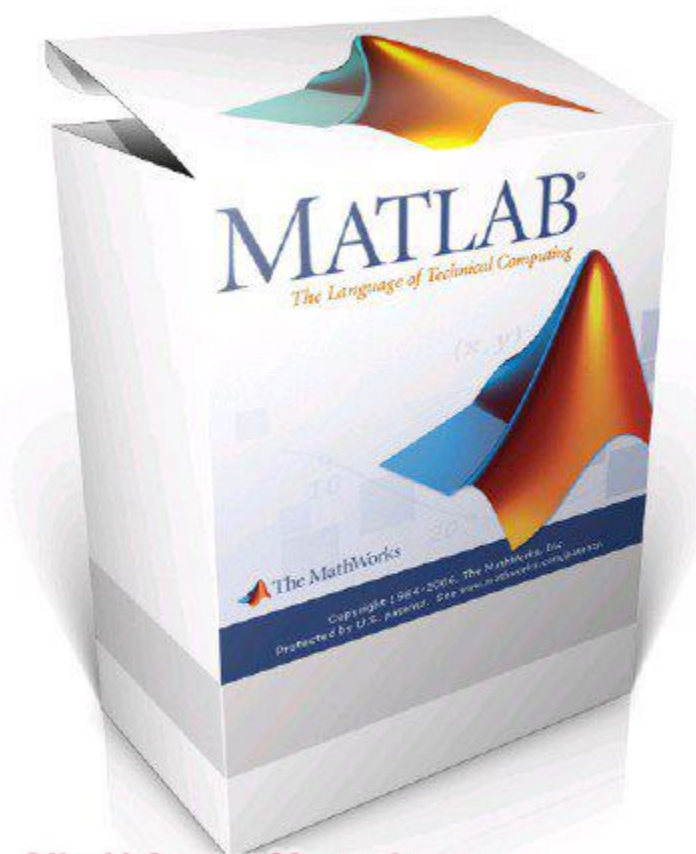


Bài tập MATLAB



<http://chunguv.blogspot.com>

**Bài 1 : chương trình bảng cửu chương**

```
% lap bang cuu chuong tu 2 den 9
clear
con = 'y' ;
while con == 'y'
    fact = input('Enter number i:10 ing character:', 's');
    switch fact
        case '1'
            for i = 1:10
                fact = 1;
                fact = fact*i;
                disp(fact);
            end
        case '2'
            for i = 1:10
                fact = 2;
                fact = fact*i;
                disp(fact);
            end
        case '3'
            for i = 1:10
                fact = 3;
                fact = fact*i;
                disp(fact);
            end
        case '4'
            for i = 1:10
                fact = 4;
                fact = fact*i;
                disp(fact);
            end
        case '5'
            for i = 1:10
                fact = 5;
                fact = fact*i;
                disp(fact);
            end
    end
end
```





```
case'6'
    for i = 1:10
        fact = 6;
        fact = fact*i;
        disp(fact);
    end
case'7'
    for i = 1:10
        fact = 7;
        fact = fact*i;
        disp(fact);
    end
case'8'
    for i = 1:10
        fact = 8;
        fact = fact*i;
        disp(fact);
    end
case'9'
    for i = 1:10
        fact = 9;
        fact = fact*i;
        disp(fact);
    end
otherwise
    disp('wrong number');
end
con = input('continue y or n:','s');
end
% end of program
```

Bài 2 : chương trình dùng lệnh For để tính giai thừa:

```
% dung lenh for de tinh giai thua
n = input('Enter positive integer:')
fact = 1
for i = 1:n;
```





```
fact = fact*i
end
% end of program
```

Bài 3 : dung lenh for tinh tong day so $s=1+3+5+...+n$

```
% dung lenh for tinh tong day so  $s=1+3+5+...+n$ 
n = input('positive integer:')
s = 0
for i = 1:2:n
    s = s + i
end
% end of program
```

Bài 4: dung lenh for tinh tong day so $s=1+3+5+...+n$

```
% dung lenh for tinh tong day so  $s=1+3+5+...+n$ 
n = input('positive integer:')
s = 0
for i = n : -2 : 1
    s = s + i
end
% end of program
```

Bài 5: các phép toán không tiết tục

```
%Beginning of program
%lenh switch tinh +,-,*,/
a = input('Enter a:');
b = input('Enter b:');
```





```
operator = input('Enter operator =, -, *, / : ', 's');
switch operator
    case '+'
        c = a+b;
        disp(c);
    case '-'
        c = a-b;
        disp(c);
    case '*'
        c = a*b;
        disp(c);
    case '/'
        c = a/b;
        disp(c);
    otherwise
        disp('wrong operator');
end
% end of program
```

Bài 6 : chương trình các phép toán có tiếp tục

```
%Beginning of program
%lenh switch continue while tính +, -, *, /
con = 'y'
while con == 'y'
    a = input( 'Enter a : ');
    b = input( 'Enter b : ');
    operator = input('Enter operator +, -, *, / : ', 's ');
    switch operator
        case '+'
            c = a+b;
            disp(c);
        case '-'
            c = a-b;
            disp(c);
        case '*'
            c = a*b;
```





```
    disp(c);
case '/'
    c = a/b;
    disp(c);
otherwise
    disp('wrong operator')
end
con = input('continue y or n:','s');
end
% end of program
```

Bài 7 : tính tổng các số dương:

```
% chương trình nhập dãy số dương và tính tổng của dãy số dương
% beginning of program
con = 'y';
sum = 0;
while con == 'y'
    nun = input('Enter number:');
    if nun < 0 continue;
    else
        sum = sum + nun;
        con = input('continue y/n:','s');
    end
end
disp(sum)
% end of program
```

Bài 8 : Cách sử dụng lệnh Breack :

```
% sử dụng lệnh break
sum = 0
while sum >= 0
    nun = input('Enter number:');
    if nun < 0 break;
    else
```



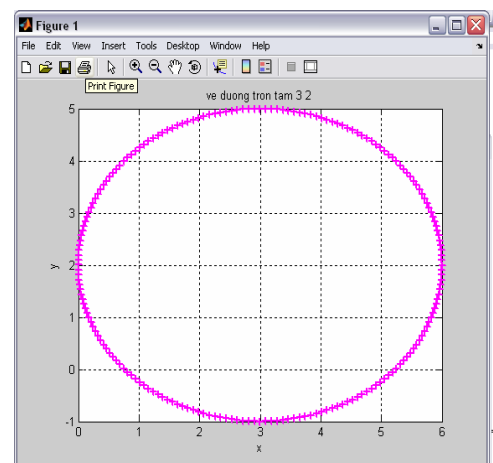
```
sum = sum + nun;  
end  
disp(sum)  
end  
% end of program
```

Bài 9 : tính tổng các số dương . nhập các số dương , nếu ta nhập 1 số âm bất kỳ thì chương trình kết thúc và cho kết quả là tổng của các số dương .

```
% su dung lenh break  
sum = 0  
while sum >=0  
    nun = input('Enter number:');  
    if nun <0 break;  
    else  
        sum = sum + nun;  
    end  
end  
disp(sum)  
% end of program
```

Bài 10 : chương trình vẽ đường tròn có tâm 3,2 trong m-file

```
% ve duong tron tam 3,2  
t = 0:pi/100:2*pi;  
x = 3 + 3*cos(t);  
y = 2 + 3*sin(t);  
plot(x,y,'m','linewidth',1.5,'linestyle','+'),grid  
xlabel('x')  
ylabel('y')  
title('ve duong tron tam 3 2')  
% End of program .
```

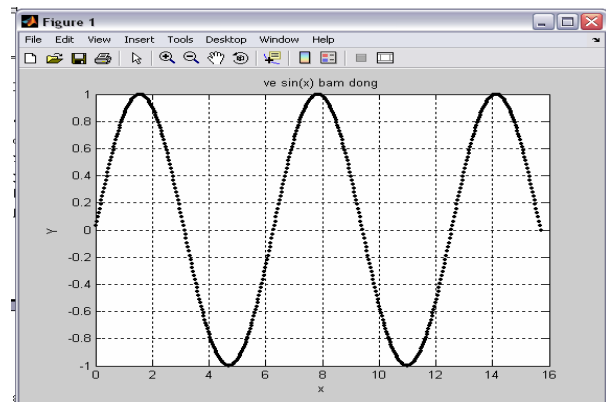


Bài 11 : Vẽ đồ thị hình $\sin(x)$ bấm động :

```

% vẽ đồ thị của hàm sin
clear
x=0;
y=sin(x);
h=0.01;
p=plot(x,y,'k','linestyle','.', 'EraseMode','none','MarkerSize',5);
axis([0 16 -1 1]),grid
for x=0:pi/100:5*pi
    y=sin(x);
    set(p,'XData',x,'YData',y);
    drawnow
    pause(0.01)
    xlabel('x'),ylabel('Y')
    title('ve sin(x) bam dong')
end
% En of program.

```

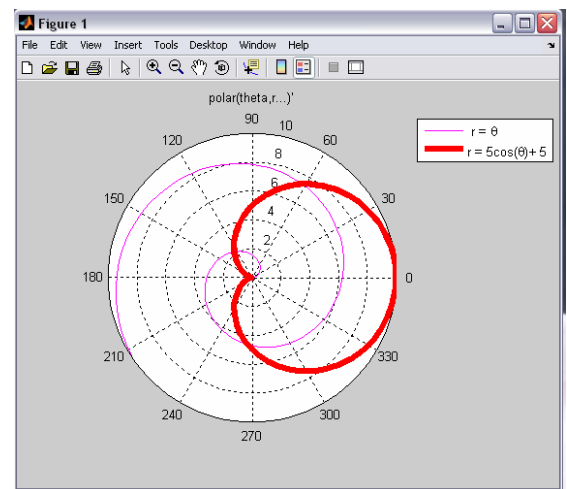
**Bài 12 : Tọa độ cực****Cách 1:**

% tọa độ cực sử dụng hàm polar:

```

% beginning of program
theta=[0.0:0.1:10.0];
r1=theta;
r2=5*cos(theta)+5;
h(1)=polar(theta,r1,'m');
set(h(1),'markersize',2*6);
hold on
h(2)=polar(theta,r2,'r');
set(h(2),'Linewidth',5);
hold off
title('polar(theta,r...)');

```




```

legend(h,'r = \theta','r = 5cos(\theta)+ 5',-1);
% End of program

```

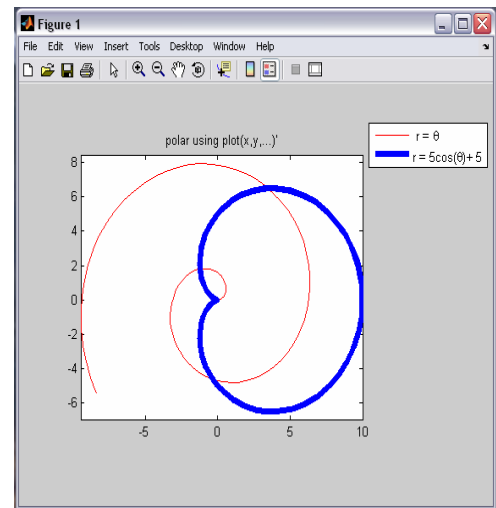
Cách 2 : toa do cuc su dung ham plot:

```

% Beginning of program:
theta=[0:0.1:10.0];
r1=theta;
r2=5*cos(theta)+5;
[x1,y1] = pol2cart(theta,r1);
[x2,y2] = pol2cart(theta,r2);
plot(x1,y1,'r','Markersize',2*6);

hold on
plot(x2,y2,'b','linewidth',5);
axis('equal')
title('polar using plot(x,y,...)');
legend('r = \theta','r = 5cos(\theta)+ 5',-1);
% End of program.

```



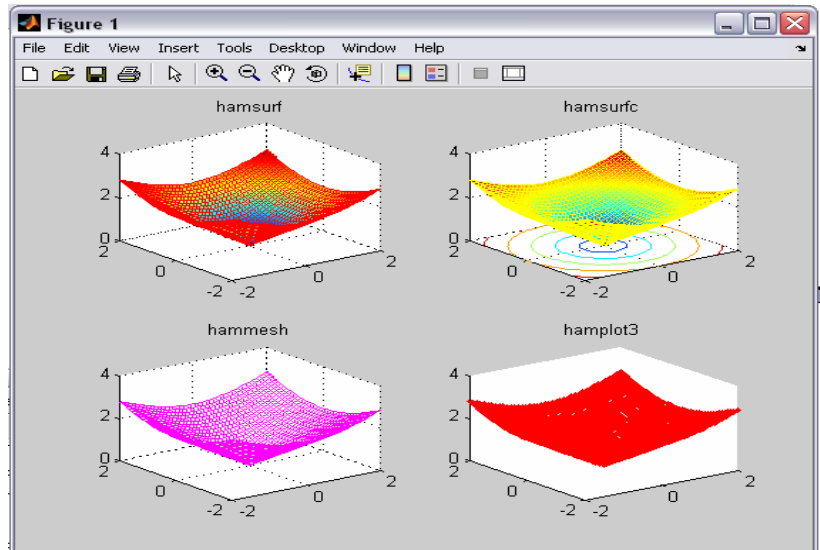
Bài 13 : Vẽ không gian 3 D

```

% ve khong gian 3 D
[x,y]=meshgrid(-2:0.1:2);
z = sqrt(x.^2+y.^2);
subplot(2,2,1);
surf(x,y,z,'Edgecolor','r'),title('hamsurf');
% ham ve do thi ba chieu z theo x va y .
subplot(2,2,2);
surfc(x,y,z,'Edgecolor','y'),title('hamsurfc');
% ham ve do thi ba chieu z theo x va y co hinh chieu tren mat phang x - y .
subplot(2,2,3);
mesh(x,y,z,'Edgecolor','m'),title('hammesh');
% ham ve do thi ba chieu z theo x va y nhung khong co to mau nen.

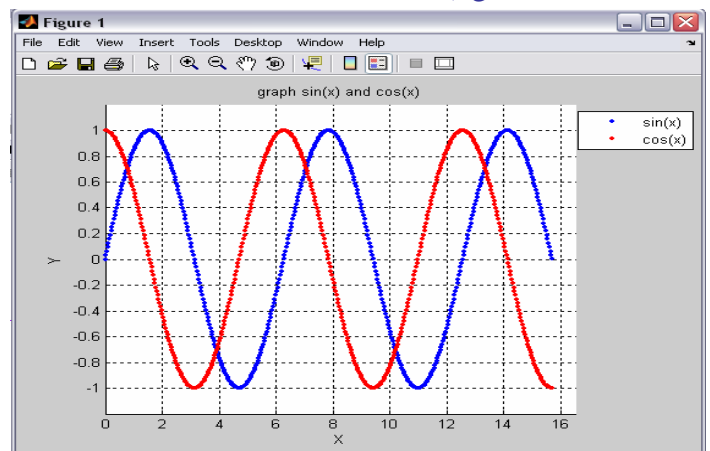
```

```
subplot(2,2,4);
plot3(x,y,z,'linewidth',2.5,'linestyle','.', 'color','r'),title('hamplot3');
% ham vẽ đồ thị ba chiều
z theo x và y có thể thay
đổi được các thuộc tính.
% End of program.
```



Bài 14 : Vẽ đồ thị sin và cos trên cùng tọa độ

```
%begin
x=0:pi/100:5*pi;
y1=sin(x);
y2=cos(x);
hold on
p=plot(x,y1,'b',x,y2,'r','linestyle','.', 'erasemode','none','markersize',10),grid;
xlabel('X'),ylabel('Y')
legend('sin(x)','cos(x)',-1)
title('graph sin(x) and cos(x)')
axis([0 16.5 -1.2 1.2])
%drawnow
%pause(0,0.1)
% end of program ..
```



Bài 15 : Vẽ đồ thị chạy hàm $\cos(x)$ bấm động :

% vẽ đồ thị của hàm $\cos(x)$:

```
clear
```

```
x = 0;
```

```
y = cos(x);
```

```
h = 0.01;
```

```
p = plot(x,y,'m','linestyle','.', 'EraseMode','none','MarkerSize',5);
```

```
axis([0 16 -1.2 1.2]),grid
```

```
hold on
```

```
for x = 0:pi/100:5*pi
```

```
    y = cos(x);
```

```
    set(p,'Xdata',x,'Ydata',y)
```

```
    xlabel('x'),ylabel('Y')
```

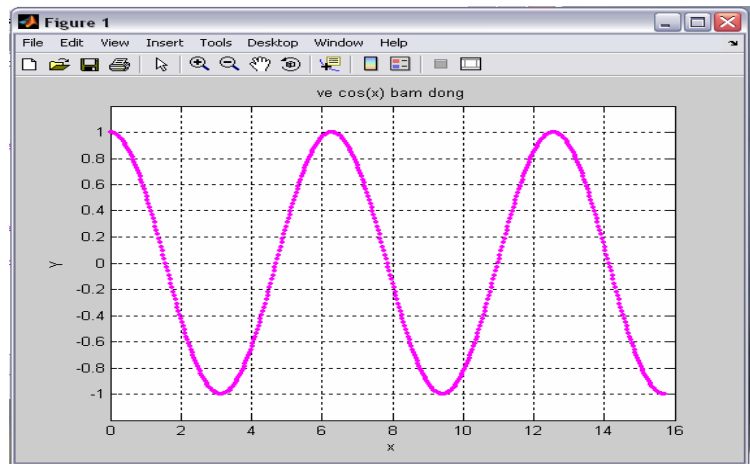
```
    title('ve cos(x) bam dong')
```

```
    drawnow
```

```
    pause(0.01)
```

```
end
```

```
% End of program.
```

**Bài 16 : Vẽ đường tròn elip**

% vẽ đường tròn nhập giá trị bán trục

```
a = input('nhập a:');
```

```
b = input('nhập b:');
```

```
r = input('nhập bán kính r:');
```

```
r1 = input('nhập trục nhỏ r1:');
```

```
r2 = input('nhập trục lớn r2:');
```

```
t = 0:pi/100:2*pi;
```

```
x1 = a+r*cos(t);
```

```
y1 = b+r*sin(t);
```

```
x2 = a + r1*cos(t);
```

```
y2 = b + r2*sin(t);
```

```
set(gcf,'defaulttextcolor','r');
```

```
h = plot(x1,y1,'r',x2,y2,'b');
```

```
set(h,'linewidth',2.25);
```



```

axis('square');
xlabel('X');
h = get(gca,'xlabel');
set(gca,'fontsize',10);
set(gca,'xtick',-4:10);
ylabel('Y');
h = get(gca,'ylabel');
set(gca,'fontsize',16);
set(gca,'xtick',-12:2:12);
title('graphs of  $(x-a)^2 + (y-b)^2 = r^2$  and  $(x-a)^2/(r1)^2 + (y-b)^2/(r2)^2 = 1$ ');
h = get(gca,'title');
set(h,'fontsize',12);
grid
% End of pro gram

```

Bài 17 : Vẽ sin và cos có giới hạn

```

% ve sin cos
h=figure('menubar','none','numbertitle','off','name','LuanVanTotNghiep','position',
,[100 100 600 600],'color','c')
axes('parent',h,'position',[0.1 0.1 0.8 0.8])
x=0:pi/100:2*pi
plot(x,sin(x),'b',x,cos(x),'r'),grid
uicontrol('parent',h,'Style','pushbutton','string','START','position',[5 5 50
50],'callback','move4')
xlabel('X')
ylabel('Y')
h0=uimenu('parent',h,'label','FILE')
uimenu('parent',h0,'label','RUN','callback','move4')
uimenu('parent',h0,'label','CLOSE','callback','CLOSE')
h1=uimenu('parent',h,'label','EDIT')
uimenu('parent',h1,'label','CLEAR','callback','CLC')
uimenu('parent',h1,'label','CLOSE','callback','CLOSE')
% end of program.

