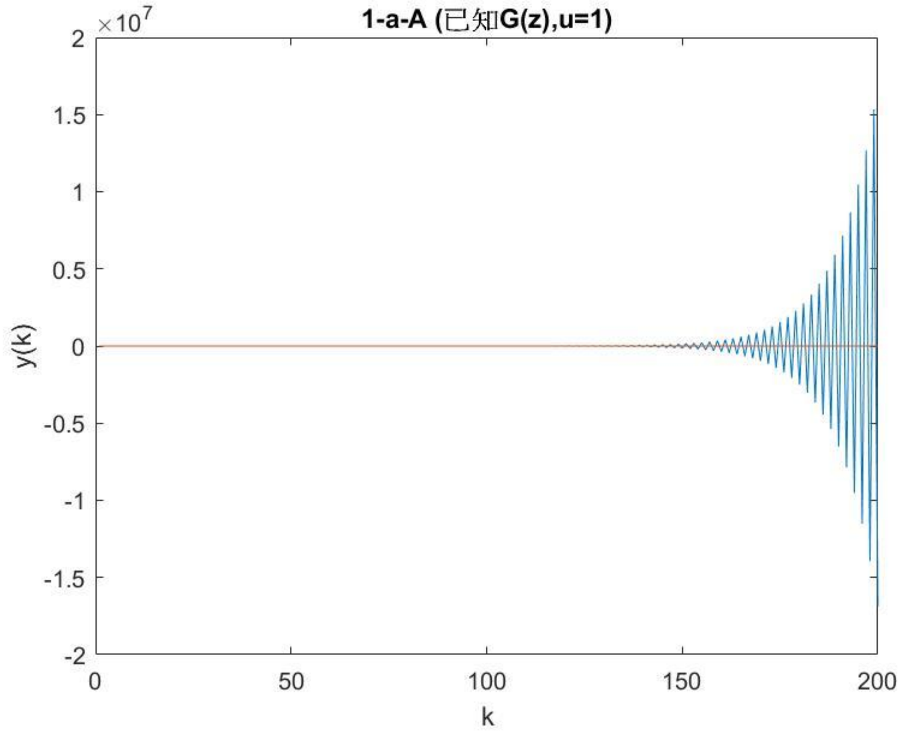
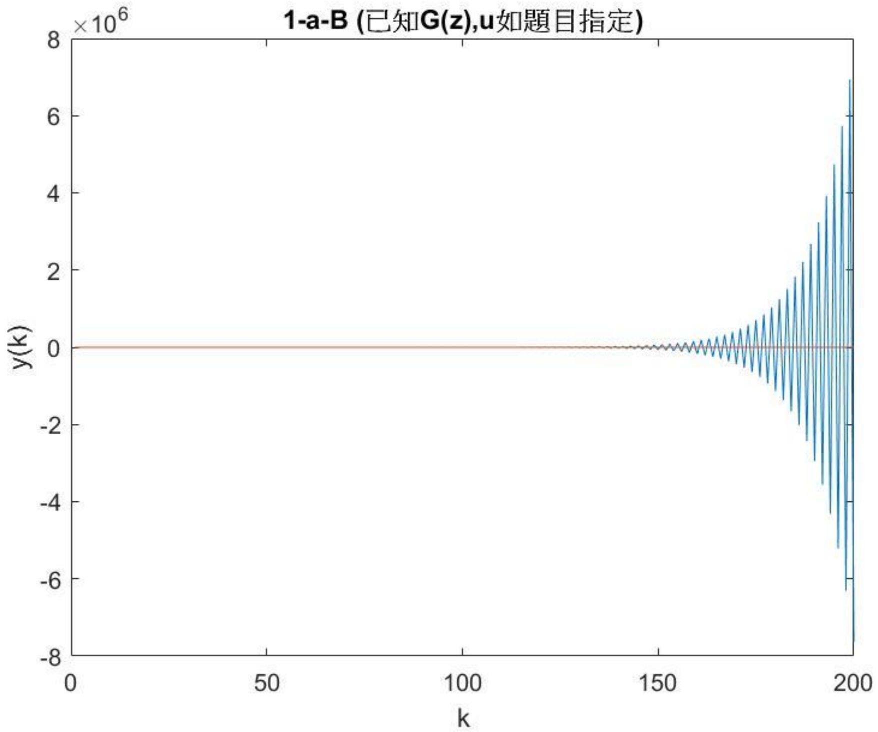
現代控制理論 HW7

104303206 黃筱晴

1-a-A 已知 G(z)求輸出 (u=1)



1-a-B 已知 G(z)求輸出 (u=一堆弦波的合成)



1-b-A

a1=0.298118 (error=-0.001882)

a2=-0.881307 (error=-0.001307)

b1=0.940994 (error=0.040994)

b2=0.543759 (error=-0.056241)

1-b-B

a1=0.299174 (error=-0.000826)

a2=-0.881930 (error=-0.001930)

b1=0.922097 (error=0.022097)

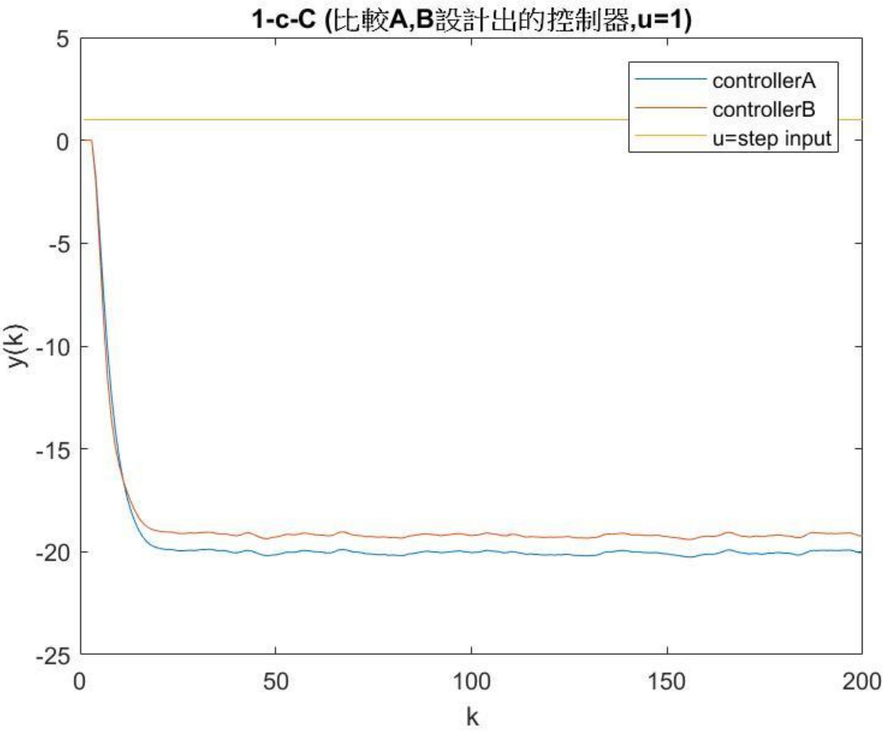
b2=0.575913 (error=-0.024087)

