

現代控制理論 HW1

104303206 黃筱晴

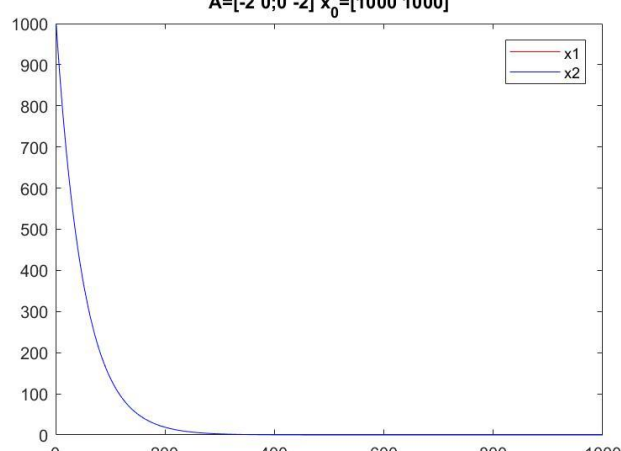
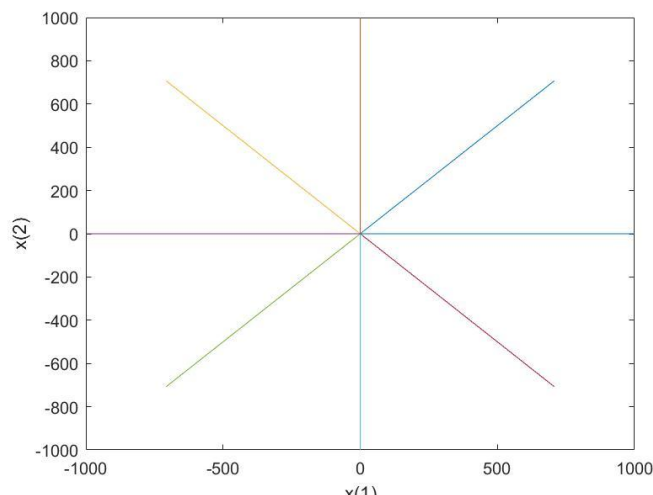
1.題目

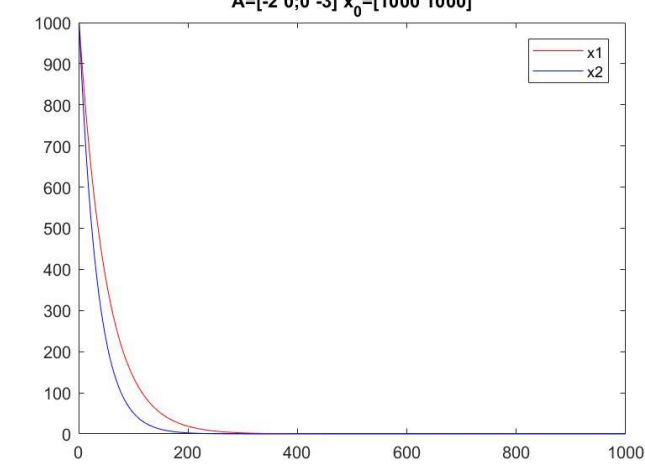
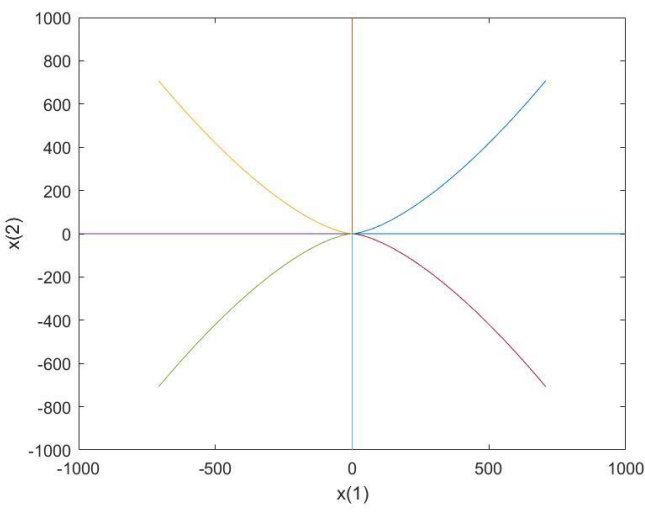
設計出自己的 linear state equation , $\dot{x} = Ax$ 。利用不同的 A , 產生五種 type 的 state portrait , 針對每種 type , 必須畫出所有初始值 $X_0 = [\cos\theta, \sin\theta]^T$ (only for node, saddle point and star)