```



**Bài 18 : Vẽ đồ thị dạng sóng theo từng phần**

```
% begining of program:
k = 0;
for n = 1:3:10
    n10 = n*10;
    x = linspace(-2,2,n10);
    y = x./(1 + x.^2);
    k = k + 1;
    subplot(2,2,k)
    plot(x,y,'g','LineWidth',2.0,'LineStyle','.')
    xlabel('X'),ylabel('Y')
    title('y = x/(1+x^2)')
    axis([-2 2 -0.8 0.8])
    grid
    pause(3);
end
% End of program.
```

Bai 19 : Chương trình cấu trúc bảng điểm :

```
%chuong trinh cau truc bang diem
clear
con = 'y';
i = 1;
while con == 'y'
    n = i;
    lop(i).STT = input('enter STT :','s');
    lop(i).name = input('enter name :','s');
    lop(i).diem = input('enter diem :');
    if lop(i).diem > 8 lop(i).hang = 'Gioi';
    elseif lop(i).diem <=8 && lop(i).diem >7 lop(i).hang = 'Kha';
    elseif lop(i).diem <=7 && lop(i).diem >5 lop(i).hang = 'Trung Binh';
    else lop(i).hang = 'Yeu';
    end
    i = i + 1;
    con = input('continue? (y/n) :');
end
```





```
i = i + 1;  
con = input('continue thanh vien khac y/n :','s');  
end  
for i = 1:n;  
    lop(i)  
end  
% end of program
```

Ví Dụ Như Sau :

```
enter STT :1  
enter name :Hung  
enter diem :9  
continue thanh vien khac y/n :y  
enter STT :2  
enter name :Binh  
enter diem :8  
continue thanh vien khac y/n :y  
enter STT :3  
enter name :Dat  
enter diem :8  
continue thanh vien khac y/n :n
```

```
ans =  
    STT: '1'  
   name: 'Hung'  
   diem: 9  
   hang: 'Gioi'
```

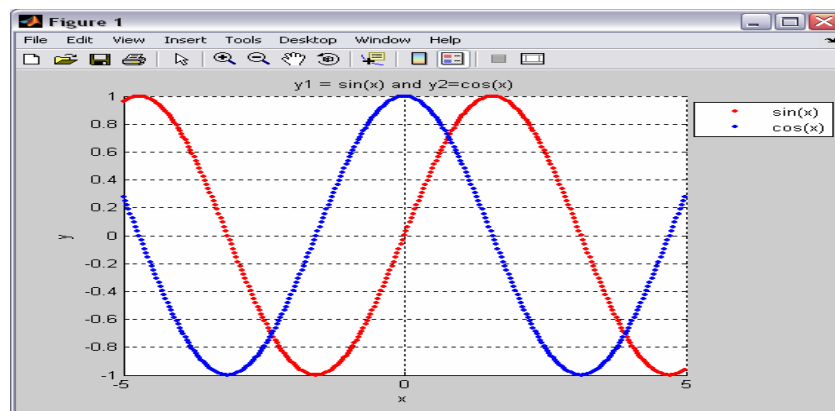
```
ans =  
    STT: '2'  
   name: 'Binh'  
   diem: 8  
   hang: 'Kha'
```

```
ans =  
    STT: '3'  
   name: 'Dat'  
   diem: 8  
   hang: 'Kha'
```



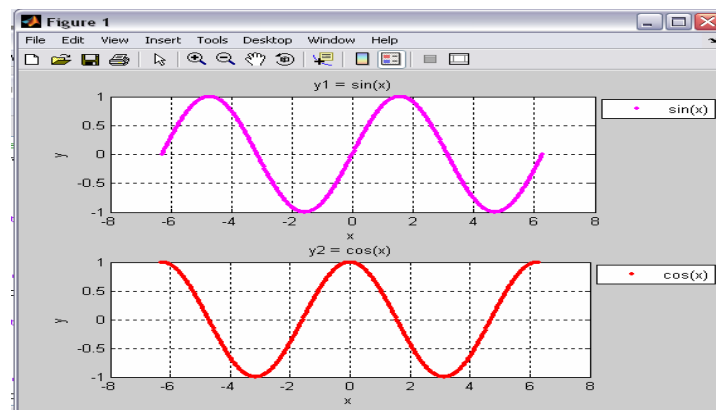
Bài 20 : vẽ đồ thị $\sin(x)$ và $\cos(x)$ trên cùng 1 hệ trục tọa độ

```
%Beginning of program
x = -2*pi:pi/100:5*pi;
y1 = sin(x);
y2 = cos(x);
hold on
plot(x,y1,'linewidth',1.0,'linestyle','.', 'color','r')
plot(x,y2,'linewidth',1.0,'linestyle','.', 'color','b')
grid
xlabel('x')
ylabel('y')
title('y1 = sin(x) and y2=cos(x)')
legend ('sin(x)','cos(x)',-1)
axis([-5 5 -1 1])% cho phép tạo giới hạn của hệ trục xoy
axis on % cho phép bỏ hệ trục tọa độ xoy.
% lệnh grid : tạo mạng lưới , lệnh xlabel(x) : gian nhãn trục x, lệnh
% title( y = sin(x)): gian nhãn cho đồ thị.
% lệnh hold on : dùng để giữ 2 đồ thị trên cùng 1 hệ trục tọa độ.
% End of program.
```



Bài 21 : Vẽ $\sin(x)$ và $\cos(x)$ trên 2 hệ tọa độ xy nhưng cùng trong 1 khung

```
%Beginning of program
x = -2*pi:pi/100:2*pi;
y1 = sin(x);
y2 = cos(x);
subplot(2,1,1) % lệnh cho phết tạo ra những khung chứa
plot(x,y1,'linewidth',1.0,'linestyle','o','color','m')
grid
xlabel('x')
ylabel('y')
title('y1 = sin(x)')
legend('sin(x)',-1) % lệnh cho phép hiện thị tên và màu của hàm vẽ.
subplot(2,1,2) % lệnh cho phết tạo ra những khung chứa
plot(x,y2,'linewidth',1.0,'linestyle','.', 'color','r')
grid
xlabel('x')
ylabel('y')
title('y2 = cos(x)')
legend('cos(x)',-1) % lệnh cho phép hiện thị tên và màu của hàm vẽ.
% End of program
% lenh grid : tao mang luoi , lenh xlabel(x) : gian nhan truc x, lenh
% title( y = sin(x)): gang nhan cho do thi.
```



**Bài 22 : Vẽ $\sin(x)$, $\cos(x)$, $\text{sinc}(x)$ và $1 - \sin(x)$ trên 4 hệ tọa độ xy nhưng cùng trong 1 khung :**

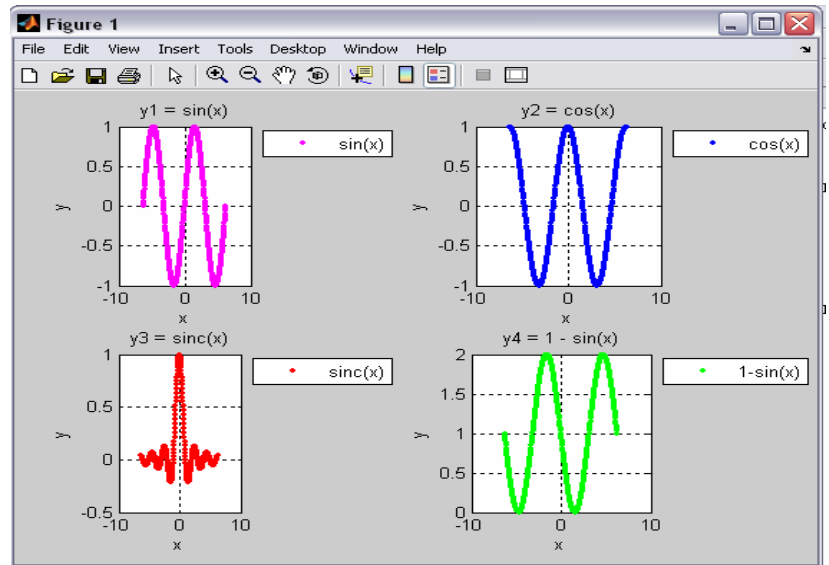
```
%Beginning of program
x = -2*pi:pi/100:2*pi;
y1 = sin(x);
y2 = cos(x);
y3 = sinc(x);
y4 = 1 - sin(x);
subplot(2,2,1)
plot(x,y1,'linewidth',1.0,'linestyle','.', 'color','m')
grid
xlabel('x')
ylabel('y')
title('y1 = sin(x)')
legend ('sin(x)',-1)
subplot(2,2,2)
plot(x,y2,'linewidth',1.0,'linestyle','.', 'color','b')
grid
xlabel('x')
ylabel('y')
title('y2 = cos(x)')
legend ('cos(x)',-1)
subplot(2,2,3)
plot(x,y3,'linewidth',1.0,'linestyle','.', 'color','r')
grid
xlabel('x')
ylabel('y')
title('y3 = sinc(x)')
legend ('sinc(x)',-1)
subplot(2,2,4)
plot(x,y4,'linewidth',1.0,'linestyle','.', 'color','g')
grid
```



```

xlabel('x')
ylabel('y')
title('y4 = 1 - sin(x)')
legend('1-sin(x)',-1)
% End of program

```



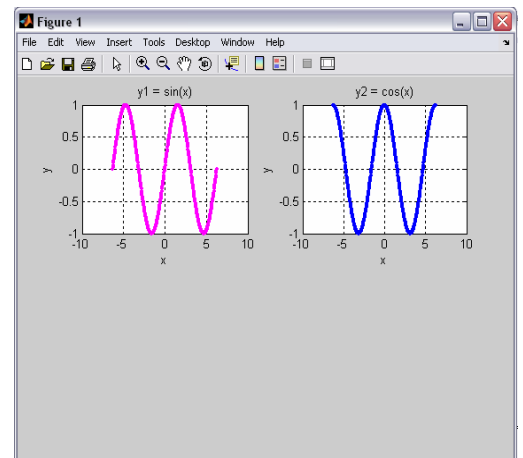
Bài 23: Vẽ $\sin(x)$, $\cos(x)$ trên 2 hệ tọa độ xoy nhưng cùng trong 1 khung, $1 - \sin(x)$ và

$1 - \cos(x)$ trên 2 hệ tọa độ xoy nhưng cùng trong 1 khung

```

%Beginning of program
x = -2*pi:pi/100:2*pi;
y1 = sin(x);
y2 = cos(x);
y3 = 1 - sin(x);
y4 = 1 - cos(x);
figure
subplot(2,2,1)
plot(x,y1,'linewidth',1.0,'linestyle','-', 'color','m')
grid
xlabel('x')
ylabel('y')
title('y1 = sin(x)')
subplot(2,2,2)
plot(x,y2,'linewidth',1.0,'linestyle','-', 'color','b')
grid
xlabel('x')
ylabel('y')

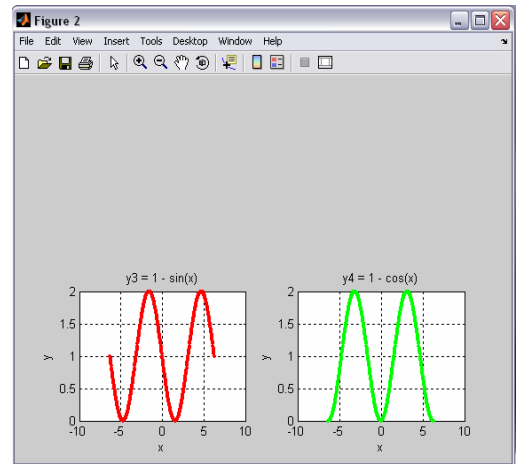
```



```

title('y2 = cos(x)')
figure
subplot(2,2,3)
plot(x,y3,'linewidth',1.0,'linestyle','.', 'color','r')
grid
xlabel('x')
ylabel('y')
title('y3 = 1 - sin(x)')
subplot(2,2,4)
plot(x,y4,'linewidth',1.0,'linestyle','.', 'color','g')
grid
xlabel('x')
ylabel('y')
title('y4 = 1 - cos(x)')
% End of program

```



Bài 24 : dùng lệnh while để tính tổng chuỗi : $s = x^n/n!$

Cách 1:

%dùng lenh while and for tinh tong $s = x^n/n!$

```
n = input('Enter n :');
```

```
x = input('Enter x :');
```

```
i = 1;
```

```
s = 0;
```

```
while i <= n; % ( kiểm tra I có <= n hay ko? Nếu <= thì bắt đầu thực hiện vòng lặp mới).
```

```
    fact = 1;
```

```
    for k = 1:i;
```

```
        fact = fact*k;
```

```
    end
```

```
    s = s + (x^i)/fact;
```

```
    i = i + 1;
```

```
end
```

```
disp(s)
```

```
% end of program
```

**Cách 2: chỉ dùng lệnh while để tính :**

```
%dung lenh while and for tinh tong s = x^n/n!
n = input('Enter n :');
x = input('Enter x :');
i = 1;
k = 1;
s = 0;
fact = 1;
while k <= n;
    fact = fact*k;
    s = s + (x^i)/fact;
    k = k + 1;
    i = i + 1;
end
disp(s)
% end of program
```

Bài 25 : cách dùng lệnh if elseif else :

```
% cach dung lech if elseif else
con = 'y'
while con == 'y'
n = input('Enter any number:');
if n > 0
    disp('positive') ;
elseif n < 0
    disp('Negative') ;
else
    disp('Zero') ;
end
con =input('continue y/n:','s');
end
% End of program
```



Bài 26 : Vẽ đồ thị hình ellips :

%Beginning of program _ve do thi hinh ellips:

a = input('Enter a :'); % nhap a = 5

b = input('Enter b :'); % nhap b = 4

c = input('Enter c :'); % nhap c = 4

d = input('Enter d :'); % nhap d = 8

t = 0:pi/100:2*pi;

x = a + c*cos(t);

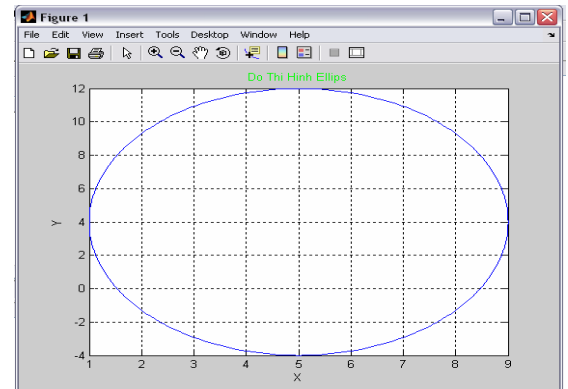
y = b + d*sin(t);

set(gcf,'DefaultTextColor','g'); % mặc định văn bản hiển thị trên đồ thị là màu xanh lá cây. .

plot(x,y,'b'),xlabel('X'),ylabel('Y'),grid;

title('Do Thi Hinh Ellips'); % tạo tên cho đồ thị

%End of program.

**Bài 27: : Vẽ đồ thị hình ellips có thể thay đổi nét vẽ**

%Beginning of program _ve do thi hinh ellips:

a = input('Enter a :'); % nhap a = 5

b = input('Enter b :'); % nhap b = 4

c = input('Enter c :'); % nhap c = 4

d = input('Enter d :'); % nhap d = 8

t = 0:pi/100:2*pi;

x = a + c*cos(t);

y = b + d*sin(t);

set(gcf,' DefaultTextColor ', 'b');

h = plot(x,y,'b'),xlabel('X'),ylabel('Y'),grid;

set(h,' linewidth ',3);% thay doi do rong cua net ve .

h = get(gca,'xlabel');% chon doi tuong la nhan cua trục x .