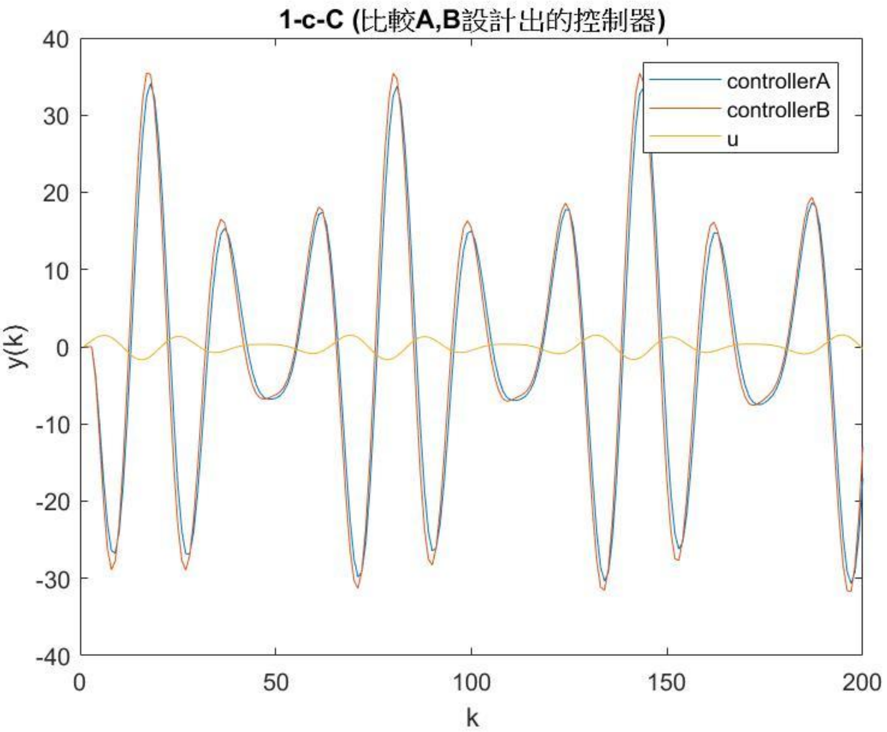
1-b-C

大部分測試結果是 A(unit step 輸入)較精準

Note:在做 1-b-A 的時候，欲使用最佳化方法($\partial J/\partial\theta=0$)求參數。其中計算到$\theta=[\Sigma\varphi(k-1)\varphi(k-1)^T]^{-1}(\Sigma y(k)\varphi(k-1))$時，因為奇異方陣沒辦法算反矩陣卡住。將 u 的第一筆先改成零後解決。但是之後測試過程也常常遇到 Matlab 說太接近奇異方陣的 Warning 訊息。

1-c





Note:當 K 再多一點(千筆以上)兩個控制器因為參數不完全精準，系統都會發散。所以要用下一題(1-d)的 adaptive control。

(1)pole assignment

目標極點位置 0.5+0.5j,0.5-0.5j,0.82

$\varphi(z)=(z-0.5+0.5j)*(z-0.5+0.5j)*(z-0.82)=z^3-1.82z^2+1.32z-0.41$

設計 $C(z)=(\beta_0 z+\beta_1)/(z+\alpha_1)$，單回授系統，令等效開環轉移函數分母

$(z+\alpha_1)(z^2+a_1z+a_2)+(\beta_0z+\beta_1)(b_1z+b_2)=z^3-1.82z^2+1.32z-0.41$。比較係數法。

解聯立方程式矩陣 $\begin{bmatrix} 1 & b1 & 0 \\ a1 & b2 & b1 \\ a2 & 0 & b2 \end{bmatrix}\begin{bmatrix} \alpha 1 \\ \beta 0 \\ \beta 1 \end{bmatrix}=\begin{bmatrix} -1.82-a1 \\ 1.32-a2 \\ -0.41 \end{bmatrix}$ 得到 α_1、β_0、β_1

(2)回授控制程式實作(程式碼於附錄)

<1>等效開環轉移函數$\frac{C(z)G(z)}{1+C(z)G(z)}$

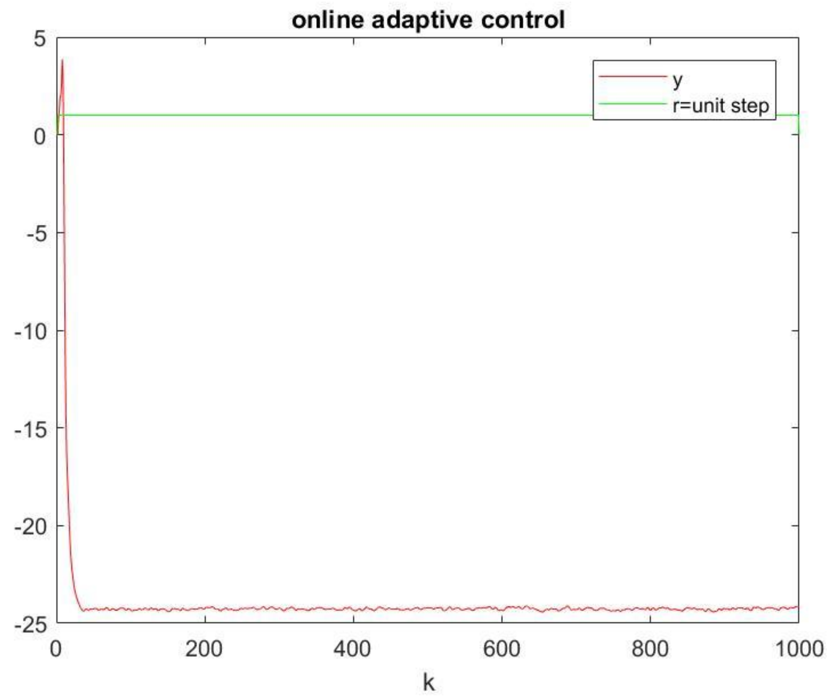
<2>寫一個這種迴圈來疊代

r(k+1)=1;
e(k+1)=r(k+1)-y(k+1); %%這裡好奇怪@@????
$u(k+1)=-\alpha_1u(k)+\beta_0e(k+1)+\beta_1e(k)$;
$y(k+2)=-a_1y(k+1)-a_2y(k)+b_1u(k+1)+b_2u(k)+d(k+2)$;

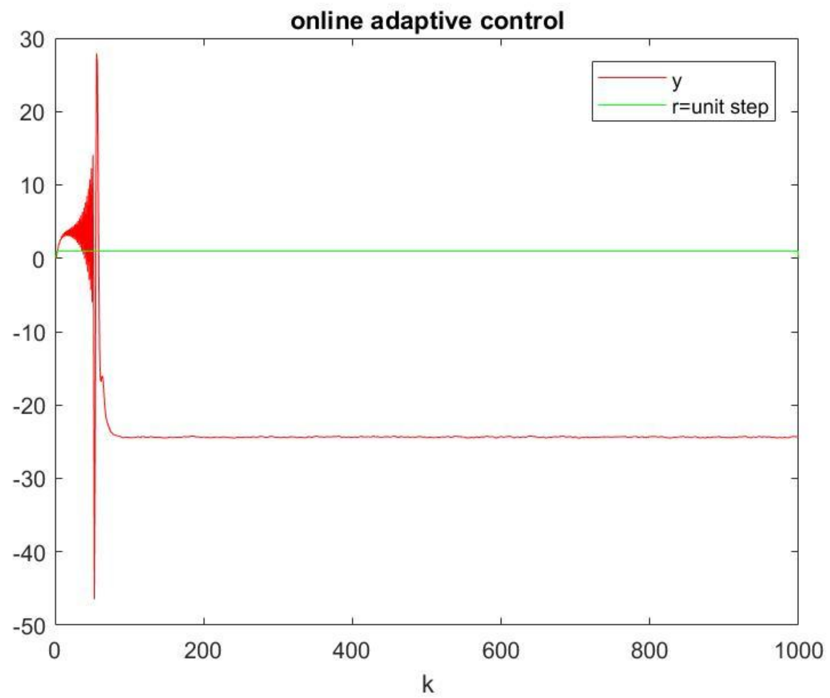
但是剛開始做的時候一直發散，後來反覆嘗試修改步數的地方，終於試出一組答案，能做出和使用等效開環轉移函數相同的結果。

