*(a\_h\*x\_1^3\*x\_2 + b\_h\*x\_1\*x\_2^3) + 82\*z\_1 + 2\*z\_2)/(x\_1^2+x\_2^2);

xNew = zeros(size(x));

ut = Bx\*ux;

dt = Bx\*(0.1\*sin((a+b)\*pi\*tarr(i)));

x\_dot = Ax\*x+ut+dt;

xNew = x + x\_dot\*delt;

xN = xNew;

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\_2;

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

y(i+1) = z\_1;

end

figure(1);plot(tarr,xarr1,'-r',tarr,xarr2,'-b');legend('x1','x2');xlabel('Time(s)');ylabel('x data');

figure(2);plot(tarr,zarr1,'-r',tarr,zarr2,'-b');legend('z1','z2');xlabel('Time(s)');ylabel('z data');

figure(3)

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

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

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

figure(3)

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

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

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

figure(5)

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