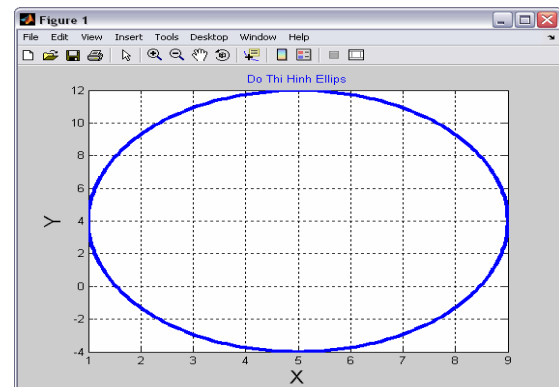
set(h,' FontSize ',18);% tang co chu cua nhan trục x.

h = get(gca,'ylabel');

set(h,' FontSize ',18);

title(' Do Thi Hinh Ellips ');

%End of program



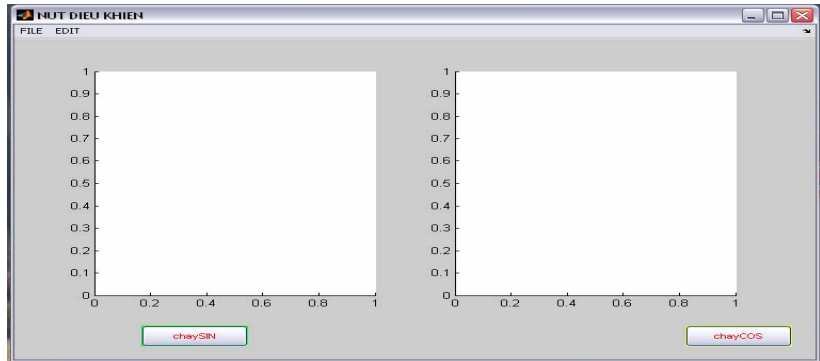


Bài 28: các lệnh viết chương trình giao diện nút điều khiển:

```
%chuong trinh chay cos va sin theo nut dieu khien
h=figure('position',[100 100 850
550],'menubar','none','numbertitle','off','name','NUT DIEU KHIEN')
h0=uimenu('parent',h,'label','FILE')
% lenh : uimenu : tao he thong thuc don trong cua so gio dien.
%lenh: 'lable','FILE' : cho phép tạo tên nhân là FILE trên thanh menubar.
uimenu('parent',h0,'label','RUN','callback','chaySIN')
% tao con của FILE có tên danh nhân là RUN
uimenu('parent',h0,'label','CLOSE','callback','close')
% tao con của FILE có tên danh nhân là CLOSE
uicontrol('parent',h,'style','pushbutton','position',[120 20 100
30],'string','chaySIN',....
'backgroundcolor','g','foregroundcolor','r','visible','on','callback','chaySIN')
% Tạo nút điều khiển chạy chương trình là đồ thị hình SIN được lưu trong m-file.
% lenh:'parent',h, : là thuộc tính cha do là của số figure.
% lenh: 'style','pushbutton' : thuộc tính tạo nút điều khiển push.
% lenh: 'position' : là thuộc tính vị trí định vị của nút điều khiển.
% lenh: 'BackgroundColor' : là thuộc tính tạo màu nền cho các nút điều khiển.
% lenh: 'ForegroundColor' : là thuộc tính tạo màu văn bản cho các nút dk.
% lenh : 'callback': là thuộc tính gọi các chương trình ứng dụng với giá
% trị là tên của chương trình được lưu dưới dạng m-file.
% lenh: 'Visible': là thuộc tính hiển thị với 2 giá trị 'on' of 'off' các
% nút điều khiển khi chọn lệnh này.
h1=uimenu('parent',h,'label','EDIT')
%lenh cho phép tạo tên nhân là EDIT trên thanh menubar.
uimenu('parent',h1,'label','CLEAR','callback','clc')
% tao con của EDIT có tên danh nhân là 'clc' khi nhân nó sẽ làm sạch của số
Window
uimenu('parent',h1,'label','CLOSE','callback','close')
% tao con của EDIT có tên danh nhân là 'close'
uicontrol('parent',h,'style','pushbutton','position',[630 20 100
30],'string','chayCOS',.....
```

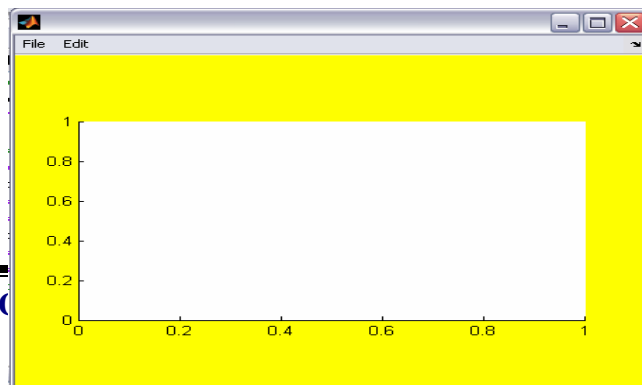


```
'backgroundcolor','y','foregroundcolor','r','callback','chayCos')
% Tao nut dieu khiem chay chuong trinh la do thi hinh COSIN duoc luu trong
% m-file.
axes('parent',h,'position',[0.1 0.2 0.35 0.7])
%cho phép tạo hệ trục xy của đồ thị sin(x) ở vị trí [0.1 0.2 0.35 0.7]
axes('parent',h,'position',[0.55 0.2 0.35 0.7])
%cho phép tạo hệ trục xy của đồ thị cos(x) ở vị trí [0.55 0.2 0.35 0.7]
% End of program.
```



Bài 29: các lệnh của cửa sổ giao diện :

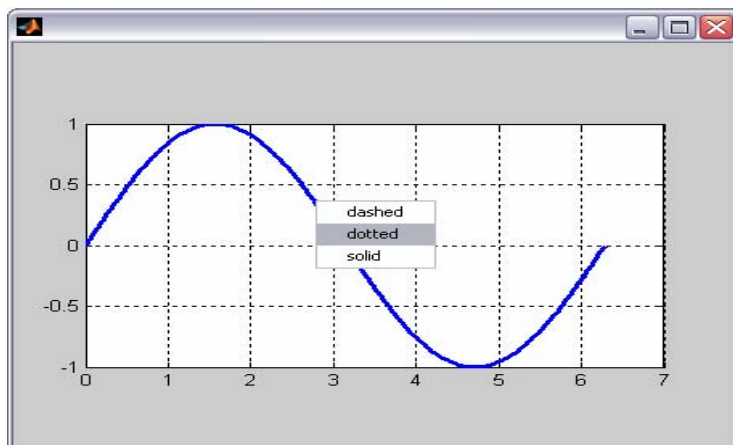
```
%Beginning of program : giao dien
h = figure('Position',[200 200 450 350],'MenuBar','none',...
'NumberTitle','off','color','y');
% lệnh : 'color' , 'y' : tọa độ tính màu nền của cửa sổ giao diện
% là màu vàng
axes('Parent',h,'Position',[0.1 0.2 0.8 0.6]);
h0 = uimenu('Parent',h,'Label','File');
uimenu('Parent',h0,'Label','Run','Callback','chaySIN');
uimenu('Parent',h0,'Label','Close','Callback','close');
h1 = uimenu('Parent',h,'Label','Edit');
uimenu('Parent',h1,'Label','Clear','Callback','clc');
uimenu('Parent',h1,'Label','Close','Callback','close');
%End of program.
```



Bài 30: Tạo giao diện người sử dụng vẽ đồ thị có thể thay đổi nét vẽ :

Tạo đối tượng đồ thị hình sin người sử dụng có thể thay đổi nét vẽ bằng cách click chuột phải trên nét vẽ :

```
% beginning of program
clear
h = figure('Position',[100 100 450 350],'MenuBar','none',...
    'NumberTitle','off');
axes('Parent',h,'Position',[0.1 0.2 0.8 0.6]);
Cmenu = uicontextmenu;
x = 0:pi/100:2*pi;
hline = plot(x,sin(x),'LineWidth',2.6,'UIContextMenu',Cmenu);
grid
cb1 = ['set(hline,"LineStyle","--")'];
cb2 = ['set(hline,"LineStyle",":")'];
cb3 = ['set(hline,"LineStyle","-")'];
item1 = uimenu(Cmenu,'Label','dashed','Callback',cb1);
item2 = uimenu(Cmenu,'Label','dotted','Callback',cb2);
item3 = uimenu(Cmenu,'Label','solid','Callback',cb3);
% End of program.
```



Bài 31: chương trình vẽ đồ thị bám động theo điểm của ellips :

%Beginning of program : ve do thi hinh elips bam dong.

t = 0;

x = 3 + 6*cos(t);

y = -2 + 9*sin(t);

h = 0.01

p = plot(x,y,'r','linestyle','o','erasemode','none','markersize',5),grid

axis([-4 10 -12 8])

hold on

for t = 0:pi/100:5*pi;

 x = 3 + 6*cos(t);

 y = -2 + 9*sin(t);

 set(p,'xdata',x,'ydata',y)

drawnow

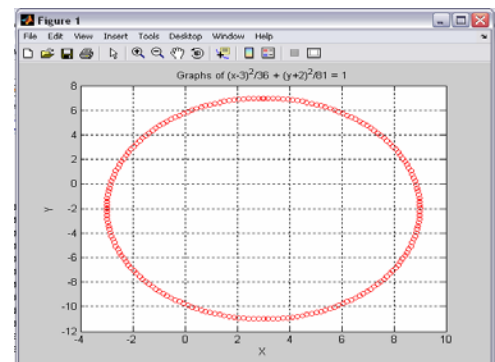
 pause(0.005) % tốc độ chấm động là 0,005 giây.

 xlabel('X'),ylabel('Y')

 title('Graphs of $(x-3)^2/36 + (y+2)^2/81 = 1$ ')

end

%End of program

**Bài 32 chương trình tọa giao diện người sử dụng với nút điều khiển chạy chương trình vẽ đồ thị câu 31**

%Beginning of program : giao dien

h = figure('Position',[200 200 500 500],'MenuBar','none',...

 'NumberTitle','off','color','y');

% lenh : 'color' , 'y' : toa thuoc tinh mau nen cua cua so giao dien

%la mau vang

axes('Parent',h,'Position',[0.1 0.2 0.8 0.6]);

uicontrol('parent',h,'style','pushbutton','position',[120 20 100 30],'string','Chay Elips',....

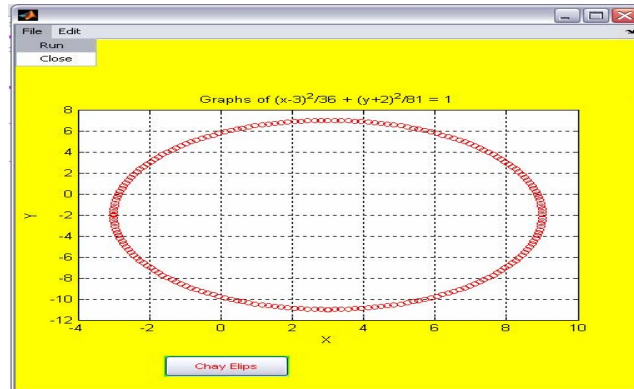
'backgroundcolor','g','foregroundcolor','r','visible','on','callback','elips7')

h0 = uimenu('Parent',h,'Label','File');

```

uimenu('Parent',h0,'Label','Run','Callback','elips7');
uimenu('Parent',h0,'Label','Close','Callback','close');
h1 = uimenu('Parent',h,'Label','Edit');
uimenu('Parent',h1,'Label','Clear','Callback','clc');
uimenu('Parent',h1,'Label','Close','Callback','close');
% End of program.

```

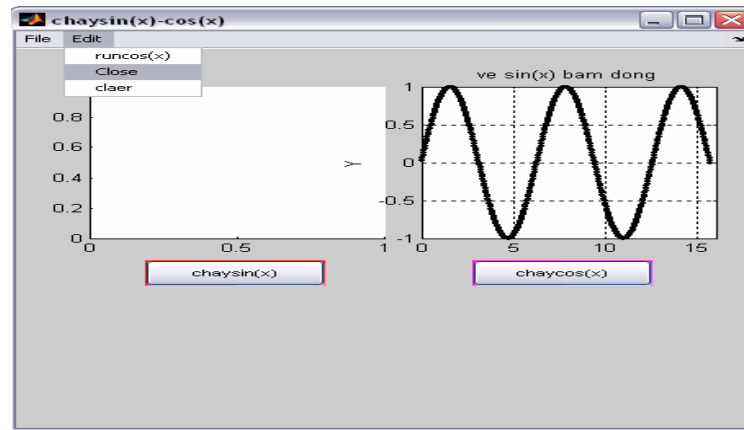


Bài 33: chương trình tạo giao diện người sử dụng với nút điều khiển chạy chương trình vẽ đồ thị $\sin(x)$ và $\cos(x)$:

```

% chương trình tạo nút điều khiển chạy sin(x) và cos(x):
h=figure('position',[100 100 450
450],'menubar','none','numbertitle','off','name','chaysin(x)-cos(x)');
h0=uimenu('parent',h,'label','File');
uimenu('parent',h0,'label','runsine(x)','callback','chaysine');
uimenu('Parent',h0,'Label','Close','Callback','close');
uicontrol('parent',h,'style','pushbutton','string','chaysin(x)','backgroundcolor','r',...
'foregroundcolor','k','position',[80 170 110 30],'callback','chaysine');
axes('parent',h,'position',[0.1 0.5 0.4 0.4]);
h1 = uimenu('parent',h,'label','Edit');
uimenu('parent',h1,'label','runcos(x)','callback','chaycos');
uimenu('Parent',h1,'Label','Close','Callback','close');
uimenu('parent',h1,'label','clear','callback','clc');
uicontrol('parent',h,'style','pushbutton','string','chaycos(x)','backgroundcolor','m',...
'foregroundcolor','k','position',[280 170 110 30],'callback','chaycos');
axes('parent',h,'position',[0.55 0.5 0.4 0.4]);
% End of program .

```



Bài 34: các lệnh của cửa sổ giao diện :

% cac lenh tao cua so giao dien :

% beginning of program

h = figure('position',[100 100 250 250],'menubar','none','numbertitle','off',...
'name','luan van tot nghiep','color','r')

% lenh: 'position',[100 100 250 250]: dinh vi tri cua cua so giao dien tren man
hinh may tinh.

% lenh: 'menubar','none': tao thuc tinh tat thanh thuc don menubar.

% lenh: 'numbertitle','off': tao thuc tinh tat ten nhan number figure tren thanh
menubar.

% lenh: 'h = figure': gan cua so giao dien figure cho bien h.

axes('parent',h,'box','off','linewidth',1.5,'TickDir','out','tickLength',[0.02 0.02],...
'Xgrid','on','Ygrid','on','visible','off','position',[0.1 0.4 0.5 0.4])

% lenh: axes: tao cua so he truc toa do xy trong cua so cha figure.

% lenh: 'Box': mac dinh la mo khung cua he truc xy. Tat la 'off'.

% lenh: 'linewidth',1.5 : tao do rong cua khung he truc xy.

% lenh: 'TickDir','out': toa thuc tinh cac duong ganh so he truc xy nam phia
ngoai.

% lenh: 'tickLength',[0.02 0.02]:tao thuc tinh do dai net gach he truc toa do xy
trong khung cua figure.

% lenh: 'Xgrid','on','Ygrid','on': tao thuc tinh duong ke luoi he truc xy.

% lenh: 'Visible','off' : la thuc tinh lam tat cua so he truc toa do xy .

% lenh: 'posotion',[0.1 0.4 0.5 0.4]: da vi tri he truc xy trong khung giao dien figure.

% End of program.



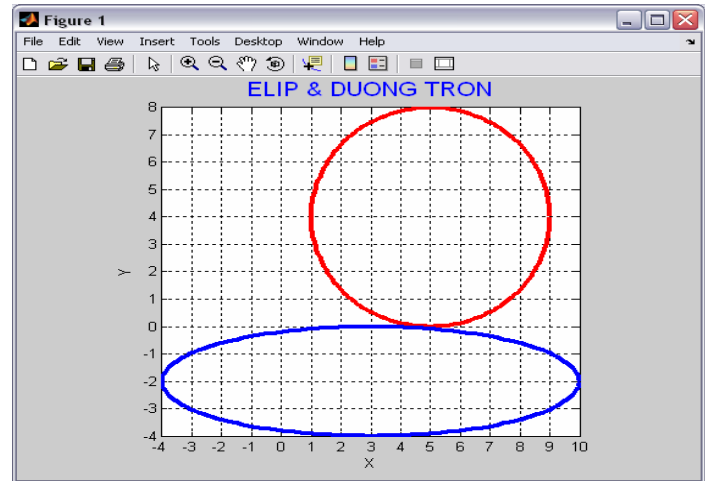
Bài 35 : vẽ hình tròn và hình elips trên cùng 1 hệ trục tọa độ :

```
%begining
a1 = input('nhap a1 : ')    ;% nhap a1 = 5.
b1 = input('nhap b1 : ')    ;% nhap b1 = 4.
r = input('nhap r : ')      ;% nhap r = 4.
a2 = input('nhap a2 : ')    ;% nhap a2 = 3.
b2 = input('nhap b2 : ')    ;% nhap b2 = -2.
r1 = input('nhap r1 : ')    ;% nhap r1 = 7.
r2 = input('nhap r2 : ')    ;% nhap r2 = 2.
t = 0:pi/100:2*pi ;
x1 = a1 + r*cos(t);
y1 = b1 + r*sin(t);
x2 = a2 + r1*cos(t);
y2 = b2 + r2*sin(t);
set(gcf,'defaulttextcolor','b')
% mac dinh van ban la mau xanh dam.
h = plot(x1,y1,'r',x2,y2,'b'),grid ;
set(h,'linewidth',3)
```

```

axis('square')
xlabel('X')
h1 = get(gca,'xlabel');
set(h1,'fontsize',10)
set(gca,'xtick',-8:10)
ylabel('Y')
h2 = get(gca,'ylabel');
set(h2,'fontsize',10)
set(gca,'ytick',-8:10)
title('ELIP & DUONG TRON')
h3 = get(gca,'title');
set(h3,'fontsize',15)
%end of program.