1-d-A

(1)1~5 蒐集數據，6~900 online adaptive control，901~1000 不再修改控制器。

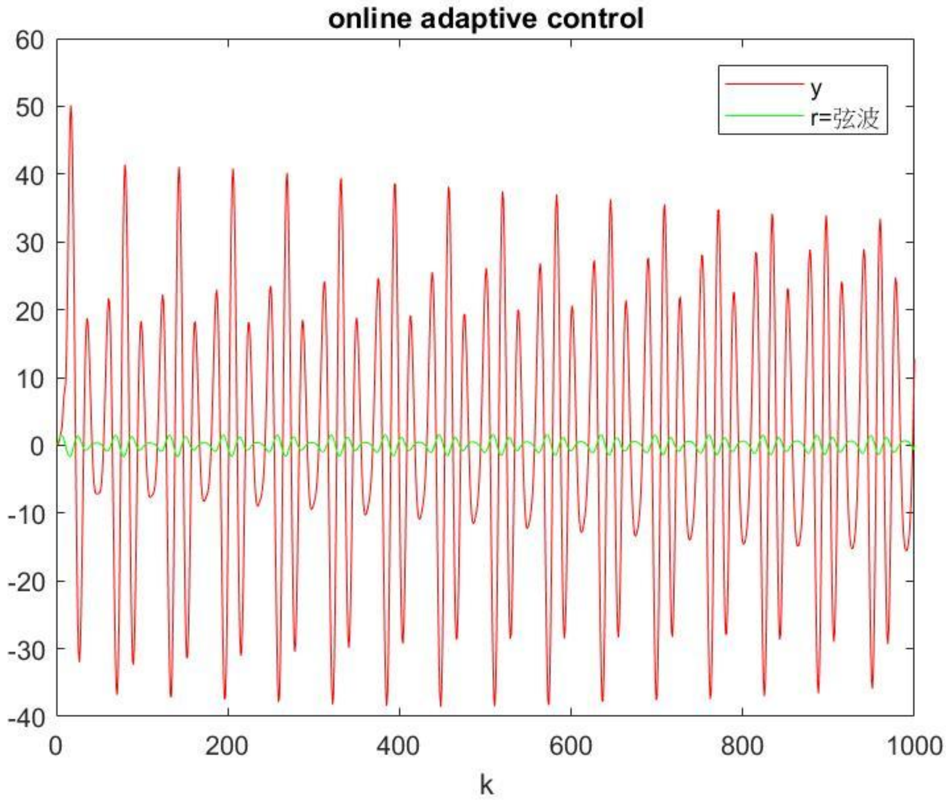


(2)1~50 蒐集數據，51~900 online adaptive control，901~1000 不再修改控制器。

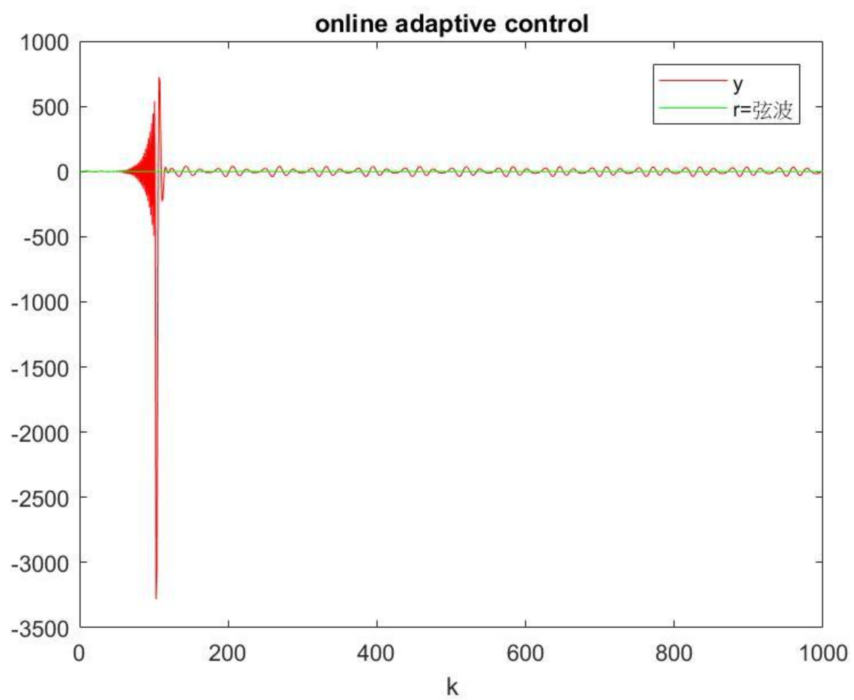


1-d-B

(1)1~5蒐集數據，6~900 online adaptive control，901~1000不再修改控制器。

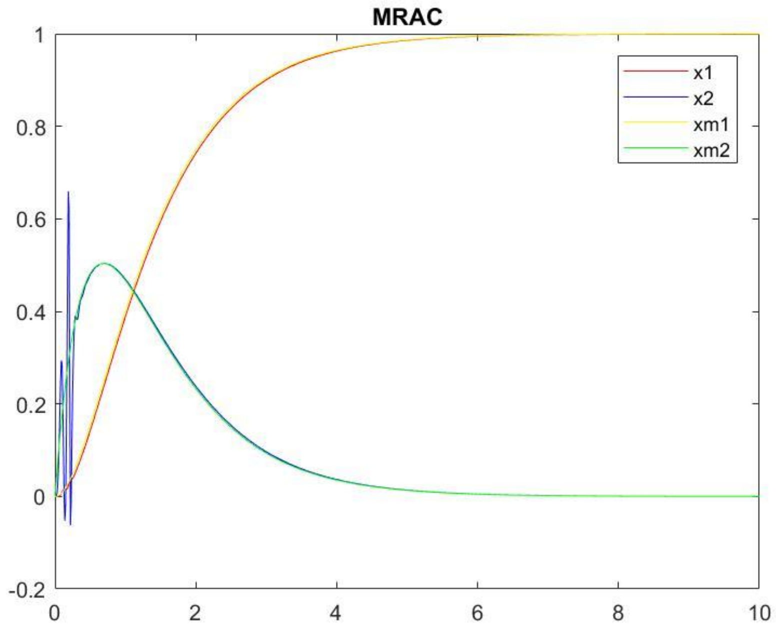


(2)1~100蒐集數據，101~900 online adaptive control，901~1000不再修改控制器。



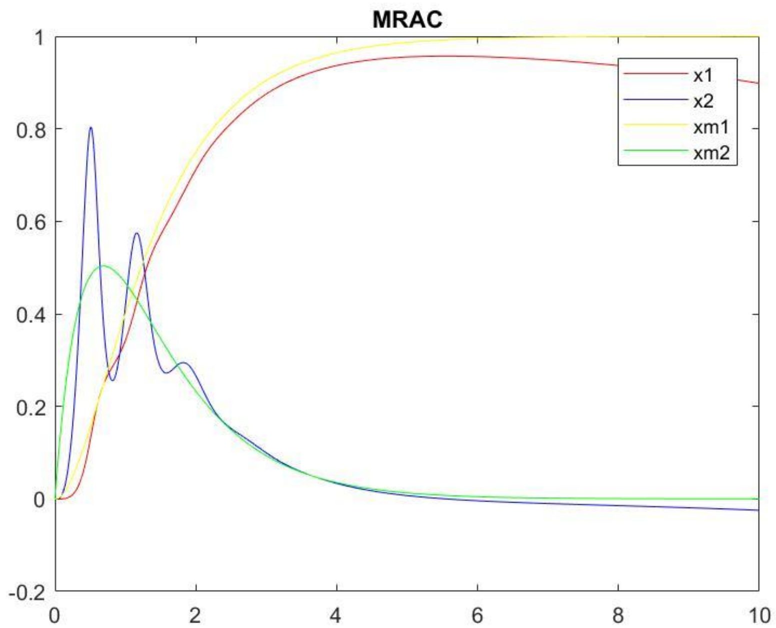
2-A. MRAC unit step 輸入模擬

(1) unit step 對照組 γ_0=0.025 γ_1=0.5 γ_2=0.005 Q_{22}=1000



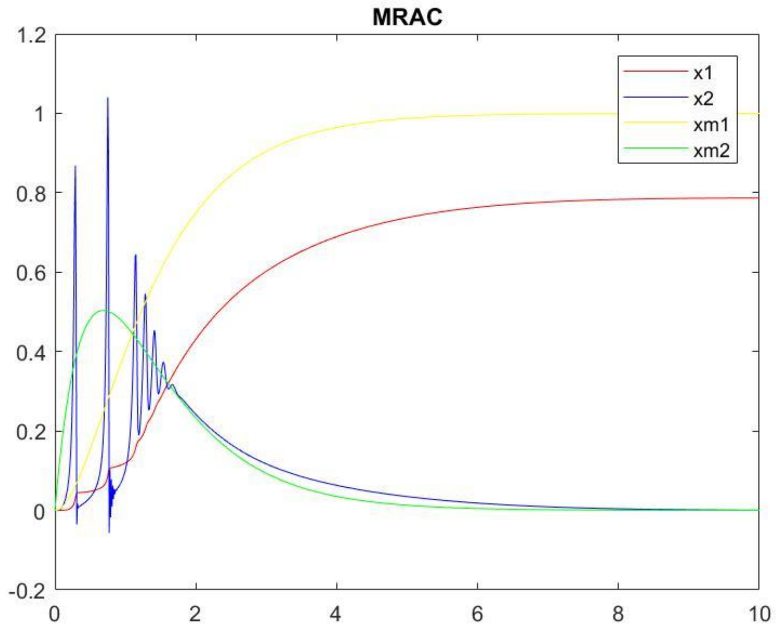
(2)Q=$\begin{bmatrix} 1 & 0 \\ 0 & q \end{bmatrix}$,明顯 Q_{22} 值越大，系統越穩定，若 Q_{22} 值不夠系統會發散。