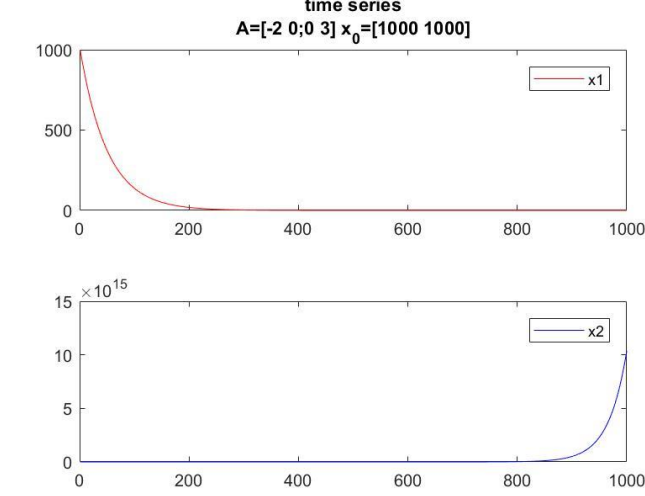
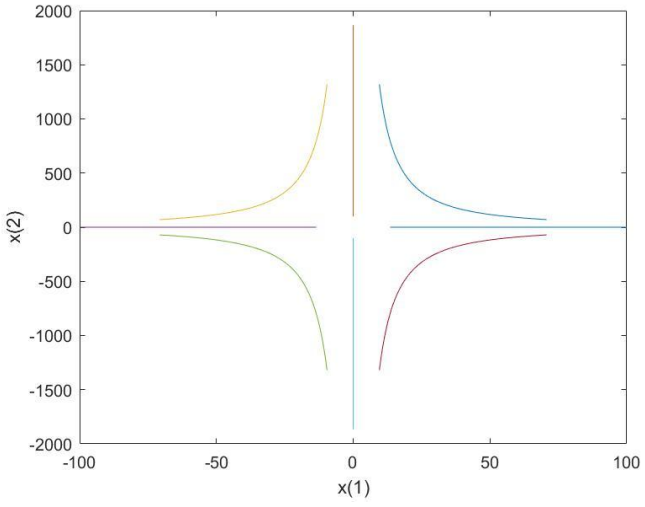
2. 執行結果截圖

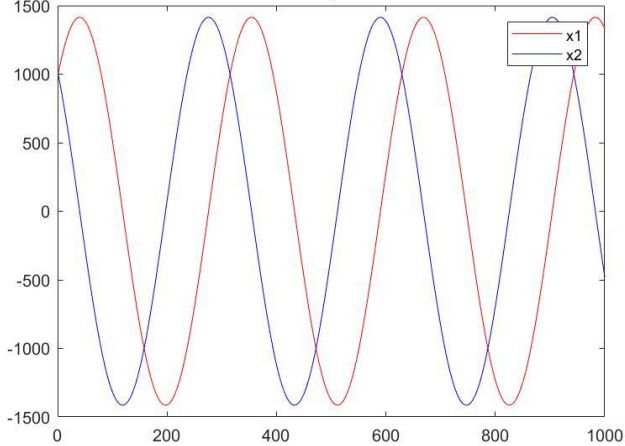
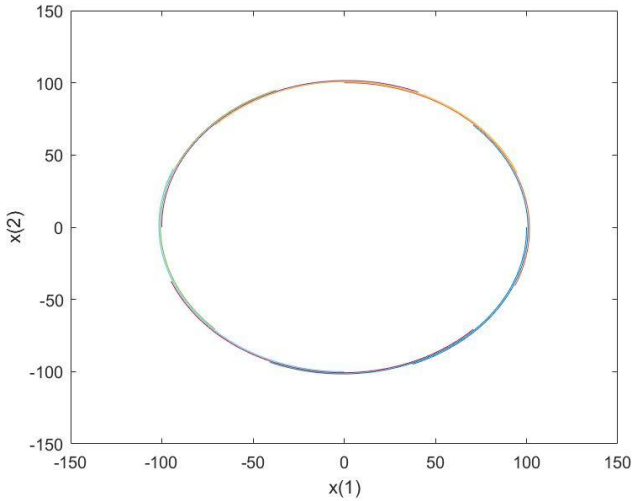
Phase Portrait 的部分都畫出 8 種初始值 $X_0 = [\cos\theta, \sin\theta]^T$, θ 分別為 :