```

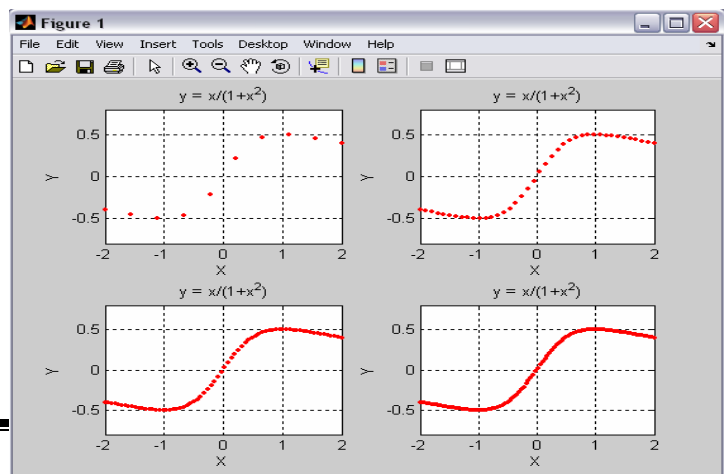


Bài 36 :

```

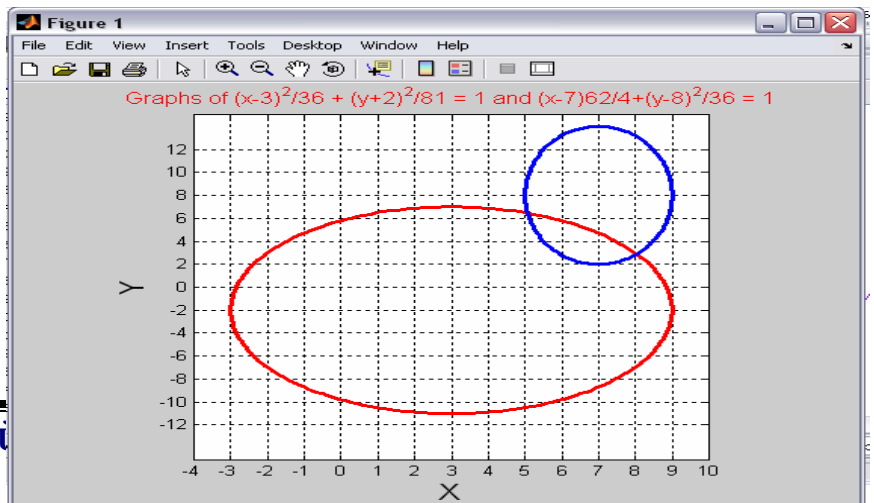
% beginning of program
k = 0;
for n = 1:3:10
    n10 = n*10;
    x = linspace(-2,2,n10);
    y = x./(1 + x.^2);
    k = k + 1;
    subplot(2,2,k)
    plot(x,y,'r','LineWidth',2.0,'LineStyle','.')
    xlabel('X'),ylabel('Y')
    title('y = x/(1+x^2)')
    axis([-2 2 -0.8 0.8]),grid
    pause(3);
end
%End of program

```



Bài 37 : Vẽ 2 hình Elips trên cùng 1 hệ trục tọa độ :

```
%Beginning of program
t = 0:pi/100:2*pi;
x1 = 3 + 6*cos(t);
y1 = -2 + 9*sin(t);
x2 = 7 + 2*cos(t);
y2 = 8 + 6*sin(t);
set(gcf,'DefaultTextColor','red') % tạo màu mặc định
h1 = plot(x1,y1, 'r',x2,y2,'b');
set(h1,'LineWidth',2.25);
axis('square');
xlabel('X')
h = get(gca,'xlabel');
set(h,'FontSize',16);
set(gca,'XTick',-4:10); % tạo 15 điểm trên trục hoành x
ylabel('Y')
h = get(gca,'ylabel');
set(h,'FontSize',16);
set(gca,'YTick',-12:2:12); % tạo 14 điểm trên trục y
title('Graphs of  $(x-3)^2/36 + (y+2)^2/81 = 1$  and  $(x-7)^2/4 + (y-8)^2/36 = 1$ ')
h = get(gca,'title');
set(h,'FontSize',12);
grid
%End of program
```





Bài 38: chương trình tọa giao diện người sử dụng với nút điều khiển chạy chương trình vẽ đồ thị câu 33.

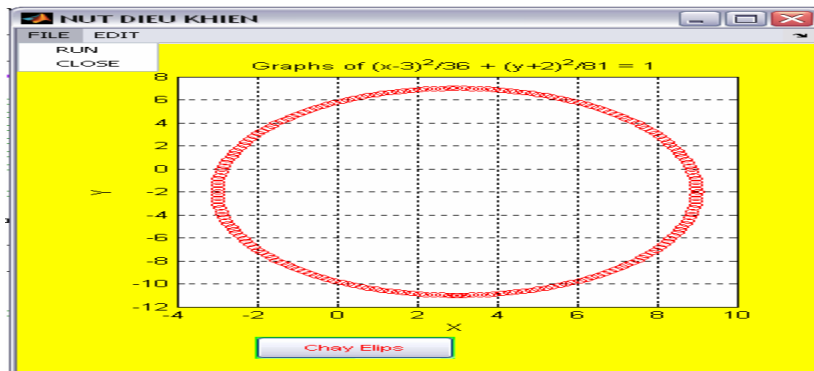
```
%chương trình chạy cos và sin theo nút điều khiển
h=figure('position',[100 100 400 450],'menubar','none',...
    'numbertitle','off','name','NUT DIEU KHIEN','color','y')
h0=uimenu('parent',h,'label','FILE')
%lệnh cho phép tạo tên nhân là FILE trên thanh menubar.
uimenu('parent',h0,'label','RUN','callback','elips7')
% tạo con của FILE có tên danh nhân là RUN
uimenu('parent',h0,'label','CLOSE','callback','close')
% tạo con của FILE có tên danh nhân là CLOSE
uicontrol('parent',h,'style','pushbutton','position',[120 20 100 30],'string','Chay
Elips',....
'backgroundcolor','g','foregroundcolor','r','visible','on','callback','elips7')
% Tạo nút điều khiển chạy chương trình là đồ thị hình SIN được lưu trong m-file.
% lệnh:'parent',h, : là thuộc tính cha mẹ là của số figure.
% lệnh: 'style','pushbutton' : thuộc tính tạo nút điều khiển push.
% lệnh: 'position' : là thuộc tính vị trí đỉnh của nút điều khiển.
% lệnh: 'BackgroundColor' : là thuộc tính tạo màu nền cho các nút điều khiển.
% lệnh: 'ForegroundColor' : là thuộc tính tạo màu văn bản cho các nút dk.
% lệnh : 'callback': là thuộc tính gọi các chương trình ứng dụng với giá
% trị là tên của chương trình được lưu dưới dạng m-file.
% lệnh: 'Visible': là thuộc tính hiển thị với 2 giá trị 'on' of 'off' các
% nút điều khiển khi chọn lệnh này.
h1=uimenu('parent',h,'label','EDIT')
%lệnh cho phép tạo tên nhân là EDIT trên thanh menubar.
uimenu('parent',h1,'label','CLEAR','callback','clc')
% tạo con của EDIT có tên danh nhân là 'clc' khi nhấn nó sẽ làm sạch của sổ
Window
uimenu('parent',h1,'label','CLOSE','callback','close')
% tạo con của EDIT có tên danh nhân là 'close'
```



```

axes('parent',h,'position',[0.2 0.2 0.7 0.7])
%cho phép tạo hệ trục xy của đồ thị sin(x) ở vị trí [0.2 0.2 0.7 0.7]
% End of program .

```

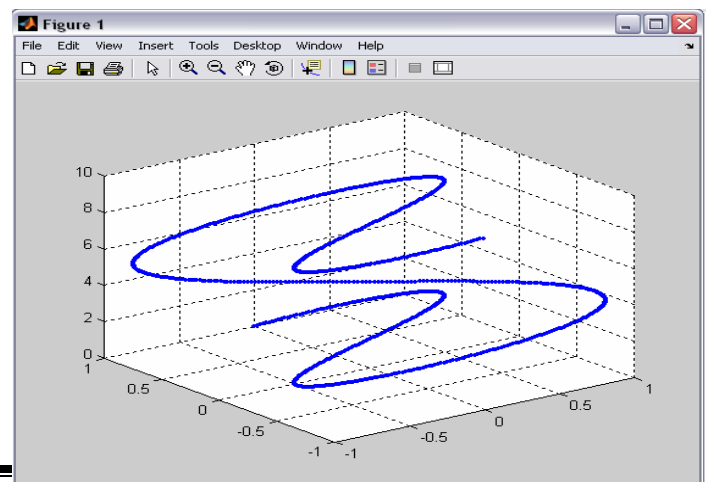


Bài 39 : Vẽ 3D bám động của hàm : $x = \sin(2t)$, $y = \cos(t)$, $z = t$

```

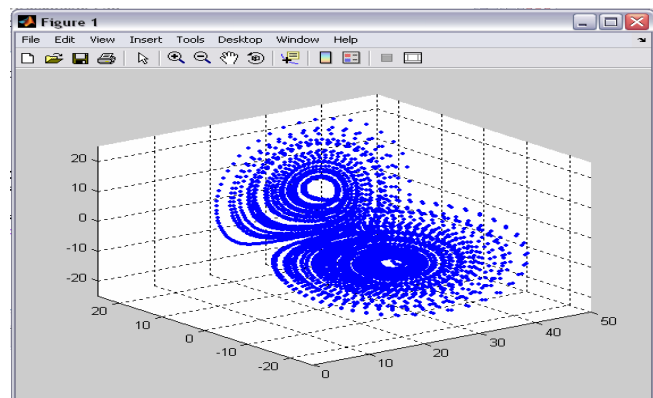
% Beginning of program
clear
t = 0;
x = sin(2*t);
y = cos(t);
z = t
p = plot3(x,y,z,'l','EraseMode','none','MarkerSize',5);
axis([-1 1 -1 1 0 10]),grid
hold on
for t = 0:0.01:3*pi
    x = sin(2*t);
    y = cos(t);
    z = t;
    set(p,'XData',x,'YData',y,'Zdata',z)
    drawnow
    pause(0.01)
end
%end of program.

```



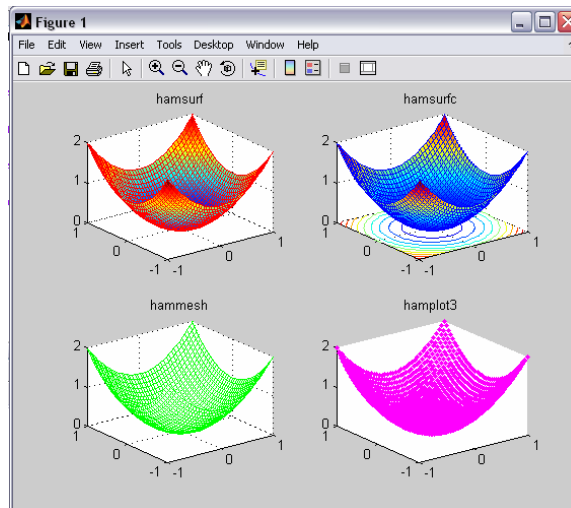
Bài 40 : Cho $dy/dt = Ay$, trong đó $A(y) = \begin{bmatrix} -8/3 & 0 & -y(2) \\ 0 & -10 & 10 \\ -y(2) & 28 & -1 \end{bmatrix}$. Nghiệm quỹ đạo chuyển động của phương trình vi phân này có thể được tìm bằng phương pháp Euler. Chương trình sau là một ví dụ minh chứng.

```
% beginning of program
clear
A = [-8/3 0 0;0 -10 10;0 28 -1];
y = [35 -10 -7]';
h = 0.01;
p = plot3(y(1),y(2),y(3),'.','EraseMode','none','MarkerSize',5);
axis([0 50 -25 25 -25 25]),grid
hold on
for i = 1:4000
    A(1,3) = y(2);
    A(3,1) = -y(2);
    ydot = A*y;
    y = y + h*ydot;
    set(p,'XData',y(1),'YData',y(2),'ZData',y(3))
    pause(0.01)
    drawnow
    i = i + 1;
end
% End of program .
```

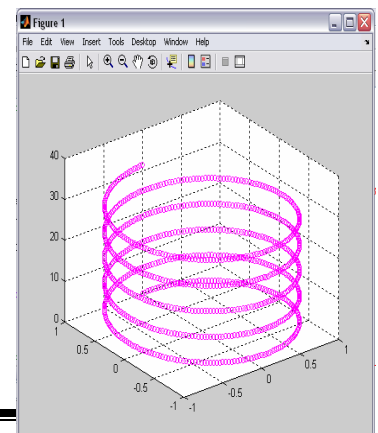


Bài 41: vẽ đồ thị 3 chiều :

```
% Beginning of program :
[x,y] = meshgrid(-1:0.05:1);
z = x.^2+y.^2;
subplot(2,2,1);
surf(x,y,z,'Edgecolor','r'),title('hamsurf');
subplot(2,2,2);
surfc(x,y,z,'Edgecolor','b'),title('hamsurfc');
subplot(2,2,3);
mesh(x,y,z,'Edgecolor','g'),title('hammesh');
subplot(2,2,4);
plot3(x,y,z,'linewidth',2.5,'linestyle','.', 'color','m'),title('hamplot3');
% End of program.
```

**Bài 42 : vẽ đồ thị 3 chiều bám động theo điểm :**

```
% vẽ ba chiều bám động theo điểm
clear
z = 0;
x = sin(z);
y = cos(z);
h = 0.01
p = plot3(x,y,z,'o','EraseMode','none','MarkerSize',5,'color','m');
axis([-1 1 -1 1 0 40]),grid
```



```

hold on
for z = 0:pi/100:10*pi
    x = sin(z);
    y = cos(z);
    set(p,'XData',x,'YData',y,'Zdata',z)
    drawnow
    pause(0.01)
end
% end of program

```

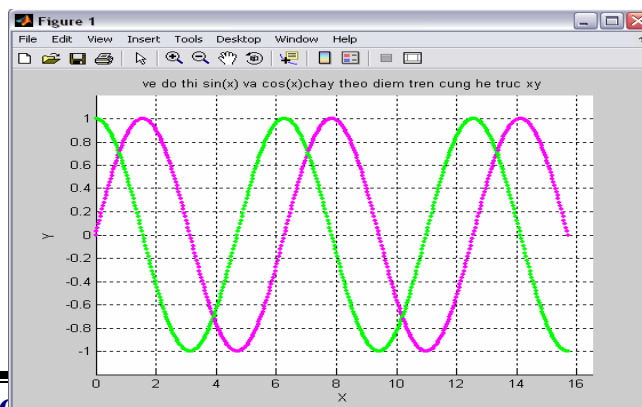
Bài 43 : Vẽ đồ thị $\sin(x)$ và $\cos(x)$ chạy bám động trên cùng hệ trục x y :

% vẽ đồ thị chạy $\sin(x)$ và $\cos(x)$ trên cùng hệ trục xy:

```

clear
x = 0;
y1 = sin(x);
y2 = cos(x);
hold on
p1 = plot(x,y1,'.', 'EraseMode','none','MarkerSize',5,'color','m');
p2 = plot(x,y2,'.', 'EraseMode','none','MarkerSize',5,'color','g');
axis([0 16.5 -1.2 1.2]),grid
hold on
for x = 0:pi/100:5*pi;
    y1 = sin(x);
    y2 = cos(x);
    set(p1,'XData',x,'YData',y1)
    set(p2,'XData',x,'YData',y2)
    xlabel('X'),ylabel('Y')
    title('vẽ đồ thị sin(x) và cos(x)chạy theo điểm trên cùng hệ trục xy')
    drawnow
    pause(0.01)
end
%end of program.

```





Bài 44: Vẽ dạng sóng hình sin của điện 3 pha :

% ve do thi chay sin(x) va cos(x) tren cung he truc xy:

clear

x = 0;

y1 = sin(x);

y2 = sin(x-2*pi/3);

y3 = sin(x+2*pi/3);

hold on

p1 = plot(x,y1,'.','EraseMode','none','MarkerSize',5,'color','m');

p2 = plot(x,y2,'.','EraseMode','none','MarkerSize',5,'color','g');

p3 = plot(x,y3,'.','EraseMode','none','MarkerSize',5,'color','r');

axis([0 16.5 -1.2 1.2]),grid

hold on

for x = 0:pi/100:5*pi;

y1 = sin(x);

y2 = sin(x-2*pi/3);

y3 = sin(x+2*pi/3);

set(p1,'XData',x,'YData',y1)

set(p2,'XData',x,'YData',y2)

set(p3,'XData',x,'YData',y3)

xlabel('X'),ylabel('Y')

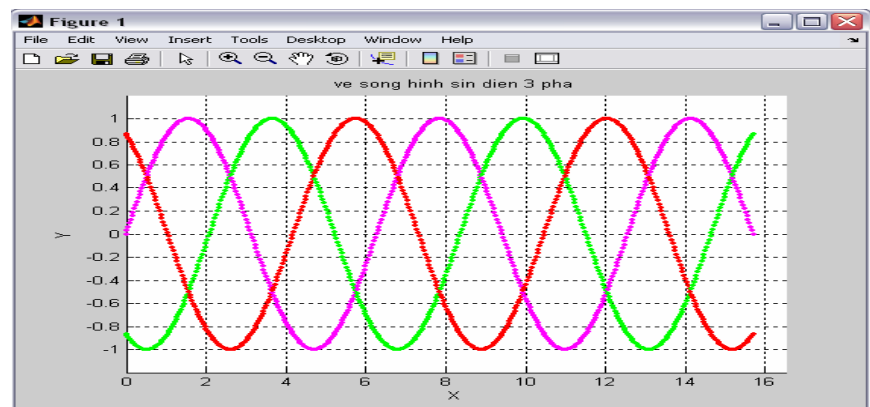
title('ve dang song hinh sin dien 3pha')

drawnow

pause(0.01)

end

%end of program.