降低 Q_{22} (γ_0=0.025 γ_1=0.5 γ_2=0.005 Q_{22}=10)



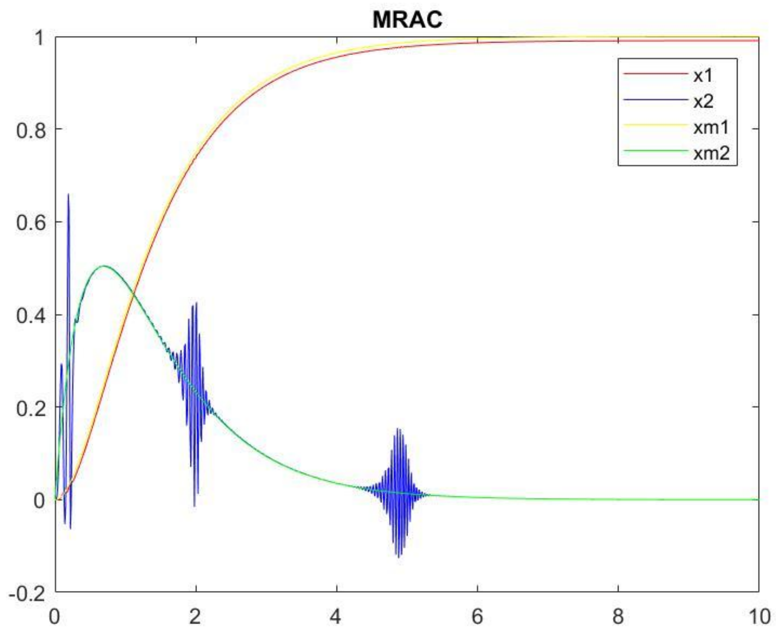
(2) γ_0越大，x1 穩態誤差越糟，x1,x2 震盪情形持續較久。

增大 γ_0 (γ_0=2.5 γ_1=0.5 γ_2=0.005 Q_{22}=10)



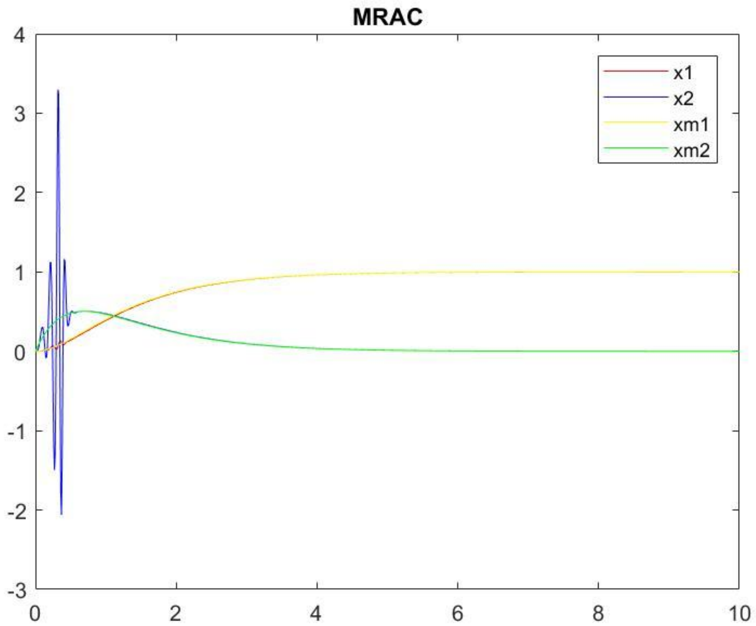
(3) γ_1=越小，x2 震盪越久。貌似越大越好，也不會發散，但是往上調至定值後看起來效果都一樣。

降低 γ_1 (γ_0=0.025 γ_1=0.005 γ_2=0.005 Q_{22}=1000)



(4) γ_2越大，x1、x2 的振幅都會變大也會振比較久，其中 x2 影響非常嚴重。

增加 γ_2 (γ_0=0.025 γ_1=0.5 γ_2=0.5 Q_{22}=1000)

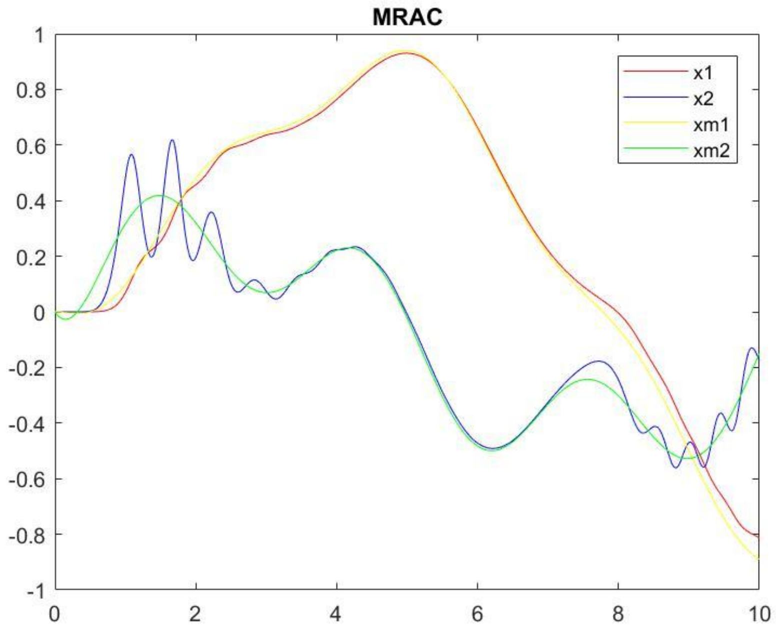


經多次測試結論：

Q_{22}要大、γ_0要小 γ_1可大 γ_2盡量小，但這些參數太小系統都容易發散。
在三個 γ 中 γ_2最為敏感也最重要。

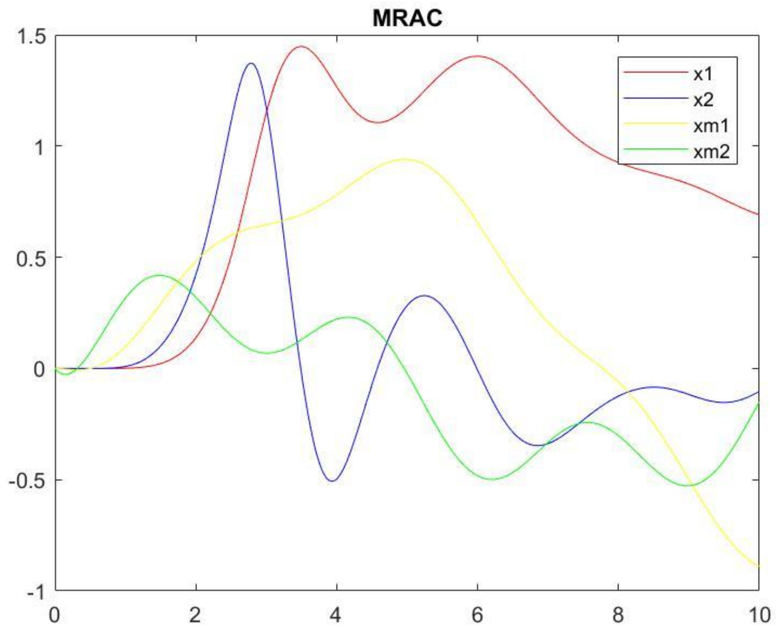
2-B.MRAC 弦波輸入模擬

(1)弦波對照組 $\gamma_0=1$ $\gamma_1=0.5$ $\gamma_2=0.4$ $Q_{22}=1000$



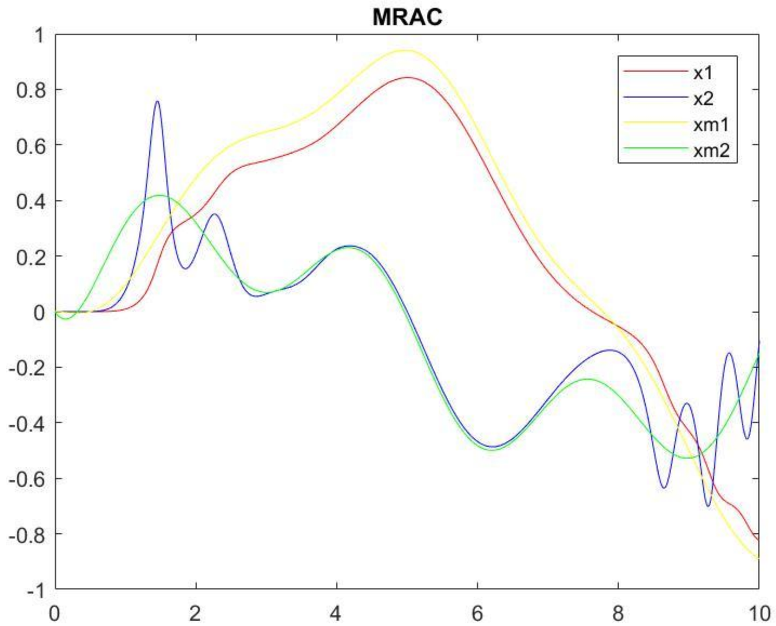
(2)同樣 Q_{22} 越大越穩定

降低 Q_{22} ($\gamma_0=1$ $\gamma_1=0.5$ $\gamma_2=0.4$ $Q_{22}=10$)