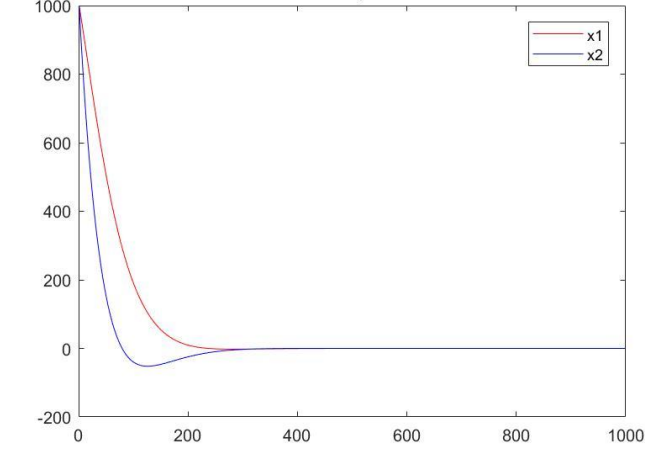
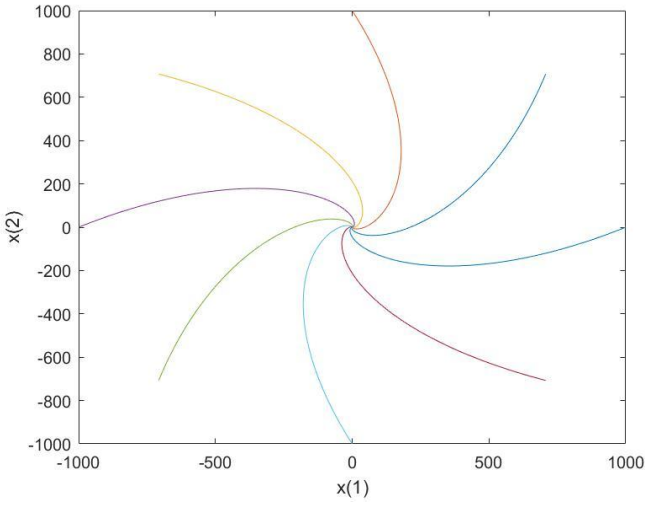
$0, \pi/4, \pi/2, 3\pi/4, \pi, 5\pi/4, 3\pi/2, 7\pi/4$ 。

star	$A = \begin{bmatrix} -2 & 0 \\ 0 & -2 \end{bmatrix}$
Time Series	<p>time series $A = \begin{bmatrix} -2 & 0 \\ 0 & -2 \end{bmatrix}$ $x_0 = [1000 \ 1000]$</p> 
Phase Portrait	
備註	

node	$A=\begin{bmatrix}-2 & 0 \\ 0 & -3\end{bmatrix}$
Time Series	<p>time series $A=\begin{bmatrix}-2 & 0 \\ 0 & -3\end{bmatrix}$ $x_0=[1000 \ 1000]$</p>  <p>The plot shows two decaying exponential curves starting at 1000. The blue curve (x2) reaches zero faster than the red curve (x1). The x-axis represents time from 0 to 1000, and the y-axis represents the state variables from 0 to 1000.</p>
Phase Portrait	 <p>The phase portrait displays hyperbolic trajectories in the $x(1)$-$x(2)$ plane. The horizontal axis is $x(1)$ and the vertical axis is $x(2)$, both ranging from -1000 to 1000. Trajectories in the first and third quadrants approach the $x(2)=0$ axis, while trajectories in the second and fourth quadrants approach the $x(1)=0$ axis. All trajectories converge to the origin (0,0).</p>
備註	

saddle	$A = \begin{bmatrix} -2 & 0 \\ 0 & 3 \end{bmatrix}$
Time Series	<p>time series $A = \begin{bmatrix} -2 & 0 \\ 0 & 3 \end{bmatrix}$ $x_0 = [1000 \ 1000]$</p>  <p>The top plot shows the time series of x_1 (red line) starting at 1000 and decaying exponentially towards 0. The bottom plot shows the time series of x_2 (blue line) starting at 10^{15} and growing exponentially towards 10^{16}.</p>
Phase Portrait	 <p>The phase portrait shows trajectories in the $x(1)$-$x(2)$ plane. The horizontal axis is $x(1)$ and the vertical axis is $x(2)$. Trajectories are shown in four quadrants, all converging towards the origin (0,0).</p>
備註	<p>畫 Phase Portrait 的時候只疊代 100 次，否則衰減的那個變數相對於遞增的那個幾乎看不出變化。</p>

center	$A = \begin{bmatrix} 0 & 2 \\ -2 & 0 \end{bmatrix}$
Time Series	<p>time series $A = \begin{bmatrix} 0 & 2 \\ -2 & 0 \end{bmatrix} \quad x_0 = \begin{bmatrix} 1000 & 1000 \end{bmatrix}$</p> 
Phase Portrait	
備註	