Bài 45: Vẽ đồ thị đường tròn và elips cùng trên 1 hệ trục xy và bấm động theo điểm ,có phương trình sau: $(x-1)^2/36 + (y+1)^2/25 = 1$ and $(x-6)^2 + (y+3)^2 = 16$.

%Beginning of program : ve do thi hinh elips bam dong.

clear

t = 0;

x1 = 6 + 4*cos(t);

y1 = -3 + 4*sin(t);

x2 = 1 + 6*cos(t);

y2 = -1 + 5*sin(t);

h = 0.01 ;

hold on

p1 = plot(x1,y1,'r','linestyle','.', 'erasemode','none','markersize',5)

p2 = plot(x2,y2,'b','linestyle','o','erasemode','none','markersize',5)

axis([-8 12 -12 8]),grid

hold on

for t = 0:pi/100:5*pi;

 x1 = 6 + 4*cos(t);

 y1 = -3 + 4*sin(t);

 x2 = 1 + 6*cos(t);

 y2 = -1 + 5*sin(t);

 set(p1,'Xdata',x1,'Ydata',y1)

 set(p2,'Xdata',x2,'Ydata',y2)

 drawnow

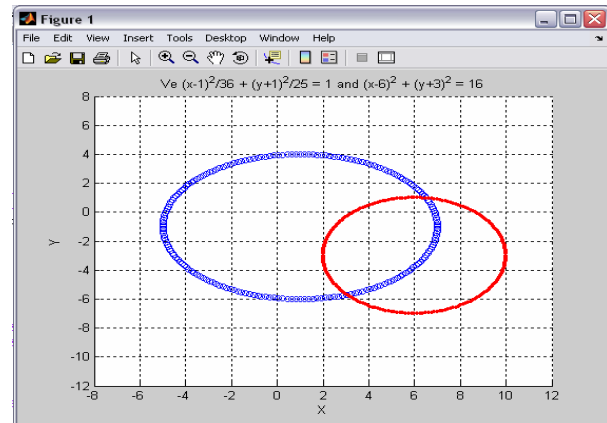
 pause(0.005)

 xlabel('X'),ylabel('Y')

 title('Ve $(x-1)^2/36 + (y+1)^2/25 = 1$ and $(x-6)^2 + (y+3)^2 = 16$ ')

end

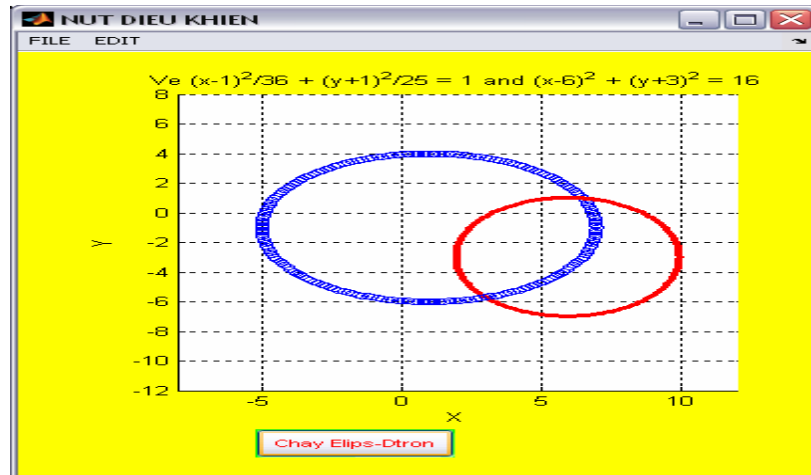
%End of program



**Bài 46: chương trình tọa giao diện người sử dụng với nút điều khiển chạy chương trình vẽ đồ thị Elips – Đ tròn câu 47:**

```
%chương trình chạy elips dtron theo nút điều khiển
h = figure('position',[100 100 400 450],'menubar','none',...
    'numbertitle','off','name','NUT DIEU KHIEN','color','y')
h0 = uimenu('parent',h,'label','FILE')
%lệnh cho phép tạo tên nhân là FILE trên thanh menubar.
uimenu('parent',h0,'label','RUN','callback','elipsdtron2')
% tạo con của FILE có tên danh nhân là RUN
uimenu('parent',h0,'label','CLOSE','callback','close')
% tạo con của FILE có tên danh nhân là CLOSE
uicontrol('parent',h,'style','pushbutton','position',[120 20 100 30],'string','Chạy
Elips-Dtron',....
'backgroundcolor','g','foregroundcolor','r','visible','on','callback','elipsdtron2')
% Tạo nút điều khiển chạy chương trình là đồ thị hình Elipsdtron được lưu trong
m-file.
% lệnh:'parent',h, : là thuộc tính cha do là của số figure.
% lệnh: 'style','pushbutton' : thuộc tính tạo nút điều khiển push.
% lệnh: 'position' : là thuộc tính vị trí định vị của nút điều khiển.
% lệnh: 'BackgroundColor' : là thuộc tính tạo màu nền cho các nút điều khiển.
% lệnh: 'ForegroundColor' : là thuộc tính tạo màu văn bản cho các nút dk.
% lệnh : 'callback': là thuộc tính gọi các chương trình ứng dụng với giá
% trị là tên của chương trình được lưu dưới dạng m-file.
% lệnh: 'Visible': là thuộc tính hiển thị với 2 giá trị 'on' of 'off' các
% nút điều khiển khi chọn lệnh này.
h1 = uimenu('parent',h,'label','EDIT')
%lệnh cho phép tạo tên nhân là EDIT trên thanh menubar.
uimenu('parent',h1,'label','CLEAR','callback','clc')
% tạo con của EDIT có tên danh nhân là 'clc'khi nhấn nó sẽ làm sạch của sổ
Window
uimenu('parent',h1,'label','CLOSE','callback','close')
% tạo con của EDIT có tên danh nhân là 'close'
axes('parent',h,'position',[0.2 0.2 0.7 0.7])
%cho phép tạo hệ trục xy của đồ thị sin(x) ở vị trí [0.1 0.2 0.35 0.7]
% End of program.
```





Bài 47 : chương trình tính tổng dãy số : $s = x/1 - x^2/2! - x^3/3! + x^4/4!...$

% chương trình tính tổng dãy số : $s = x/1 - x^2/2! - x^3/3! + x^4/4!...$

```
clear
```

```
n = input('Enter n :');
```

```
x = input('Enter x :'); % các giá trị của x luôn giống nhau.
```

```
i = 1;
```

```
count = 1;
```

```
sum = 0 ;
```

```
while i <= n
```

```
    fact = 1;
```

```
    for k = 1:i
```

```
        fact = fact*k;
```

```
    end
```

```
    if count == 1
```

```
        sum = sum + x^i/fact;
```

```
    else
```

```
        sum = sum - x^i/fact;
```

```
    end
```

```
    count = count + 1;
```

```
    if count == 4
```

```
        count = 1;
```

```
    end
```

```
    i = i + 1;
```




```
end  
sum  
%End of program .
```

Ví dụ nhập giá trị $x(i)$ như sau:

```
Enter n :4  
Enter x :2  
sum =  
-0.6667
```

Bài 48 : cấu trúc thiết bị : tính tổng các cột : Sl , p , Umin , Imin, Cosphi, Ksd:

```
% chong trinh tao mang cau truc  
i = 1;  
sum(1) = 0;  
sum(2) = 0;  
sum(3) = 0;  
sum(4) = 0;  
sum(5) = 0;  
sum(6) = 0;  
while i <= 3  
    Tb(i).ten = input('Nhap ten thiet bi :','s');  
    Tb(i).kyhieu = input('Nhap ky hieu thiet bi :','s');  
    Tb(i).Sl = input('Nhap so luong :');  
    Tb(i).p = input('Nhap cong suat :');  
    Tb(i).Umin = input('Nhap dien ap min :');  
    Tb(i).Imin = input('Nhap dong dien min :');  
    Tb(i).Cosphi = input('Nhap he so cosphi:');  
    Tb(i).Ksd = input('Nhap Ksd :');  
    sum(1) = sum(1) + Tb(i).Sl;  
    sum(2) = sum(2) + Tb(i).p;  
    sum(3) = sum(3) + Tb(i).Umin;  
    sum(4) = sum(4) + Tb(i).Imin;  
    sum(5) = sum(5) + Tb(i).Cosphi;  
    sum(6) = sum(6) + Tb(i).Ksd;  
    i = i + 1;  
end
```





```
end
sum(1)
sum(2)
sum(3)
sum(4)
sum(5)
sum(6)
% End of program .
```

Bài 49: tính tổng $S = \text{tổng} [x(i) * i!]$:

Trường hợp giá trị $x(i)$ khác nhau:

```
% tính s = tong[x(i)*i!]
n = input('Enter nhap n :');
i = 1;
s = 0;
fact = 1;
k = 1;
while i <= n
    x(i) = input('Nhap gia tri x(i) :');
    fact = fact*k;
    s = s + x(i)*fact;
    i = i + 1;
    k = k + 1;
end
s
% End of program.
```

Ví dụ nhập giá trị $x(i)$ như sau:

```
Enter nhap n :4
Nhap gia tri x(i) :2
Nhap gia tri x(i) :3
Nhap gia tri x(i) :4
Nhap gia tri x(i) :5
```

```
s =
```



152

Trường hợp giá trị $x(i)$ giống nhau :

```
% tính tong[x*i!]  
n = input('Enter nhap n :');  
x = input('Nhap gia tri x :');  
i = 1;  
s = 0;  
k = 1;  
while i <= n  
    fact = 1;  
    for k = 1:i  
        fact = fact*k;  
    end  
    s = s + x*fact;  
    i = i + 1;  
end  
s  
% End of program.
```

Ví Dụ Như Sau :

Enter nhap n :4

Nhap gia tri x :2

s =

66

**Bài 50 : Xác định tâm của phụ tải : $X = \text{tổng} [x(i) * p(i)] / \text{tổng} [p(i)]$
và $Y = \text{tổng} [y(i) * p(i)] / \text{tổng} [p(i)]$**

% xác định tâm của phụ tải cung cấp diện :

i = 1;

x1 = 0;

y1 = 0;

p1 = 0;



```
n = input('Nhap n :');  
while i <= n  
    x(i)= input('Nhap gia tri x(i) :');  
    y(i)= input('Nhap gia tri y(i) :');  
    p(i)= input('Nhap cong suat p(i) :');  
    x1 = x1 + x(i)*p(i);  
    y1 = y1 + y(i)*p(i);  
    p1 = p1 + p(i);  
    i = i + 1;  
end  
x = x1/p1  
y = y1/p1  
% End of program .
```

Ví dụ như sau :

```
Nhap n :4  
Nhap gia tri x(i) :1  
Nhap gia tri y(i) :2  
Nhap cong suat p(i) :20  
Nhap gia tri x(i) :2  
Nhap gia tri y(i) :3  
Nhap cong suat p(i) :30  
Nhap gia tri x(i) :3  
Nhap gia tri y(i) :4  
Nhap cong suat p(i) :20  
Nhap gia tri x(i) :4  
Nhap gia tri y(i) :1  
Nhap cong suat p(i) :20  
x =  
    2.4444  
y =  
    2.5556
```



Bài 52 : chương trình vẽ đường tròn bán động theo điểm :

```
%Beginning of program : ve do thi hinh tron bam dong.
a = input('nhap a :');
b = input('nhap b :');
r = input('nhap r :');
t = 0;
x = a + r*cos(t);
y = b + r*sin(t);
h = 0.01;
p = plot(x,y,'r','linestyle','o','erasemode','none','markersize',5),grid
axis([-4 10 -12 8]);
hold on
for t = 0:pi/100:5*pi;
    x = a + r*cos(t);
    y = b + r*sin(t);
    set(p,'Xdata',x,'Ydata',y)
    drawnow
    pause(0.01)
    xlabel('X'),ylabel('Y')
    title('Graphs of  $(x - a)^2 + (y - b)^2 = r^2$ ')
end
%End of program
```

Bài 53 : tính tổng $S = \sum [x(i) / i!]$:

Trường hợp giá trị $x(i)$ khác nhau:

```
% tinh tong[x(i)/i!]
n = input('Enter nhap n :');
i = 1;
s = 0;
fact = 1;
k = 1;
while i <= n
    x(i) = input('Nhap gia tri x(i) :');
    fact = fact*k;
    s = s + x(i)/fact;
```



```
i = i + 1;  
k = k + 1;  
end  
s  
% End of program.
```

Ví Dụ Như :

Enter nhập n :3
Nhập giá trị x(i) :2
Nhập giá trị x(i) :3
Nhập giá trị x(i) :4

```
s =  
4.1667
```

Trường hợp giá trị x(i) giống nhau :

```
% tính tong[x/i!]  
n = input('Enter nhập n :');  
x = input('Nhập giá trị x :'); % Gia tri cua x bang nhau  
i = 1;  
s = 0;  
k = 1;  
while i <= n  
    fact = 1;  
    for k = 1:i  
        fact = fact*k;  
    end  
    s = s + x/fact;  
    i = i + 1;  
end  
s  
% End of program.
```

Ví Dụ Cụ Thể :

Enter nhập n :4
Nhập giá trị x :2
s =
3.4167





Bài 54 : chương trình tính tổng dãy số : $s = x/1 + x^2/2! - x^3/3! - x^4/4! + x^5/5! + x^6/6! - x^7/7! - x^8/8! \dots$

% chương trình tính tổng dãy số : $s = x/1 + x^2/2! - x^3/3! - x^4/4! \dots$

clear

n = input('Enter n :');

x = input('Enter x :');

i = 1;

count = 1;

sum = 0 ;

while i <= n

 fact = 1;

 for k = 1:i

 fact = fact*k;

 end

 if count == 1

 sum = sum + x^i/fact;

 elseif count == 2

 sum = sum + x^i/fact;

 else

 sum = sum - x^i/fact;

 end

 count = count + 1;

 if count == 5

 count = 1 ;

 end

 i = i + 1;

end

sum

% End of program .

Ví Dụ Như :

Enter n :8

Enter x :2

sum =

2.3238





Hoặc
Enter n :4
Enter x :2
sum =
2.0000

Bài 55 : Chương trình tính tổng dãy số :

```
S = 2+x^3/3! + x^6/6! + x^9/9! +...+ x^n/n!  
% Begining of program : s = 2+x^3/3! + x^6/6! + x^9/9! +...+ x^n/n!  
clear  
n = input('Enter n :');  
x = input('Enter x :');  
i = 3;  
count = 3;  
sum = 2 ;  
while i <= n  
    fact = 1;  
    for k = 1:i  
        fact = fact*k;  
    end  
    if count == 3  
        sum = sum + x^i/fact;  
    else  
        sum = sum ;  
    end  
    count = count + 1;  
    if count == 6  
        count = 3 ;  
    end  
    i = i + 1;  
end  
sum  
%End of program .
```

Ví Dụ Như :
Enter n :9





Enter x :2
sum =
3.4236

Bài 56 : Chương trình tính tổng dãy số : với giá trị x(i) bất kỳ :

```
%dung lenh while and for tinh tong s = x1/1! + x2^2/2! +....+ x^n/n!  
n = input('Enter n :');  
i = 1;  
s = 0;  
k = 1;  
fact = 1;  
while i <= n;  
    x = input('Enter x :');  
    fact = fact*k;  
    s = s + (x^i)/fact;  
    k = k + 1;  
    i = i + 1;  
end  
s  
% end of program
```

Ví Dụ Như :

```
Enter n :4  
Enter x :3  
Enter x :2  
Enter x :5  
Enter x :6  
s =  
79.8333
```

Bài 57 : Chương trình cấu trúc bảng điểm :

```
%chuong trinh cau truc bang diem  
clear  
con = 'y';  
i = 1;
```





```
while con == 'y'
    n = i;
    lop(i).STT = input('enter STT :','s');
    lop(i).name = input('enter name :','s');
    lop(i).diem = input('enter diem :');
    if lop(i).diem >= 9 lop(i).hang = 'Gioi';
    elseif lop(i).diem < 9 && lop(i).diem >= 7 lop(i).hang = 'Kha';
    elseif lop(i).diem < 7 && lop(i).diem >= 5 lop(i).hang = 'Trung Binh';
    else lop(i).hang = 'Yeu';
end
    i = i + 1;
    con = input('continue thanh vien khac y/n :','s');
end
for i = 1:n;
    lop(i)
end
% End of program
```

Ví Dụ Như :

```
enter STT :1
enter name :Hung
enter diem :9
continue thanh vien khac y/n :y
enter STT :2
enter name :Anh
enter diem :7
continue thanh vien khac y/n :y
enter STT :3
enter name :Sang
enter diem :6
continue thanh vien khac y/n :y
enter STT :4
enter name :Giang
enter diem :4
continue thanh vien khac y/n :n
```

```
ans =
    STT: '1'
```





```

    name: 'Hung'
    diem: 9
    hang: 'Gioi'
ans =
    STT: '2'
    name: 'Anh'
    diem: 7
    hang: 'Kha'
ans =
    STT: '3'
    name: 'Sang'
    diem: 6
    hang: 'Trung Binh'
ans =
    STT: '4'
    name: 'Giang'
    diem: 4
    hang: 'Yeu'

```

Bài 58 : Chương trình tính tổng : $P = a_n * x^n + a_{n-1} * x^{(n-1)} + \dots + a_1 * x + a_0$

```

% chương trình tính tổng :p = an*x^n + a(n-1)*x^(n-1) + ...+ a1*x + a0
clear
n = input('Enter n :');
x = input('Enter x :');
i = 0;
sum = 0 ;
while i <= n
    k = n;
    a.n = input('Enter a.n :');
    sum = sum + a.n*x^k;
    n = n -1;
end
sum
%End of program .