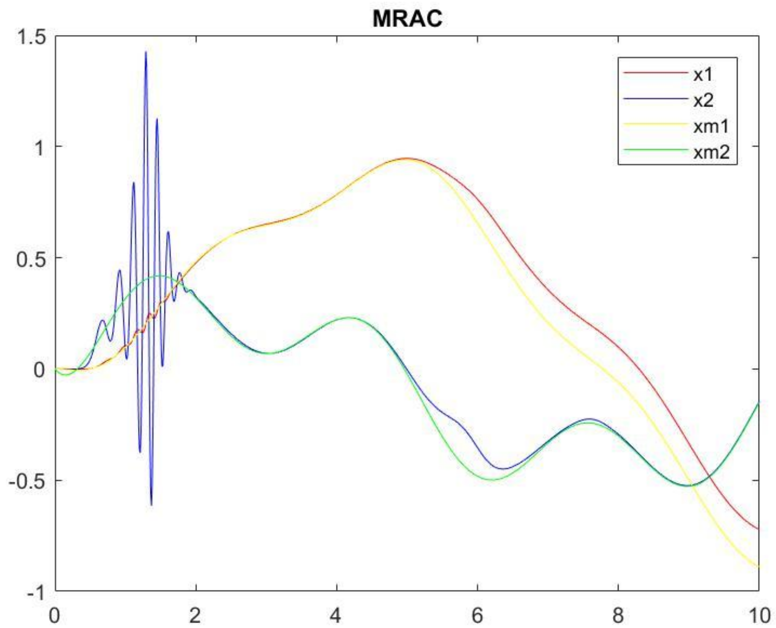


(2)

增加 γ_0=(γ_0=10 γ_1=0.5 γ_2=0.4 Q_{22}=1000)，誤差變大

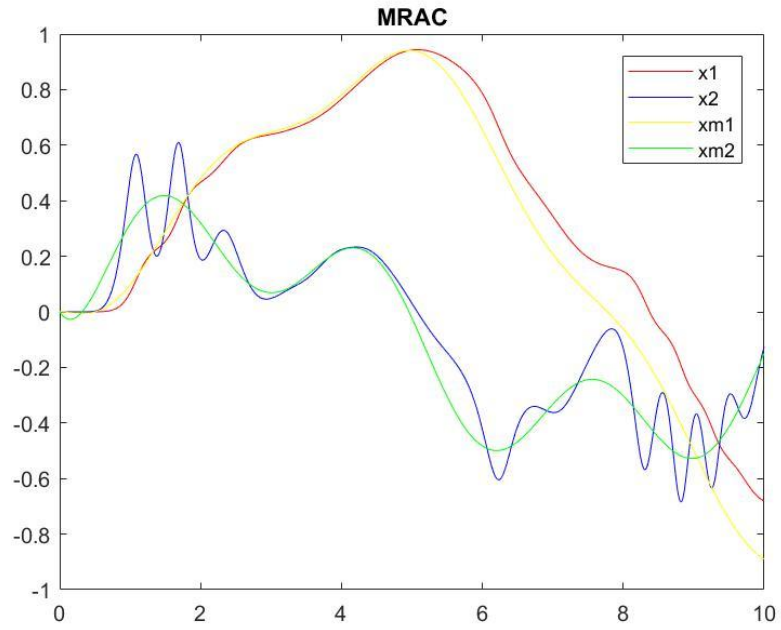


降低 γ_0=(γ_0=0.05 γ_1=0.5 γ_2=0.4 Q_{22}=1000)，暫態響應的震盪變嚴重。

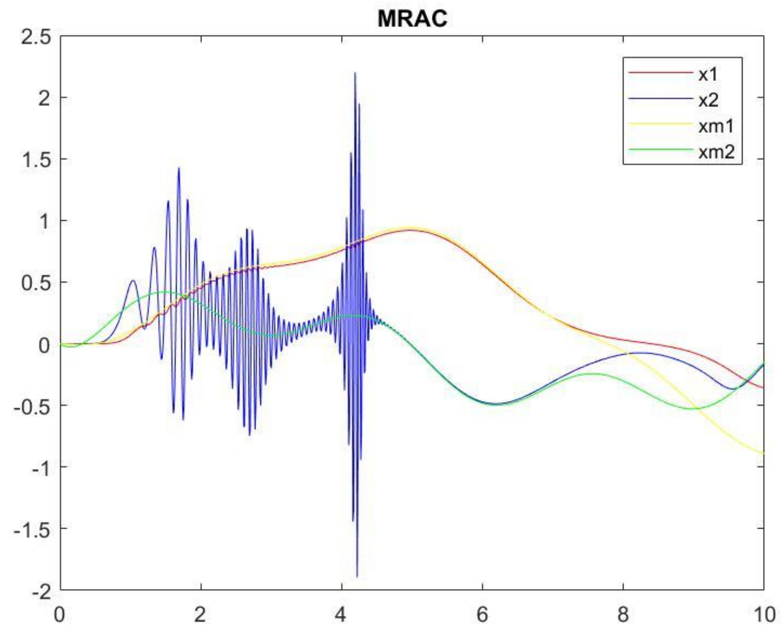


(3)