focus	$A=\begin{bmatrix}-2 & 1 \\ -1 & -2\end{bmatrix}$
Time Series	<p>time series $A=\begin{bmatrix}-2 & 1 \\ -1 & -2\end{bmatrix}$ $x_0=[1000 \ 1000]$</p>  <p>The plot shows two time series, x1 (red line) and x2 (blue line), starting at 1000 at time 0. Both series decay towards zero. x1 decays more rapidly than x2. x2 shows a slight undershoot, reaching a minimum of approximately -100 around time 150, before settling back to zero. The x-axis represents time from 0 to 1000, and the y-axis represents the state variables from -200 to 1000.</p>
Phase Portrait	 <p>The phase portrait shows the trajectories of the system in the x(1)-x(2) plane. The trajectories spiral outwards from the origin (0,0), indicating an unstable equilibrium point. The trajectories are colored in a gradient from purple to yellow. The x(1) and x(2) axes range from -1000 to 1000.</p>
備註	

3.程式碼

(1)函式 使用 Euler Method 近似：

```
function xNew=Euler(x,delta,A)
xDot=[0 0]';
xNew=xDot;
xDot(1)=A(1,1)*x(1)+A(1,2)*x(2);
xDot(2)=A(2,1)*x(1)+A(2,2)*x(2);
% fprintf('xDot=%f %f\n', xDot(1),xDot(1));
xNew(1)=x(1)+xDot(1)*delta;
xNew(2)=x(2)+xDot(2)*delta;
% fprintf('xNew=%f %f\n', xNew(1),xNew(1));
return;
```

(2)函式 使用Runge Kutta Method近似：

```
function xNew=RungeKutta(x_0,delta,A)
k1=[0 0]';k2=k1;k3=k1;k4=k1;tmp=k1;xNew=k1;

k1(1)=A(1,1)*x_0(1)+A(1,2)*x_0(2);
k1(2)=A(2,1)*x_0(1)+A(2,2)*x_0(2);
tmp(1)=x_0(1)+k1(1)*(delta/2);
tmp(2)=x_0(2)+k1(2)*(delta/2);
k2(1)=A(1,1)*tmp(1)+A(1,2)*tmp(2);
k2(2)=A(2,1)*tmp(1)+A(2,2)*tmp(2);
tmp(1)=x_0(1)+k2(1)*(delta/2);
tmp(2)=x_0(2)+k2(2)*(delta/2);
k3(1)=A(1,1)*tmp(1)+A(1,2)*tmp(2);
k3(2)=A(2,1)*tmp(1)+A(2,2)*tmp(2);
tmp(1)=x_0(1)+k3(1)*(delta);
tmp(2)=x_0(2)+k3(2)*(delta);
k4(1)=A(1,1)*tmp(1)+A(1,2)*tmp(2);
k4(2)=A(2,1)*tmp(1)+A(2,2)*tmp(2);

xNew(1)=x_0(1)+delta*(k1(1)+2*k2(1)+2*k3(1)+k4(1))/6;
xNew(2)=x_0(2)+delta*(k1(2)+2*k2(2)+2*k3(2)+k4(2))/6;
return;
```

(3)畫Phase Portrait

```
clear;clc;

% A=[-2 0;0 -2];%star
% A=[-2 0;0 -3];%node
A=[-2 0;0 3];%saddle
% A=[0 2;-2 0];%center
% A=[-2 1;-1 -2]%focus

num=8;theta=0; %total of the different kind of initial condition
datasize=100;
for j=1:8
    theta=j*(2*pi/num);
    x1array(1)=real(datasize*exp(1i*theta));
    x2array(1)=imag(datasize*exp(1i*theta));

    for i=1:(datasize-1)
        x(1)=x1array(i); x(2)=x2array(i);
        xNext=RungeKutta(x,0.01,A);
        x1array(i+1)=xNext(1);
        x2array(i+1)=xNext(2);
    end

    xlabel('x(1)');
    ylabel('x(2)');
    plot(x1array,x2array);
    hold on;
end
```


(4)畫Time Series

```
clear;clc;

% A=[-2 0;0 -2];%star
% A=[-2 0;0 -3];%node
% A=[-2 0;0 3];%saddle
% A=[0 2;-2 0];%center
A=[-2 1;-1 -2]%focus

datasize=1000;
x1array(1)=1000;
x2array(1)=1000;
for i=1:(datasize-1)
    x(1)=x1array(i); x(2)=x2array(i);
    xNext=RungeKutta(x,0.01,A);
    x1array(i+1)=xNext(1);
    x2array(i+1)=xNext(2);
end

plot(x1array,'r');
hold on;
plot(x2array,'b');
legend('x1','x2');
title({'time series';'A=[-2 1;-1 -2] x_0=[1000 1000]'});
```