```

Ví Dụ Như :





Enter n :5
Enter x :2
Enter a.n :5
Enter a.n :3
Enter a.n :4
Enter a.n :2
Enter a.n :1
Enter a.n :6
sum =
256

Bài 59 : : chương trình tính tổng dãy số : $s = x/1 + x^2/2! - x^3/3! + x^4/4! - x^5/5! + x^6/6! - x^7/7! + x^8/8! \dots$

```
% chương trình tính tổng dãy số :  $s = x/1 + x^2/2! - x^3/3! + x^4/4! - x^5/5! + \dots$ 
clear
n = input('Enter n :');
x = input('Enter x :');
i = 1;
count = 1;
sum = 0 ;
while i <= n
    fact = 1;
    for k = 1:i
        fact = fact*k;
    end
    if count == 1
        sum = sum + x^i/fact;
    elseif count == 2
        sum = sum + x^i/fact;
    else
        sum = sum - x^i/fact;
    end
    count = count + 1;
    if count == 4
        count = 2 ;
    end
    i = i + 1;
end
```





```
end
i = i + 1;
end
sum
%End of program .
```

Ví Dụ Như :

Enter n :5

Enter x :2

sum =
3.0667

Bài 60 : Tính tổng sau: $S = [\text{tổng } x(i)]^2 / [\text{tổng } x(i)^2]$

```
% tính tong: S = [tong x(i)]^2/[tong x(i)^2]
n = input('Enter nhap n :');
i = 1;
s1 = 0;
s2 = 0;
while i <= n
    x(i) = input('Nhap gia tri x(i) :');
    s1 = s1 + x(i);
    s2 = s2 + x(i)^2;
    i = i + 1;
end
s = s1^2/s2
% End of program.
```

Ví Dụ Như sau :

Enter nhap n :5

Nhap gia tri x(i) :2

Nhap gia tri x(i) :3

Nhap gia tri x(i) :4

Nhap gia tri x(i) :5

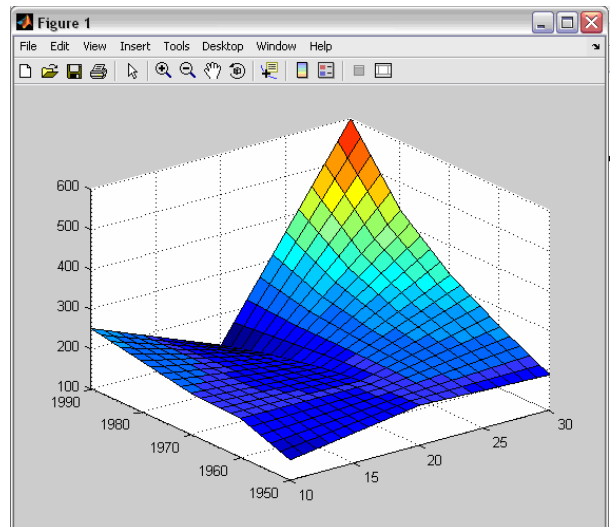
Nhap gia tri x(i) :6

s =
4.4444



Bài 61: Vẽ Đồ Thị Nội Suy 2 Chiều :

```
% begining of program
years = 1950:10:1990;
service = 10:10:30;
wage = [150.697 199.592 187.625;197.323 195.072 250.287;....
        203.212 179.092 322.767;226.505 153.706 426.730;249.633 120.281 598.234];
x = 10:1:30;
y = 1950:2:1990;
z = interp2(service,years,wage,15,1975);
[x,y] = meshgrid(10:1:30,1950:2:1990);
z = interp2(service,years,wage,x,y);
surf(x,y,z)
% End of program
```

**Bài 62 : tính trị của biểu thức : $T = \sum_{i=1:n} k_i \cdot x^i / i!$**

$k_i = 1$ neu i không phải là một số nguyên tố (2,4,6,8...)

$k_i = -1$ neu i là một số nguyên tố (1,3,5,7,9....)

```
% Beginning of program
```

```
clear
```

```
n = input('Enter n :');
```

```
x = input('Enter x :');
```

```
i = 1;
```

```
count = 1;
```

```
sum = 0 ;
```

```
while i <= n
```

```
    fact = 1;
```



```
for k = 1:i
    fact = fact*k;
end
if count == 1
    sum = sum - x^i/fact;
else
    sum = sum + x^i/fact;
end
count = count + 1;
if count == 3
    count = 1 ;
end
i = i + 1;
end
sum
%End of program .
```

Ví Dụ Như :

Enter n :7

Enter x :2

sum =

-0.8698

Bài 63 : Chương trình tính tổng các tích 2 số liên tiếp từ 1 đến n :

$$s = 1*2 + 2*3 + 3*4 + 4*5 + \dots + (n-1)*n$$

% Begining of program

n = input('Enter n:');

sum = 0;

for i = 1:n

sum = sum + (i-1)*i;

end

sum

% End of program

Ví Dụ Như :

Enter n:8

sum =

168





3 Bài Thi Học Kỳ Vừa Rồi

Bài 2 : Chương trình mang cấu trúc và tính cosphi trung bình và dòng diện :

```
% chong trình tao mang cau truc thiat bi:
n = input('enter n :');
i = 1;
sum(1) = 0;
sum(2) = 0;
while i <= n
    TB(i).ten = input('Nhap ten thiet bi :','s');
    S(i) = input('Nhap so luong :');
    P(i) = input('Nhap cong suat :');
    U(i) = input('Nhap dien ap :');
    C(i) = input('Nhap cosphi :');
    sum(1) = sum(1) + P(i)*S(i)*C(i);
    sum(2) = sum(2) + P(i);
    I(i) = P(i)/(sqrt(3)*U(i)*C(i))
    i = i + 1;
end
Ctb = sum(1)/sum(2)
%End of program
```

Bài 3 : Chương trình vẽ đồ thị nội suy cosphi bám động theo điểm , thiết lập tỉ lệ hệ trục tọa độ , có đánh nhãn và tiêu đề :

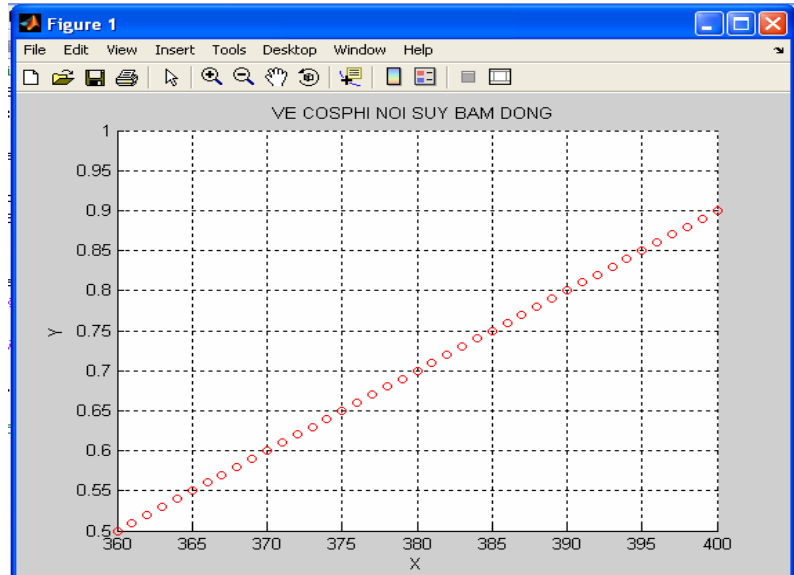
```
% Begining of program
U = 360:5:400;
C = 0.5:0.05:0.9;
x = 0 ;
y = interp1(U,C,x,'spline');
hold on
p = plot(x,y,'o','Erasemode','none','Markersize',5,'color','r')
axis([360 400 0.5 1]),grid
hold on
```




```

for x = 360:1:400 ;
y = interp1(U,C,x,'spline');
set(p,'Xdata',x,'Ydata',y)
Xlabel('X'),Ylabel('Y')
title('Ve noi suy cosphi')
drawnow
pause(0.01)
end
% End of program

```



Bài 4 : Chương trình giao diện có tiêu đề , hệ trục tọa độ , thực đơn FILE chứa lệnh chạy và nút điều khiển chương trình vẽ đồ thị o câu 68:

```

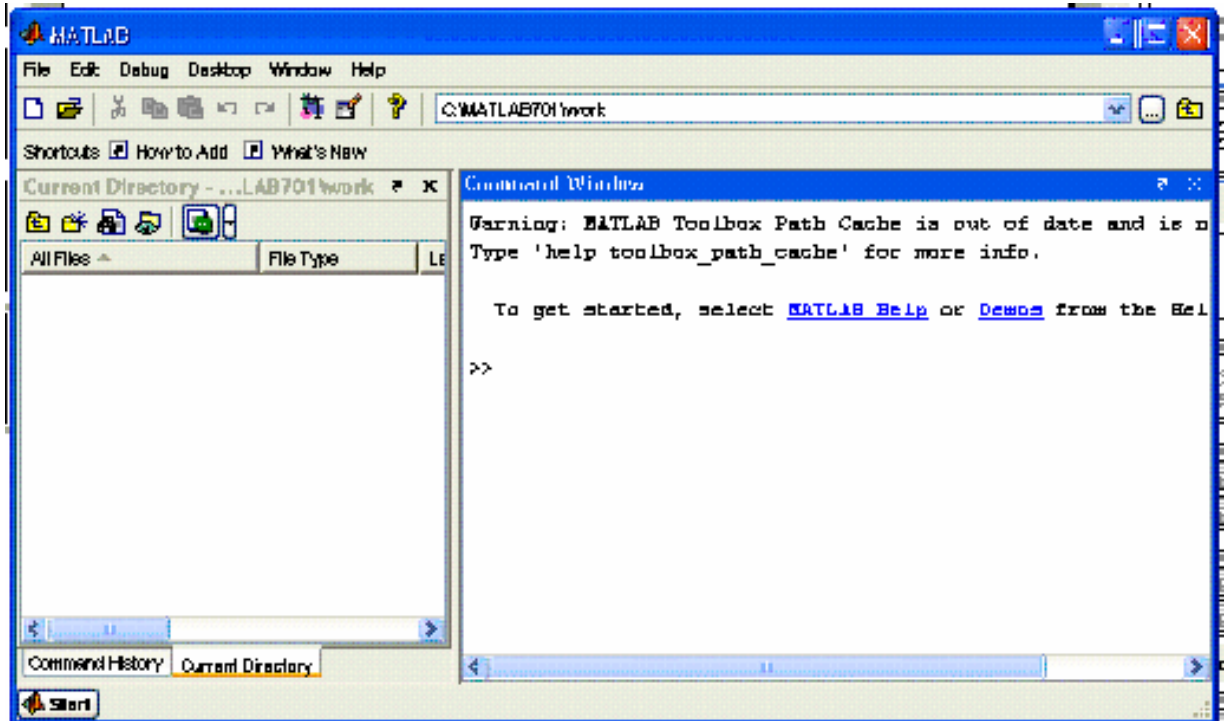
%chương trình tạo giao diện sử dụng chạy chương trình vẽ đồ thị nội suy cosphi
h = figure('position',[80 100 400 500],'menubar','none',...
    'numbertitle','off','name','NUT DIEU KHIEN','color','y')
h0 = uimenu('parent',h,'label','FILE')
%lệnh cho phép tạo tên nhân là FILE trên thanh menubar.
uimenu('parent',h0,'label','RUN','callback','noisuy')
% tạo con của FILE có tên danh nhân là RUN
uimenu('parent',h0,'label','CLOSE','callback','close')
% tạo con của FILE có tên danh nhân là CLOSE
uicontrol('parent',h,'style','pushbutton','position',[120 20 100 30],...,
    'string','ve noi suy cosphi','backgroundcolor','g',...
    'foregroundcolor','r','callback','noisuy')
axes('parent',h,'position',[0.2 0.2 0.7 0.7])
%End of program

```

BÀI TẬP THỰC HÀNH SỐ 1

Thực hành lệnh, kiểu dữ liệu, toán tử, truy cập, mảng, hàm trên cửa sổ lệnh.

Làm quen với môi trường cửa sổ làm việc Matlab.



- + Command Window (Cửa sổ lệnh).
- + Current Directory (Cửa sổ thư mục hiện hành).
- + Command History (Cửa sổ lịch sử).
- + Work Space (Không gian làm việc)

I .Các chức năng trên cửa sổ lệnh:

Clc: làm sạch màn hình.

Clear: làm sạch cửa sổ workspace.

Help: lấy thông tin trợ giúp.

Edit: soạn thảo chương trình.

Type: xem nội dung của file.

←, → : dịch chuyển con trỏ trái phải.

↑, ↓: chọn lại những lệnh đã thực hiện.

II . Gán các kiểu dữ liệu cho biến:

Thực hiện các dòng lệnh sau và kiểm tra kết quả.

>> a = 5 % gán dữ liệu số nguyên

a =
5



```
>> b = 7.43 % gán dữ liệu số thực
b =
    7.4300
```

```
>> c = 3+4i % gán dữ liệu số phức
c =
    3.0000 + 4.0000i
```

```
>> d = pi % gán số pi = 3.14159265 cho biến d
d =
    3.1416
```

```
>> e = eps % gán số chính xác của dấu chấm động  $2^{-52}$ 
e =
    2.2204e-016
```

```
>> f = realmin % gán số chấm động dương nhỏ nhất  $2^{-1022}$ 
f =
    2.2251e-308
```

```
>> g = realmax % gán số chấm động dương lớn nhất  $2^{1022}$ 
g =
    1.7977e+308
```

```
>> h = nan % không phaỉi số
h =
    NaN
```

III .định dạng các kiểu dữ liệu số với lệnh format : long, long e, short, short e, trong đó e là cơ số 10.

Thực hiện 2 dòng lệnh sau:

```
>> format long % số chấm cố định là 15 con số
>> A = pi
A =
    3.14159265358979
```

Thực hiện tiếp 2 dòng lệnh sau rồi so sánh kết quả.

```
>> format long e % số dấu chấm động là 15 con số
>> A = pi
A =
    3.141592653589793e+000
```





So sánh kết quả khi lần lượt thực hiện các dòng lệnh trên cho thấy : kết quả bằng nhau. Những biểu diễn khác nhau:

vì $A = 3,14159265358973e+000$ trong đó $e + 000 = 10^0 = 1$

Tương tự thực hiện các dòng lệnh sau.

```
>> format short      % số chấm cố định là 5 con số ( mặc định )
```

```
>> A = pi
```

```
A =
```

```
3.1416
```

Và

```
>> format short e    % số dấu chấm động là 5 con số
```

```
>> A = pi
```

```
A =
```

```
3.1416e+000
```

Tổng thì so sánh kết quả khi lần lượt thực hiện các dòng lệnh tiếp theo ở trên cho thấy : kết quả bằng nhau. Những biểu diễn khác nhau:

vì $A = 3,1416e+000$ trong đó $e + 000 = 10^0 = 1$

IV .Qui ước biến và thực hiện các phép toán trên cửa sổ lệnh.

Cộng +, trừ -, nhân *, chia phải /, chia trái \, lũy thừa ^.

Độ ưu tiên	Phép toán	Tính ưu tiên
1	(,)	trong ra ngoài
2	^	trái qua phải
3	$\pm a$	
4	*, / , \	trái qua phải
5	+, -	trái qua phải

Tính bằng tay các biểu thức sau đó thực hiện bằng matlab và lưu lại kết quả.

a. $2 / 2 * 3$

✓ Tính bằng tay: $2/2*3 = (2/2)*3 = 3$

✓ Tính bằng matlab :

```
>> 2/2*3
```

```
ans =
```

```
3
```





b. $6 - 2 / 5 + 7^2 - 1$

✓ Tính bằng tay: $6 - 2/5 + 7^2 - 1 = 6 - 0,4 + 49 - 1 = 53,6$

✓ Tính bằng matlab :

```
>> 6-2/5+7^2-1
```

```
ans =
```

```
53.6000
```

c. $10 / 2 \setminus 5 - 3 + 2 * 4$

✓ Tính bằng tay: $10/2 \setminus 5 - 3 + 2*4 = 5/(10/2) - 3 + 8 = 5 / 5 - 3 + 8 = 6$

✓ Tính bằng matlab :

```
>> 10/2\5-3+2*4
```

```
ans =
```

```
6
```

Dùng matlab tính các biểu thức sau và lưu lại kết quả:

a. $\frac{3+4}{5+\sqrt{2}}$

```
>> ( 3 + 4 )/(5+sqrt(2))
```

```
ans =
```

```
1.0913
```

b. $2\pi^2$

```
>> 2*pi^2
```

```
ans =
```

```
19.7392
```

c. $\sqrt[3]{4.5}$

```
>> 4.5^(1/3)
```

```
ans =
```

```
1.6510
```

V. Làm việc với mảng trên cửa sổ lệnh.

1. Gán dữ liệu nguyên, thực, phức cho mảng một chiều, hai chiều bằng **chỉ số**.

- mảng 1 chiều.

Thực hiện các dòng lệnh sau lưu lại kết quả .

```
>> X(3) = 4 % phần tử thứ 3 của mảng là 4.
```

```
X =
```

```
0 0 4
```





```
>> X(2) = 1 + 3i    % phần tử thứ 2 của mảng là 1 + 3i
X =
    0    1.0000 + 3.0000i    4.0000
```

- mảng 2 chiều :

Thực hiện các dòng lệnh sau lưu lại kết quả.

```
>> T(1,1) = 1    % gán giá trị cho mảng theo chỉ số ,
T =
    1
```

```
>> T(1,2) = 46
T =
    1    46
```

```
>> T(2,1) = 3 + 2i
T =
    1.0000    46.0000
    3.0000 + 2.0000i    0
```

```
>> T(2,2) = 4.34
T =
    1.0000    46.0000
    3.0000 + 2.0000i    4.3400
```

2. Gán dữ liệu nguyên, thực, phức cho mảng một chiều, hai chiều bằng nội dung.

- mảng 1 chiều.