增加 γ_1 (γ_0=1 γ_1=5 γ_2=0.4 Q_{22}=1000)，誤差變大。

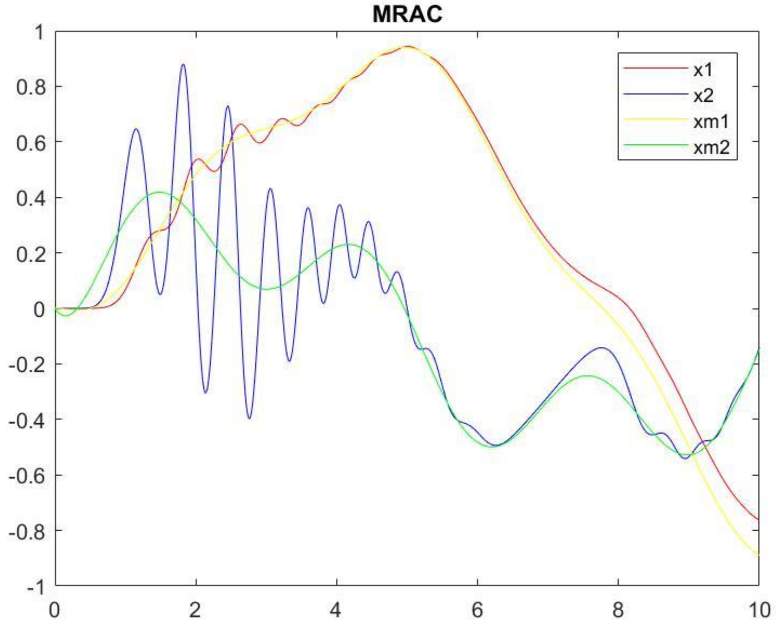


降低 γ_1 (γ_0=1 γ_1=0.005 γ_2=0.4 Q_{22}=1000)，震盪情形持續較久。

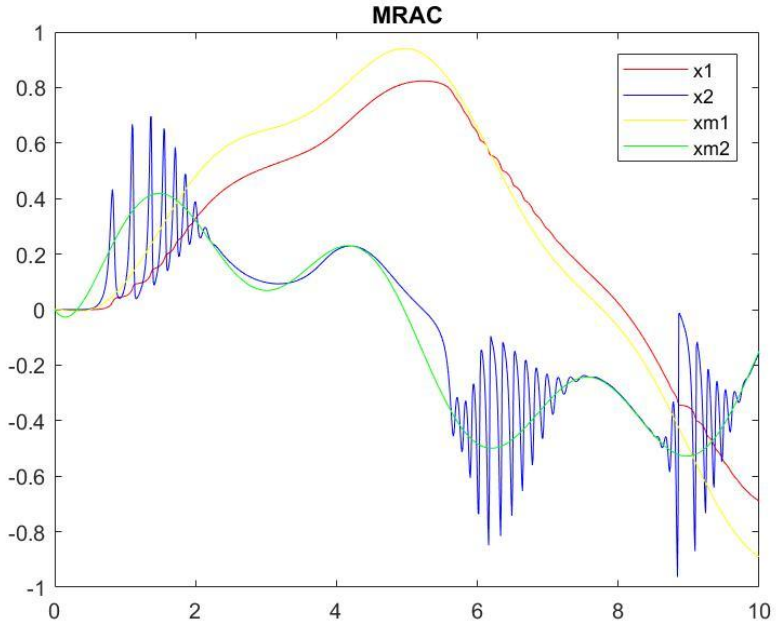


(4)

增加 γ_2 (γ_0=1 γ_1=0.5 γ_2=4 Q_{22}=1000)，X 要花更多時間追上 Xm。



降低 γ_2 (γ_0=1 γ_1=0.5 γ_2=0.005 Q_{22}=1000)，後期震盪變明顯。



經多次測試結論：

Q_{22} 要大，三個 γ 都有其較適當的範圍，不能太大或太小。

附錄(程式碼)

(1)閉迴路控制器

```
%close loop control
%result same as using equivalent open loop transfer func.
clear;clc;
totalStep=1000;
a1=0.3;a2=-0.88;b1=0.9;b2=0.6;%system
for k=1:totalStep
    d(k)=0.01*(rand-0.5);
end

poly=conv([1,-0.82],conv([1,-0.5+0.5i],[1,-0.5-0.5i]));%pole
assignment
A=[1 b1 0;a1 b2 b1;a2 0 b2];
b=[poly(2)-a1;poly(3)-a2;poly(4)];
x=inv(A)*b;
alpha1=x(1);beta0=x(2);beta1=x(3);

y=[1:totalStep]*0;
u=[1:totalStep]*0;
r=[1:totalStep]*0;
e=[1:totalStep]*0;
r(1)=0;
r(2)=1;
r(3)=1;
for k=1:totalStep-2
    r(k+1)=1;
    e(k+1)=r(k+1)-y(k+1);
    u(k+1)=(-alpha1*u(k)+beta0*e(k+1)+beta1*e(k));
    y(k+2)=-a1*y(k+1)-a2*y(k)+b1*u(k+1)+b2*u(k)+d(k+2);
end
figure(1)
plot(y,'b');hold on;plot(u,'g');hold on;plot(r,'y');plot(e,'r');
xlabel("k");
legend('y','u','r','e');
title("³á³á³á§Ú°µ¥X-t¦^±Â¤F (r=unit step)");

```

```
y=[1:totalStep]*0;
u=[1:totalStep]*0;
r=[1:totalStep]*0;
e=[1:totalStep]*0;
r(1)=sin(6*1/20)+0.5*cos(6*1/15+3.2)+0.2*sin(2.57*1/13+1.36);
r(2)=sin(6*2/20)+0.5*cos(6*2/15+3.2)+0.2*sin(2.57*2/13+1.36);
r(3)=sin(6*3/20)+0.5*cos(6*3/15+3.2)+0.2*sin(2.57*3/13+1.36);
for k=1:totalStep-2

r(k+1)=sin(6*(k+1)/20)+0.5*cos(6*(k+1)/15+3.2)+0.2*sin(2.57*(k+1)/13+
1.36);
    e(k+1)=r(k+1)-y(k+1);
    u(k+1)=(-alpha1*u(k)+beta0*e(k+1)+beta1*e(k));
    y(k+2)=-a1*y(k+1)-a2*y(k)+b1*u(k+1)+b2*u(k)+d(k+2);
end
figure(2)
plot(y,'b');hold on;plot(u,'g');hold on;plot(r,'y');plot(e,'r');
xlabel("k");
legend('y','u','r','e');
title("³á³á³á§Ú°µ¥X-t¦^±Â¤F (r=©¶ªi)");
```

(2)閉迴路控制器使用等效開環轉移函數

```
%design C(z) and using equivalent open loop transfer func.
clear;clc;
totalStep=1000;
a1=0.3;a2=-0.88;b1=0.9;b2=0.6;%system
for k=1:totalStep
    d(k)=0.01*(rand-0.5);
end

poly=conv([1,-0.82],conv([1,-0.5+0.5i],[1,-0.5-0.5i]));%pole
assignment
A=[1 b1 0;a1 b2 b1;a2 0 b2];
b=[poly(2)-a1;poly(3)-a2;poly(4)];
x=inv(A)*b;
alpha1=x(1);beta0=x(2);beta1=x(3);
fprintf("alpha1=%f;beta0=%f;beta1=%f;\n",alpha1,beta0,beta1);

num=conv([b1,b2],[beta0,beta1]);%C(z)*G(z)
den=conv([1,a1,a2],[1,alpha1]);%C(z)*G(z)
fb_num=num;%equivalent open loop sys.
fb_den=[0,num]+den;%equivalent open loop sys.

y=[1:totalStep]*0;
u=[1:totalStep]*0;
r=[1:totalStep]*0;
y(1)=0;y(2)=0;y(3)=0;
u(1)=0;u(2)=1;u(3)=1;
for k=1:totalStep-3
    u(k+3)=1;

y(k+3)=-fb_den(2)*y(k+2)-fb_den(3)*y(k+1)-fb_den(4)*y(k)+fb_num(1)*u(
k+2)+fb_num(2)*u(k+1)+fb_num(3)*u(k)+d(k+3);
end
figure(1);
plot(y);hold on;plot(u);
legend('y','u=unit step');
title("design C(z) and using equivalent open loop transfer func.");

```

```
y=[1:totalStep]*0;
u=[1:totalStep]*0;
r=[1:totalStep]*0;
y(1)=0;y(2)=0;y(3)=0;
u(1)=sin(6*1/20)+0.5*cos(6*1/15+3.2)+0.2*sin(2.57*1/13+1.36);
u(2)=sin(6*2/20)+0.5*cos(6*2/15+3.2)+0.2*sin(2.57*2/13+1.36);
u(3)=sin(6*3/20)+0.5*cos(6*3/15+3.2)+0.2*sin(2.57*3/13+1.36);
for k=1:totalStep-3

u(k+3)=sin(6*(k+2)/20)+0.5*cos(6*(k+2)/15+3.2)+0.2*sin(2.57*(k+2)/13+
1.36);

y(k+3)=-fb_den(2)*y(k+2)-fb_den(3)*y(k+1)-fb_den(4)*y(k)+fb_num(1)*u(
k+2)+fb_num(2)*u(k+1)+fb_num(3)*u(k)+d(k+3);
end
figure(2);
plot(y);hold on;plot(u);
legend('y','u=©¶ªi');
title("design C(z) and using equivalent open loop transfer func.");
```