Thực hiện các dòng lệnh sau lưu lại kết quả:

```
>> x = [ 3 1 5 ]    % gán vevtor hàng 3 1 5 cho biến x.
x =
    3    1    5
```

```
>> y = [ 2.1 4.6 3.8 ]
y =
    2.1000    4.6000    3.8000
```

```
>> z = [ 2+3i 7-4i 9+5i ]
z =
    2.0000 + 3.0000i    7.0000 - 4.0000i    9.0000 + 5.0000i
```





- mảng 2 chiều:

Thực hiện các dòng lệnh sau lưu lại kết quả.

```
>> t = [ 6 5 ; 5 3 ]
```

```
t =
```

```
6    5
5    3
```

```
>> u = [ 8.1 3.6 ; 3.5 7.2 ]
```

```
u =
```

```
8.1000    3.6000
3.5000    7.2000
```

```
>> v = [ 3+i 7-7i ; 8-4i 4+5i ]
```

```
v =
```

```
3.0000 + 1.0000i    7.0000 - 7.0000i
8.0000 - 4.0000i    4.0000 + 5.0000i
```

3. Truy cập mảng một chiều, hai chiều bằng chỉ số.

$A(i,j)$ truy cập phần tử có chỉ số i,j

$A(1:k,j)$ truy cập từ phần tử thứ nhất đến phần tử thứ k của cột j

$A(i,1:k)$ truy cập từ thứ 1 đến phần tử thứ k của hàng i

Trước tiên ta gán nội dung cho mảng.

```
>> z = [ 2    3.4  4+3i ; 2    3+2i  7.2 ]
```

```
z =
```

```
2.0000    3.4000    4.0000 + 3.0000i
2.0000    3.0000 + 2.0000i    7.2000
```

Truy cập phần tử có chỉ số 2,1

```
>> z(2,1)    % truy cập phần tử thứ 2 của cột 1.
```

```
ans =
```

```
2
```

Truy cập từ phần tử thứ 1 đến phần tử thứ 2 của cột 1

```
>> z(1:2,1)    % truy cập phần tử : từ 1 đến 2 của cột 1.
```

```
ans =
```

```
2
```

```
2
```

Truy cập từ phần tử thứ 1 đến phần tử thứ 3 của hàng 2.

```
>> z(2,1:3)    % truy cập phần tử thứ 1 đến 3 của hàng 2.
```

```
ans =
```

```
2.0000    3.0000 + 2.0000i    7.2000
```



**VI. Thực hành một số hàm với mảng trên cửa sổ lệnh.**

- sum, prod, diff, trace, diag, eye, ones, zeros, magic, randn, inv, eig . . .
- sum(A) : tạo ra vector hàng với các phần tử là tổng của các phần tử của các cột tương ứng.
 - prod(A) : tạo ra vector hàng với các phần tử là tích của các phần tử của các cột tương ứng.
 - diff(A) : tính các sai biệt giữa các phần tử kế nhau của ma trận A.
 - trace(A) : tính vết của ma trận A (tổng các phần tử trên đường chéo).
 - eye(n) : tạo ra ma trận đơn vị.
 - magic(n) : tạo ma trận vuông với tổng các phần tử hàng, cột và đường chéo là bằng nhau.
 - rand(n) : tạo ma trận vuông với các phần tử được phân bố ngẫu nhiên.

Thực hiện các lệnh sau và lưu lại kết quả.

```
>> z = [ 3.4 4+3i ; 2 3+2i ] % gán ma trận cấp 2x2 cho biến z
```

```
z =  
    3.4000    4.0000 + 3.0000i  
    2.0000    3.0000 + 2.0000i
```

```
>> I = sum(z) % tính tổng ma trận theo cột tương ứng.
```

```
I =  
    5.4000    7.0000 + 5.0000i
```

```
>> J = prod(z) % tính tích các cột tương ứng
```

```
J =  
    6.8000    6.0000 + 17.0000i
```

```
>> K = diff(z)
```

```
K =  
   -1.4000   -1.0000 - 1.0000i
```

```
>> L = trace(z)
```

```
L =  
    6.4000 + 2.0000i
```





```
>> P = eye(4)
```

```
P =
```

```
1  0  0  0
0  1  0  0
0  0  1  0
0  0  0  1
```

```
>> Q = magic(3)
```

```
Q =
```

```
8  1  6
3  5  7
4  9  2
```

```
>> M = rand(5)
```

```
M =
```

```
0.9501  0.7621  0.6154  0.4057  0.0579
0.2311  0.4565  0.7919  0.9355  0.3529
0.6068  0.0185  0.9218  0.9169  0.8132
0.4860  0.8214  0.7382  0.4103  0.0099
0.8913  0.4447  0.1763  0.8936  0.1389
```

1. Cho ma trận $A = [2 \ 4 \ 1 ; 6 \ 7 \ 2 ; 3 \ 5 \ 9]$, viết lệnh Matlab để :

```
>> A = [ 2 4 1 ; 6 7 2 ; 3 5 9 ]
```

```
A =
```

```
2  4  1
6  7  2
3  5  9
```

- a. Tạo vector x là hàng thứ nhất của A :

```
>> x = A(1,1:3) % truy cập phần tử thứ 1 đến 3 của hàng 1 ma trận A
```

```
x =
```

```
2  4  1
```

- b. Tạo ma trận y là hai hàng còn lại (cuối) của A :

```
>> y = A(2:3,1:3) % truy cập phần tử thứ 1 đến 3 của hàng 2 và hàng 3  
% của ma trận A
```

```
y =
```

```
6  7  2
3  5  9
```





c. Tính tổng theo hàng ma trận A:

```
>> (sum(A'))'
```

```
ans =
```

```
7
```

```
15
```

```
17
```

d. Tính tổng theo cột ma trận :

```
>> sum(A)
```

```
ans =
```

```
11 16 12
```

2. Cho ma trận $C = \begin{bmatrix} 6 & 9 & 5 & 1 \\ 8 & 7 & 2 & 3 \\ 1 & 3 & 4 & 4 \\ 5 & 2 & 8 & 2 \end{bmatrix}$ và $D = \begin{bmatrix} 4 & 8 \\ 3 & 7 \\ 2 & 3 \\ 5 & 1 \end{bmatrix}$, dùng Matlab:

```
>> C = [6 9 5 1; 8 7 2 3; 1 3 4 4; 5 2 8 2] % gán các giá trị của ma trận  
% 4x4 cho biến C theo nội dung.
```

```
C =
```

```
6 9 5 1
```

```
8 7 2 3
```

```
1 3 4 4
```

```
5 2 8 2
```

```
>> D = [4 8 ; 3 7 ; 2 3 ; 5 1] % gán các giá trị của ma trận  
% 2x4 cho biến D theo nội dung.
```

```
D =
```

```
4 8
```

```
3 7
```

```
2 3
```

```
5 1
```





a. Tạo ma trận E1 là 2 cột nằm giữa của ma trận C sử dụng toán tử ':'

```
>> E1 = C(1:4,2:3) % truy cập phần tử thứ 2 đến 3 của  
% hàng 1 đến 4 của ma trận C
```

E1 =

```
9  5  
7  2  
3  4  
2  8
```

b. Tạo ma trận E2 từ hàng 1 và 2 và cột 2 và 3 của ma trận C sử dụng toán tử ':'

```
>> E2 = C(1:2,2:3)
```

E2 =

```
9  5  
7  2
```

c. Tạo ma trận E3 bằng cách ghép 2 ma trận E1 và D với nhau:

```
>> E3 = [E1,D]
```

E3 =

```
9  5  4  8  
7  2  3  7  
3  4  2  3  
2  8  5  1
```

VII. Thực hành các phép toán mảng trên cửa sổ lệnh.

- + : công ma trận.
- - : trừ ma trận.
- * : nhân ma trận.
- / : chia phải ma trận.
- \ : chia trái ma trận.
- ^ : toán tử lũy thừa ma trận.
- .* : toán tử nhân hai phần tử tương ứng của hai ma trận.
- ./ : chia phải hai phần tử tương ứng của hai ma trận.
- .\ : chia trái hai phần tử tương ứng của hai ma trận.





➤ \wedge : toán tử lũy thừa từng mỗi phần tử của ma trận.

Cho $x = [1 \ 4 \ 8]$, $y = [2 \ 1 \ 5]$ và $A = [3 \ 1 \ 6 ; 5 \ 2 \ 7]$

```
>> x = [ 1 4 8 ]
```

```
x =  
    1    4    8
```

```
>> y = [ 2 1 5 ]
```

```
y =  
    2    1    5
```

```
>> A = [ 3 1 6 ; 5 2 7 ]
```

```
A =  
    3    1    6  
    5    2    7
```

Thực hiện các phép tính sau:

a. $x*2 + y*3$:

```
>> x*2+y*3
```

```
ans =  
    8   11   31
```

b. $x - y.^3$:

```
>> x - y.^3
```

```
ans =  
   -7    3  -117
```

c. $A*3 / x$

```
>> A*3/x
```

```
ans =  
    2.0370  
    2.5556
```

d. $A.^2$

```
>> A.^2
```

```
ans =  
    9    1   36  
   25    4   49
```

=====Hết Bài Tập Thực Hành Số 1=====



BÀI TẬP THỰC HÀNH SỐ 2

Thực hành lập trình với m-file

Thực hành các lệnh trên cửa sổ soạn thảo m-file.

File-New : tạo tập tin mới.

File-Open: mở tập tin đã có sẵn.

File-Save: lưu tập tin lệnh đã.

File-Save as lưu tập tin lên đĩa với một tên khác.

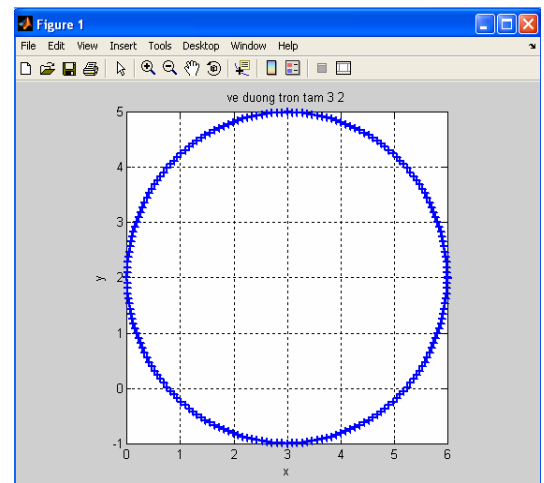
I. Lập trình với m-file.

Sử dụng vòng lặp for, vòng lặp while, lệnh rẽ nhánh if, if-else, if-elseif-else, lệnh kiểm tra trường hợp case, lệnh break...

1. Lập trình với Script m-file.

a. Chương trình vẽ đường tròn tâm (3,2) và bán kính 3.

```
% ve duong tron tam 3,2
t = 0:pi/100:2*pi;
x = 3 + 3*cos(t);
y = 2 + 3*sin(t);
plot(x,y,'b','linewidth',1.5,'linestyle','+'),grid
xlabel('x')
ylabel('y')
title('ve duong tron tam 3 2')
% End of program .
% lệnh 'linewidth' : dùng cài đặt độ rộng nét vẽ.
% lệnh 'linestyle' cài đặt kiểu của nét vẽ.
```



Save lại với tên là Duongtron.m

Nhấn F5 để run hay trong cửa sổ command window gõ lệnh:

```
>>Duongtron
```

Cho kết quả là hình bên.

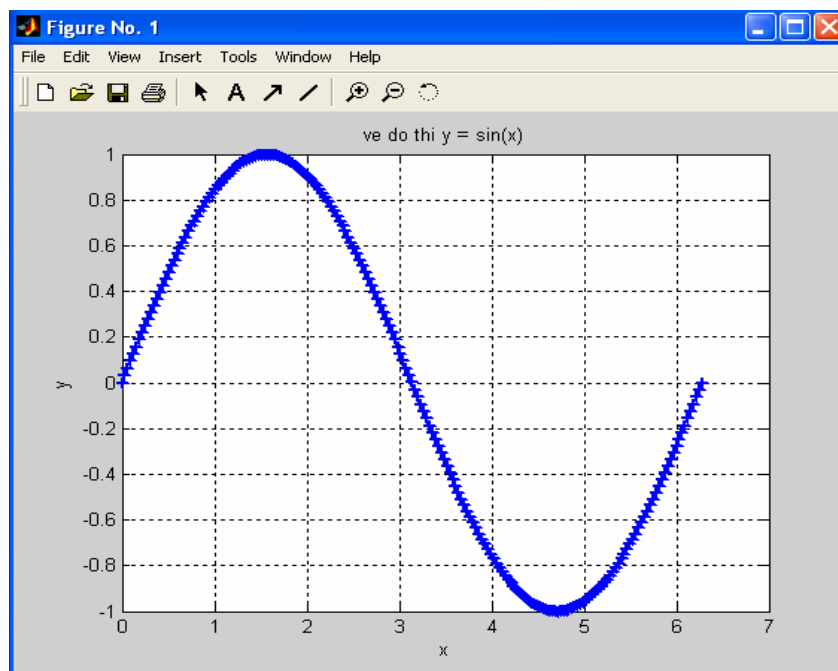
b. Sinh viên tự viết 1 Script m-file có nhiệm vụ vẽ 1 hình Sin

```
% Begin of program
x = 0:pi/100:2*pi ;
y = sin(x) ;
plot(x,y,'b','linewidth',1.5,'linestyle','+'),grid
xlabel('x')
ylabel('y')
title('ve do thi y = sin(x)')
% End of program
```

Hay ta có thể gõ lần lượt các lệnh sau trước dấu nhắc của cửa sổ commandWindow

```
>> x = 0:pi/100:4*pi;  
>> y = sin(x);  
>> plot(x,y,'b');  
>> grid on  
>> xlabel('X')  
>> ylabel('Y')  
>> title(' đồ thị y = sin(x)')
```

Chạy chương trình cho ta kết quả như hình :



2. Lập trình với function m-file.

a. Chương trình giải pt bậc 2.

```
% Function m-file  
function [x1, x2] = quadratic(a,b,c)  
x = roots([a b c]);  
x1 = x(1)  
x2 = x(2)  
%End
```

Save lại với tên là quadratic.m , trong cửa sổ command window gõ lệnh

```
>> quadratic ( 1 ,2 ,-3)           % giai pt: x^2 + 2*x -3 = 0
```

```
x1 =
```



-3

x2 =
1

b. Sinh viên tự viết 1 *function m-file* giải phương trình bậc 2 mà không dùng hàm roots mà matlab đã hỗ trợ.

% chương trình giải phương trình bậc 2 không dùng hàm roots

a = input('nhập hệ số a :');

b = input('nhập hệ số b :');

c = input('nhập hệ số c :');

radical = sqrt(b^2 - 4*a*c);

x1 = (-b+radical)/(2*a)

x2 = (-b-radical)/(2*a)

% End of program

Chạy chương trình cho kết quả sau :

% giải pt: $a*x^2 + b*x + c = 0$

nhập hệ số a :1

nhập hệ số b :2

nhập hệ số c :-3

x1 =
1

x2 =
-3

II. Thực hành với các lệnh nhập xuất dữ liệu input và disp trong Script và function m-file.

a. Thực hiện lại chương trình giải pt bậc 2 với các hệ số a, b, c nhập từ bàn phím.