(3)1-a,b,c 總程式碼

```
clear;clc;
totalStep=200;
a1=0.3;a2=-0.88;b1=0.9;b2=0.6;%system
for k=1:totalStep
    d(k)=0.1*(rand-0.5);
end
%1-a-A%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%just run with unit step input
y1A=[1:totalStep]*0;
u1A=[1:totalStep]*0;
y1A(1)=0;y1A(2)=0;u1A(1)=0;u1A(2)=1;
for k=1:totalStep-2
    u1A(k+2)=1;
    y1A(k+2)=-a1*y1A(k+1)-a2*y1A(k)+b1*u1A(k+1)+b2*u1A(k)+d(k+2);
end
figure(1);
plot(y1A);hold on;plot(u1A);
xlabel("k");
ylabel("y(k)");
title("1-a-A (¤wª¾G(z),u=1)");

%1-a-B%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%just run with the given input
y1B=[1:totalStep]*0;
u1B=[1:totalStep]*0;
y1B(1)=0;y1B(2)=0;
u1B(1)=sin(6*1/20)+0.5*cos(6*1/15+3.2)+0.2*sin(2.57*1/13+1.36);
u1B(2)=sin(6*2/20)+0.5*cos(6*2/15+3.2)+0.2*sin(2.57*2/13+1.36);
for k=1:totalStep-2

u1B(k+2)=sin(6*(k+2)/20)+0.5*cos(6*(k+2)/15+3.2)+0.2*sin(2.57*(k+2)/1
3+1.36);
    y1B(k+2)=-a1*y1B(k+1)-a2*y1B(k)+b1*u1B(k+1)+b2*u1B(k)+d(k+2);
end
figure(2);
```

```
plot(y1B);hold on;plot(u1B);
xlabel("k");
ylabel("y(k)");
title("1-a-B (¤wª¾G(z),u¦pÃD¥Ø«ü©w)");


%1-b-A%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%use the optimal method to find the system parameter whit dataA
%find matrix phi%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for k=3:length(y1A)+1
   phiA(k-1,1)=y1A(k-1);
   phiA(k-1,2)=y1A(k-2);
   phiA(k-1,3)=u1A(k-1);
   phiA(k-1,4)=u1A(k-2);
end
%find matrix theta%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tmp1A=[0 0 0 0;0 0 0 0;0 0 0 0;0 0 0 0];
tmp2A=[0 0 0 0];
for k=3:totalStep
   for i=1:4
      for j=1:4
         tmp1A(i,j)=tmp1A(i,j)+phiA(k-1,j)*phiA(k-1,i);
      end
      tmp2A(i)=tmp2A(i)+y1A(k)*phiA(k-1,i);
   end
end
thetaA=inv(tmp1A)*tmp2A';
a1A=-thetaA(1);
a2A=-thetaA(2);
b1A=thetaA(3);
b2A=thetaA(4);

%1-b-B%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%use the optimal method to find the system parameter whit dataB
%find matrix phi%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for k=3:length(y1B)+1
```

```
    phiB(k-1,1)=y1B(k-1);
    phiB(k-1,2)=y1B(k-2);
    phiB(k-1,3)=u1B(k-1);
    phiB(k-1,4)=u1B(k-2);
end
%find matrix theta%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tmp1B=[0  0  0  0;0  0  0  0;0  0  0  0;0  0  0  0];
tmp2B=[0  0  0  0];
for  k=3:totalStep
    for  i=1:4
        for  j=1:4
            tmp1B(i,j)=tmp1B(i,j)+phiB(k-1,j)*phiB(k-1,i);
        end
        tmp2B(i)=tmp2B(i)+y1B(k)*phiB(k-1,i);
    end
end
thetaB=inv(tmp1B)*tmp2B';
a1B=-thetaB(1);
a2B=-thetaB(2);
b1B=thetaB(3);
b2B=thetaB(4);

%1-b-C%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%compare the ans
fprintf("Ans from A:\n");
fprintf("a1=%f (error=%2f )\n",a1A,(a1A-a1));
fprintf("a2=%f (error=%2f )\n",a2A,(a2A-a2));
fprintf("b1=%f (error=%2f )\n",b1A,(b1A-b1));
fprintf("b2=%f (error=%2f )\n",b2A,(b2A-b2));
fprintf("Ans from B:\n");
fprintf("a1=%f (error=%2f )\n",a1B,(a1B-a1));
fprintf("a2=%f (error=%2f )\n",a2B,(a2B-a2));
fprintf("b1=%f (error=%2f )\n",b1B,(b1B-b1));
fprintf("b2=%f (error=%2f )\n",b2B,(b2B-b2));

%1-c-A%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
```

```
%pole assignment
poly=conv([1,-0.82],conv([1,-0.5+0.5i],[1,-0.5-0.5i]));%characteristi
c poly.
const=[1 b1A 0;a1A b2A b1A;a2A 0 b2A];
b=[poly(2)-a1A;poly(3)-a2A;poly(4)];
x=inv(const)*b;
alpha1A=x(1);beta0A=x(2);beta1A=x(3);
%1-c-B%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%pole assignment
poly=conv([1,-0.82],conv([1,-0.5+0.5i],[1,-0.5-0.5i]));%characteristi
c poly.
const=[1 b1B 0;a1B b2B b1B;a2B 0 b2B];
b=[poly(2)-a1B;poly(3)-a2B;poly(4)];
x=inv(const)*b;
alpha1B=x(1);beta0B=x(2);beta1B=x(3);


%1-c-C%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%compare the control design from dataA and dataB
%controller A
numA=conv([b1,b2],[beta0A,beta1A]);%C(z)*G(z)
denA=conv([1,a1,a2],[1,alpha1A]);%C(z)*G(z)
fb_numA=numA;%equivalent open loop sys.
fb_denA=[0,numA]+denA;%equivalent open loop sys.
%controller B
numB=conv([b1,b2],[beta0B,beta1B]);%C(z)*G(z)
denB=conv([1,a1,a2],[1,alpha1B]);%C(z)*G(z)
fb_numB=numB;%equivalent open loop sys.
fb_denB=[0,numB]+denB;%equivalent open loop sys.

%controller A with unit step input
y3A=[1:totalStep]*0;
u3A=[1:totalStep]*0;
y3A(1)=0;y3A(2)=0;y3A(3)=0;
u3A(1)=1;u3A(2)=1;u3A(3)=1;
for k=1:totalStep-3
```

```
    u3A(k+3)=1;

y3A(k+3)=-fb_denA(2)*y3A(k+2)-fb_denA(3)*y3A(k+1)-fb_denA(4)*y3A(k)+f
b_numA(1)*u3A(k+2)+fb_numA(2)*u3A(k+1)+fb_numA(3)*u3A(k)+d(k+3);
end
%controller B with unit step input
y3B=[1:totalStep]*0;
u3B=[1:totalStep]*0;
y3B(1)=0;y3B(2)=0;y3B(3)=0;
u3B(1)=1;u3B(2)=1;u3B(3)=1;
for k=1:totalStep-3
    u3B(k+3)=1;

y3B(k+3)=-fb_denB(2)*y3B(k+2)-fb_denB(3)*y3B(k+1)-fb_denB(4)*y3B(k)+f
b_numB(1)*u3B(k+2)+fb_numB(2)*u3B(k+1)+fb_numB(3)*u3B(k)+d(k+3);
end
figure(3);
plot(y3A);hold on;plot(y3B);hold on;plot(u3B);
xlabel("k");
ylabel("y(k)");
legend('controllerA','controllerB','u=step input');
title("1-c-C (¤ñ¸ûA,B³]-p¥Xª°±±¨î¾¹,u=1)");

%controller A with given input
y3A=[1:totalStep]*0;
u3A=[1:totalStep]*0;
y3A(1)=0;y3A(2)=0;y3A(3)=0;
u3A(1)=sin(6*1/20)+0.5*cos(6*1/15+3.2)+0.2*sin(2.57*1/13+1.36);
u3A(2)=sin(6*2/20)+0.5*cos(6*2/15+3.2)+0.2*sin(2.57*2/13+1.36);
u3A(3)=sin(6*3/20)+0.5*cos(6*3/15+3.2)+0.2*sin(2.57*3/13+1.36);
for k=1:totalStep-3

u3A(k+3)=sin(6*(k+3)/20)+0.5*cos(6*(k+3)/15+3.2)+0.2*sin(2.57*(k+3)/1
3+1.36);

y3A(k+3)=-fb_denA(2)*y3A(k+2)-fb_denA(3)*y3A(k+1)-fb_denA(4)*y3A(k)+f
b_numA(1)*u3A(k+2)+fb_numA(2)*u3A(k+1)+fb_numA(3)*u3A(k)+d(k+3);
end
```

```
%controller B with given input
y3B=[1:totalStep]*0;
u3B=[1:totalStep]*0;
y3B(1)=0;y3B(2)=0;y3B(3)=0;
u3B(1)=sin(6*1/20)+0.5*cos(6*1/15+3.2)+0.2*sin(2.57*1/13+1.36);
u3B(2)=sin(6*2/20)+0.5*cos(6*2/15+3.2)+0.2*sin(2.57*2/13+1.36);
u3B(3)=sin(6*3/20)+0.5*cos(6*3/15+3.2)+0.2*sin(2.57*3/13+1.36);
for k=1:totalStep-3

u3B(k+3)=sin(6*(k+3)/20)+0.5*cos(6*(k+3)/15+3.2)+0.2*sin(2.57*(k+3)/1
3+1.36);

y3B(k+3)=-fb_denB(2)*y3B(k+2)-fb_denB(3)*y3B(k+1)-fb_denB(4)*y3B(k)+f
b_numB(1)*u3B(k+2)+fb_numB(2)*u3B(k+1)+fb_numB(3)*u3B(k)+d(k+3);
end
figure(4);
plot(y3A);hold on;plot(y3B);hold on;plot(u3B);
xlabel("k");
ylabel("y(k)");
legend('controllerA','controllerB','u');
title("1-c-C (¤ñ¸ûA,B³]-p¥Xªº±±¨î¾¹)");
```

(4)1-d Online Adaptive Control (A、B 小題大同小異就只附上一個)

```
%online adaptive control
%(unit step input)
clear;clc;
totalStep=1000;step1=50;step2=900;
a1=0.3;a2=-0.88;b1=0.9;b2=0.6;%system
Correctalpha1=4.435315;%C(z) design from the real sys.
Correctbeta0=-7.283683;
Correctbeta1=5.821795;
for k=1:totalStep
    d(k)=0.01*(rand-0.5);
end

y=[1:totalStep]*0;
u=[1:totalStep]*0;
r=[1:totalStep]*0;
e=[1:totalStep]*0;

%k=1~step1%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%just run with unit step input
r(1)=0;r(2)=1;r(3)=1;%unit stpe input
u(1)=0;u(2)=1;u(3)=1;
for k=1:step1
    r(k+2)=1;%unit stpe input
    u(k+2)=r(k+2);
    y(k+2)=-a1*y(k+1)-a2*y(k)+b1*u(k+1)+b2*u(k)+d(k+2);
    
end

%first time to find the system parameter
%find matrix phi%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for k=3:step1
    phiA(k-1,1)=y(k-1);
    phiA(k-1,2)=y(k-2);
    phiA(k-1,3)=u(k-1);
    phiA(k-1,4)=u(k-2);
end
```

```
%find matrix theta%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tmp1A=[0  0  0  0;0  0  0  0;0  0  0  0;0  0  0  0];
tmp2A=[0  0  0  0];
for  k=3:step1
    for  i=1:4
        for  j=1:4
            tmp1A(i,j)=tmp1A(i,j)+phiA(k-1,j)*phiA(k-1,i);
        end
        tmp2A(i)=tmp2A(i)+y(k)*phiA(k-1,i);
    end
end
thetaA=inv(tmp1A)*tmp2A';
a1A=-thetaA(1);
a2A=-thetaA(2);
b1A=thetaA(3);
b2A=thetaA(4);
fprintf("a1=%f  (error=%2f  )\n",a1A,(a1A-a1));
fprintf("a2=%f  (error=%2f  )\n",a2A,(a2A-a2));
fprintf("b1=%f  (error=%2f  )\n",b1A,(b1A-b1));
fprintf("b2=%f  (error=%2f  )\n",b2A,(b2A-b2));

%design  C(z)  with  optimal  parameter  (pole  assignment)
poly=conv([1,-0.82],conv([1,-0.5+0.5i],[1,-0.5-0.5i]));%characteristi
c poly.
const=[1  b1A  0;a1A  b2A  b1A;a2A  0  b2A];
b=[poly(2)-a1A;poly(3)-a2A;poly(4)];
x=inv(const)*b;
alpha1A=x(1);beta0A=x(2);beta1A=x(3);%C(z)  with  optimal  parameter
fprintf("alpha1=%f  (err=%f)  \n",alpha1A,abs(Correctalpha1-alpha1A));
fprintf("beta0=%f  (err=%f)  \n",beta1A,abs(Correctbeta0-beta0A));
fprintf("beta1=%f  (err=%f)  \n",beta1A,abs(Correctbeta1-beta1A));


%k=step1+1~step2%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%online  adaptive
%close  loop  control  &  redesign  C(z)
for  k=step1+1:step2
```

```
fprintf("Now Step:k=%d !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!\n",k);
r(k+1)=1;
e(k+1)=r(k+1)-y(k+1);
u(k+1)=(-alpha1A*u(k)+beta0A*e(k+1)+beta1A*e(k));
y(k+2)=-a1*y(k+1)-a2*y(k)+b1*u(k+1)+b2*u(k)+d(k+2);



%redesign C(z)
%find matrix phi%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i=3:k
    phiA(i-1,1)=y(i-1);
    phiA(i-1,2)=y(i-2);
    phiA(i-1,3)=u(i-1);
    phiA(i-1,4)=u(i-2);
end
%find matrix theta%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tmp1A=[0 0 0 0;0 0 0 0;0 0 0 0;0 0 0 0];
tmp2A=[0 0 0 0];
for m=3:k
    for i=1:4
        for j=1:4
            tmp1A(i,j)=tmp1A(i,j)+phiA(m-1,j)*phiA(m-1,i);
        end
        tmp2A(i)=tmp2A(i)+y(m)*phiA(m-1,i);
    end
end
thetaA=inv(tmp1A)*tmp2A';
a1A=-thetaA(1);
a2A=-thetaA(2);
b1A=thetaA(3);
b2A=thetaA(4);
fprintf("a1=%f (error=%2f )\n",a1A,(a1A-a1));
fprintf("a2=%f (error=%2f )\n",a2A,(a2A-a2));
fprintf("b1=%f (error=%2f )\n",b1A,(b1A-b1));
fprintf("b2=%f (error=%2f )\n",b2A,(b2A-b2));

%design C(z)  (pole assignment)
```

```
poly=conv([1,-0.82],conv([1,-0.5+0.5i],[1,-0.5-0.5i]));%characteristi
c poly.
    const=[1 b1A 0;a1A b2A b1A;a2A 0 b2A];
    b=[poly(2)-a1A;poly(3)-a2A;poly(4)];
    x=inv(const)*b;
    alpha1A=x(1);beta0A=x(2);beta1A=x(3);%C(z) with optimal parameter
    fprintf("alpha1=%f (err=%f)
\n",alpha1A,abs(Correctalpha1-alpha1A));
    fprintf("beta0=%f (err=%f) \n",beta1A,abs(Correctbeta0-beta0A));
    fprintf("beta1=%f (err=%f) \n",beta1A,abs(Correctbeta1-beta1A));
end

%k=step2+1~end%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%close loop control only
for k=step2+1:totalStep-2
    r(k+1)=1;
    e(k+1)=r(k+1)-y(k+1);
    u(k+1)=(-alpha1A*u(k)+beta0A*e(k+1)+beta1A*e(k));
    y(k+2)=-a1*y(k+1)-a2*y(k)+b1*u(k+1)+b2*u(k)+d(k+2);
end

plot(y,'r');hold on;plot(r,'g');
xlabel("k");
legend('y','r=unit step');
title("online adaptive control");
```

(5)MRAC

```
%MRAC
clear;clc;
totaltime=10;
delta=0.01;
totalstep=totaltime/delta;

%model
xm1(1)=0;xm2(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)=-2*xm1(k)-3*xm2(k)+2*r(k);
    xm1(k+1)=xm1(k)+xm1_dot(k)*delta;
    xm2(k+1)=xm2(k)+xm2_dot(k)*delta;
end

%select para.
Q=[1 0;0 1000];
Am=[0 1;-2 -3];bm=2;
b=1;
P=lyap(Am,Q);
gamma0=1;gamma1=0.5;gamma2=4;

theta0(1)=0;theta1(1)=0;theta2(1)=0;
x1(1)=0;x2(1)=0;
for k=1:totalstep
%     r(k)=1;
    r(k)=sin(0.5*k*delta)+0.3*cos(2*k*delta+4);
    u(k)=theta0(k)*r(k)+theta1(k)*x1(k)+theta2(k)*x2(k);

    x1_dot(k)=x2(k);
    x2_dot(k)=0.4*x1(k)+1.8*x2(k)+1*u(k);
    x1(k+1)=x1(k)+x1_dot(k)*delta;
    x2(k+1)=x2(k)+x2_dot(k)*delta;


```

```
    e1(k)=xm1(k)-x1(k);
    e2(k)=xm2(k)-x2(k);
    zeta(k)=0.5*(P(1,2)*e1(k)+P(2,2)*e2(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);

    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;

end


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