% Script m-file

a = input('nhập hệ số a:');

b = input('nhập hệ số b:');

c = input('nhập hệ số c:');

disp(' nghiệm của pt bậc 2 là :')

x = roots([a b c]);

x1 = x(1)

x2 = x(2)

% End of program

Save lại với tên là quadratic2.m , trong cửa sổ command window gõ lệnh:





```
>> quadratic2
nhap he so a:1
nhap he so b:3
nhap he so c:-4
  nghiem cua pt bac 2 la :
x1 =
    - 4
x2 =
     1
```

b. Sinh viên tự thực hiện chương trình vẽ đường cong parabol $y = ax^2 + b$ với các hệ số a , b , nhập từ keyboard:

Cách vẽ 1:

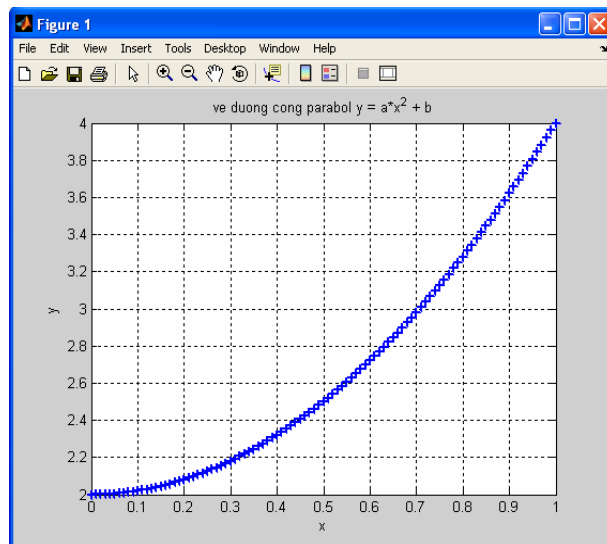
```
% Begin of program
a = input ('nhap he so a:'); % nhập hệ số a từ bàn phím
b = input ('nhap he so b:'); % nhập hệ số b từ bàn phím
x = 0:0.01:1;
p = [ a 0 b]; % tạo ra đa thức: y = a*x^2 + 0*x + b = a*x^2 + b
y = polyval(p,x); % xác định y ứng với các giá trị của x.
plot(x,y,'b','linewidth',1.5,'linestyle','+'),grid
xlabel('x')
ylabel('y')
title('ve duong cong parabol y = a*x^2 + b')
% End of program.
```

Cách vẽ 2:

```
% Begin of program
a = input ('nhap he so a:'); % nhập hệ số a từ bàn phím
b = input ('nhap he so b:'); % nhập hệ số b từ bàn phím
x = 0:0.01:1;
y = a*x.^2 + b
plot(x,y,'b','linewidth',1.5,'linestyle','+'),grid
xlabel('x')
ylabel('y')
title('ve duong cong parabol y = a*x^2 + b')
% End of program
```

Chạy chương trình trên cho ta kết quả như hình dưới đây:





c. Sinh viên tự thực hiện chương trình tính trung bình các số với số lượng các số (ít hơn 10 số) và giá trị các số được nhập từ bàn phím và xuất giá trị trung bình cuối cùng ra màn hình.

```
% Chương trình tính trung bình :
n = input('Enter n :'); % nhập số lượng các số cần tính trung bình
i = 1; % gán số 1 cho biến i
s = 0; % gán số 0 cho biến s
while i <= n; % dùng vòng lặp while, nếu i <= n cho phép vòng lặp mới
    x = input('Enter x :'); % nhập giá trị x thứ nhất
    s = s + x; % s = s cũ + với giá trị x mới nhập
    i = i + 1; % tăng giá trị biến i lên 1 đơn vị
end % từ khóa kết thúc vòng lặp while
Tb = s/n % xuất giá trị trung bình cuối cùng ra màn hình command window
% end of program
```

Chạy chương trình trên cho ta kết quả sau :

```
Enter n :5
Enter x :3
Enter x :4
Enter x :5
Enter x :6
Enter x :7
Tb =
    5
```



III. Thực hành hàm chính, hàm con .

Tạo 1 hàm trong đó có nhiều hàm con hàm đầu tiên là hàm chính và phải có tên cùng tên với m-file, còn các hàm con phải có tên khác với m-file.

a. Thực hiện đoạn code sau :

```
% Function m-file
function [z,s] = main(x,y)
    z = total(x,y)
    s = subtract(x,y)
function z = total(x,y)
    z = x + y;
function s = subtract(x,y)
    s = x - y;
%end of function
```

Save lại với tên là main.m , trong cửa sổ command window gõ lệnh

```
>> main(9,4)
```

```
z =
    13
```

```
s =
     5
```

```
>> main(9,6)
```

```
z =
    15
```

```
s =
     3
```

b. Sinh viên tự tạo ra 1 Script m-file cho phép nhập chiều dài và chiều rộng của 1 hình chữ nhật từ bàn phím và xuất giá trị độ dài đường chéo ra màn hình (phải sử dụng hàm con trong chương trình).

Chương trình thực hiện yêu cầu :

```
% Function m-file
function [c] = chunhat(a,b)
    c = total(a,b)
function c = total(a,b)
    c = sqrt( a.^2 + b.^2)
%End of function
```





Chạy chương trình và cho ta kết quả sau:

```
>> chunhat(3,4)
c =
    5
```

Cách khác không dùng hàm con:

```
% Begin of program
a = input('nhap gia tri a :');
b = input('nhap gia tri b :');
c = sqrt(a.^2 + b.^2)
% End of program.
```

Ví dụ :

```
nhap gia tri a :4
nhap gia tri b :3
```

```
c =
    5
```

IV. Thực hành các hàm xử lý chuỗi.

Các hàm matlab hỗ trợ.

double: đổi chuỗi sang mã ASCII.
char: tạo mảng ký tự từ mảng số.
strcmp: so sánh chuỗi.
strcat: nối chuỗi.
num2str: đổi số sang chuỗi.
str2num: đổi chuỗi sang số.
int2str: đổi số nguyên sang chuỗi.

a. Thực hiện các đoạn code sau.

```
>> name = 'Truong Dai Hoc DL Cong Nghe Sai Gon'
name =
Truong Dai Hoc DL Cong Nghe Sai Gon
>> name(8:35) % truy cập chuỗi từ ký tự 8 đến ký tự 35.
ans =
Dai Hoc DL Cong Nghe Sai Gon
```



**Tiếp tục thực hiện:**

```
>> text1 = 'toi hoc tai'      % gán chuỗi toi hoc tai cho text1.  
>> text = [text1 ' ' name]    % kết hợp 2 chuỗi
```

```
>> text1 = 'Toi hoc tai'  
text1 =  
Toi hoc tai  
>> text = [text1 ' ' name]  
text =  
Toi hoc tai Truong Dai Hoc DL Cong Nghe Sai Gon
```

b. Thực hiện các đoạn code sau

```
>> A = 'Dai Hoc Sai Gon'  
A =  
Dai Hoc Sai Gon  
  
>> B = double(A)  
B =  
Columns 1 through 12  
  
68  97  105  32  72  111  99  32  83  97  105  32  
  
Columns 13 through 15  
  
71  111  110  
  
>> C = char(B)  
C =  
Dai Hoc Sai Gon
```

c. thực hiện các đoạn code và cho kết quả sau :

```
>> Tg = 2.2774;  
>> xg = 144.6364;  
>> disp(['time of flight:' num2str(Tg) 's'])  
time of flight:2.2774s  
>> disp(['distance traveled:' num2str(xg) 'ft'])  
distance traveled:144.6364ft
```

V. Thực hành mảng cấu trúc trên cửa sổ lệnh.

**- Tạo mảng cấu trúc.**

```
>> SinhVien(1).hovaten = 'Nguyen Thanh Liem';  
>> SinhVien(1).diemtoan = 6;  
>> SinhVien(1).diemly = 7;  
>> SinhVien(2).hovaten = 'Nguyen Hoang Anh';  
>> SinhVien(2).diemtoan = 7;  
>> SinhVien(2).diemly = 8;
```

- Truy cập mảng cấu trúc.

```
>> SinhVien(1) % truy cập thông tin về thành viên thứ 1 của mảng.  
ans =  
    hovaten: 'Nguyen Thanh Liem'  
    diemtoan: 6  
    diemly: 7  
  
>> SinhVien(2) % truy cập thông tin về thành viên thứ 2 của mảng.  
ans =  
    hovaten: 'Nguyen Hoang Anh'  
    diemtoan: 7  
    diemly: 8  
  
>> SinhVien(1).diemtoan % xem điểm toán của thành viên 1.  
ans =  
    6  
  
>> SinhVien(2).hovaten % xem họ và tên của thành viên 2.  
ans =  
    Nguyen Hoang Anh
```

=====Hết Bài Tập Thực Hành Số 2=====



BÀI TẬP THỰC HÀNH SỐ 3

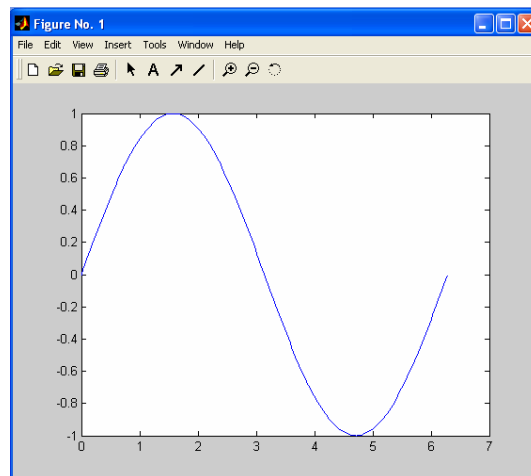
Xử lý đồ họa

I. Xử lý đồ họa hai chiều.

1. Hàm vẽ đồ thị hai chiều `plot(x,y, s)`, trong đó `y` là vector điểm vẽ theo vector điểm `x`, `s` là chuỗi ký tự ký hiệu thiết lập màu tô và nét vẽ. (dùng lệnh `>>help plot` để biết thêm cấu trúc hàm `plot`).

```
>> x = 0:pi/100:2*pi;  
>> y = sin(x);  
>> plot(x,y)
```

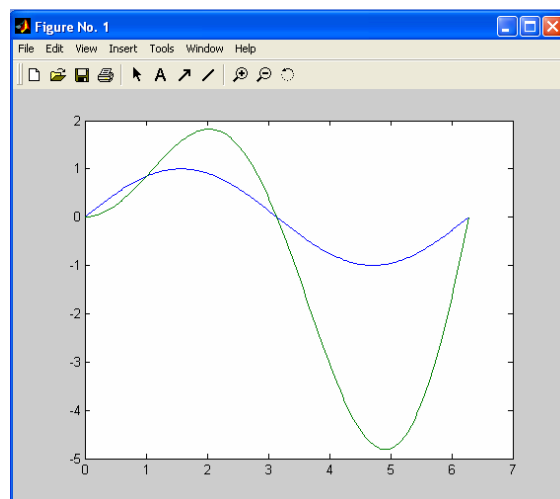
Kiểm tra và lưu kết quả.



Để vẽ thêm 1 hàm `x.sin(x)` lên cùng 1 đồ thị ta dùng thêm các lệnh sau.

```
>> z = x.*sin(x);  
>> plot(x,y,x,z)
```

Kiểm tra và lưu kết quả.



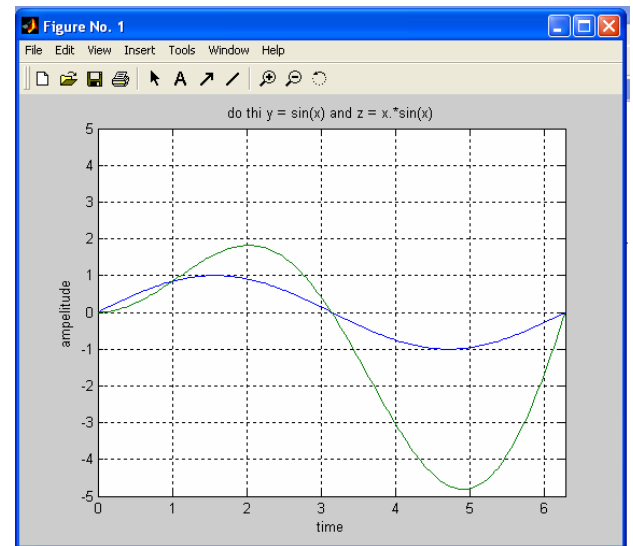
2.

- Hàm xlabel tạo nhãn trên trục tọa độ x và hàm ylabel tạo nhãn trên trục tọa độ y, grid tạo mảng kẻ lưới, hold on/off giữ lại đồ thị trước đó.
- Hàm title tạo tiêu đề đồ thị và hàm legend tạo nhãn các đồ thị tương ứng trên cửa sổ hình.
- Hàm axis([Xmin Xmax Ymin Ymax]) thiết lập tọa độ nhỏ nhất và lớn nhất trên trục tọa độ X-Y.

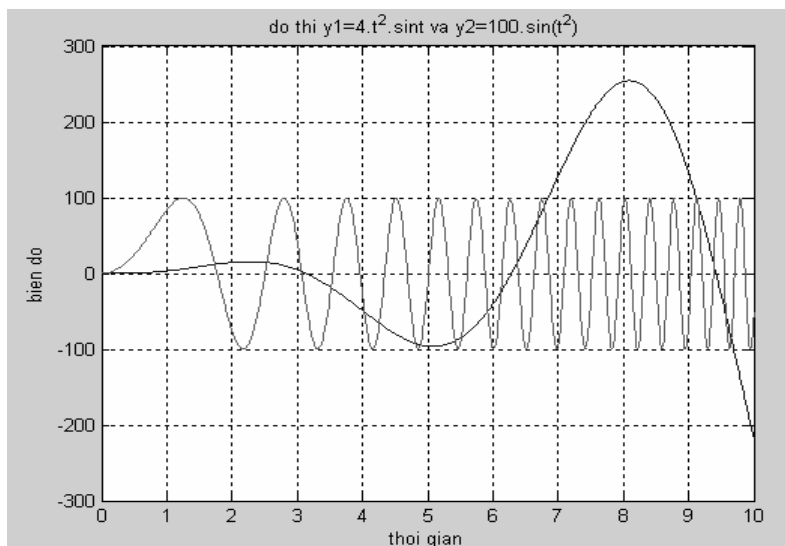
Tiếp theo phần 1 ta thực hiện tiếp

```
>> legend('sin(x)', 'x.sin(x)')
>> xlabel('time')
>> ylabel('amplitude')
>> title('do thi y=sin(x) and z=x.sin(x)')
>> axis([0 2*pi -5 5])
>> grid on % tạo mảng kẻ lưới
```

Kiểm tra và lưu kết quả.

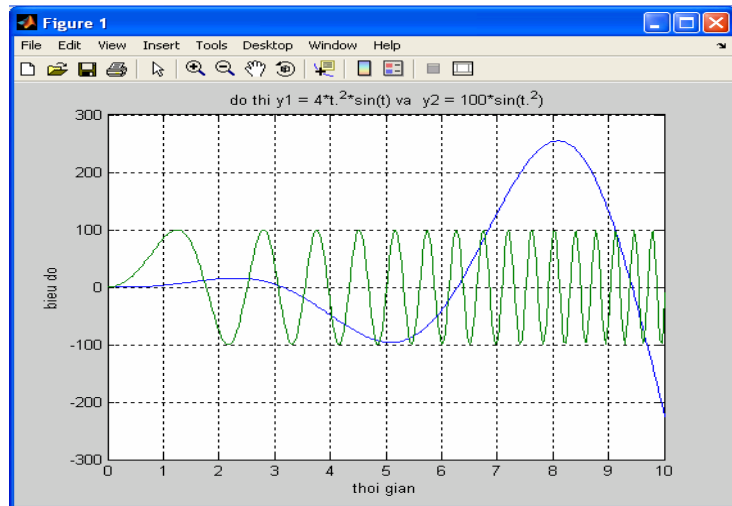


Sinh viên tự thực hiện đồ thị sau.



chương trình thực hiện yêu cầu trên :

```
% Begin of program
t = 0:pi/100:4*pi;
y1 = 4*t.^2.*sin(t);
y2 = 100*sin(t.^2);
plot(t,y1,t,y2);
Xlabel('thời gian');
ylabel('biểu đồ');
axis([0 10 -300 300]);grid
title(' do thi y1 = 4*t.^2.*sin(t) va 100*sin(t.^2)')
% End of program
```



Chương trình 2 :

```
% Begin of program
t = 0:pi/100:4*pi;
y1 = 4*(t.^2).*sin(t);
y2 = 100*sin(t.^2);
plot(t,y1,t,y2);
Xlabel('thời gian');
ylabel('biểu đồ');
axis([0 10 -300 300]);grid ;
title(' do thi y1 = 4*t.^2.*sin(t) va 100*sin(t.^2)');
% End of program
```

Hay ta có thể gõ trực tiếp các lệnh trong cửa sổ CommandWindow như sau:

```
>> t = 0:pi/100:4*pi;
>> y1 = 4*t.^2.*sin(t);
>> y2 = 100*sin(t.^2);
>> plot(t,y1,t,y2);
>> Xlabel('thời gian');
>> ylabel('biểu đồ');
>> axis([0 10 -300 300]);
>> grid on
```



```
>> title(' đồ thị  $y_1 = 4*t.^2.*\sin(t)$  và  $100*\sin(t.^2)$  ');
```

3. Hàm subplot(m,n,p) chia khung hình figure ra nhiều khung con, trong đó mỗi khung chứa một đồ thị với m là số khung theo dòng, n là số khung theo cột, p là số thứ tự của các khung được thiết lập.

- Chia khung hình figure ra làm 2 mỗi khung chứa 1 đồ thị mà ta đã thực hiện ở phần trên 1 và 2.

Gợi ý:

Trong quá trình vẽ đồ thị $y=\sin(x)$ và $z=x.\sin(x)$ thay thế dòng lệnh $plot(x,y,x,z)$ thành $subplot(2,1,1), plot(x,y,x,z)$ khi đó đồ thị sẽ vẽ ở nửa trên khung hình.

Kiểm tra và lưu kết quả.

```
%Beginning of program
```

```
x = 0:pi/100:2*pi;
```

```
y = sin(x);
```

```
z = x.*sin(x);
```

```
subplot(2,1,1); % lệnh cho phet tạo ra nhưng khung chưa
```

```
plot(x,y,x,z);
```

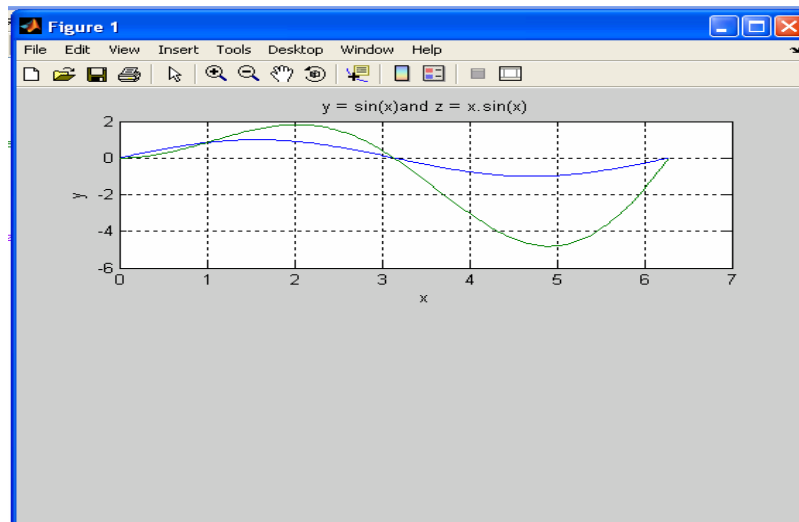
```
grid;
```

```
xlabel('x');
```

```
ylabel('y');
```

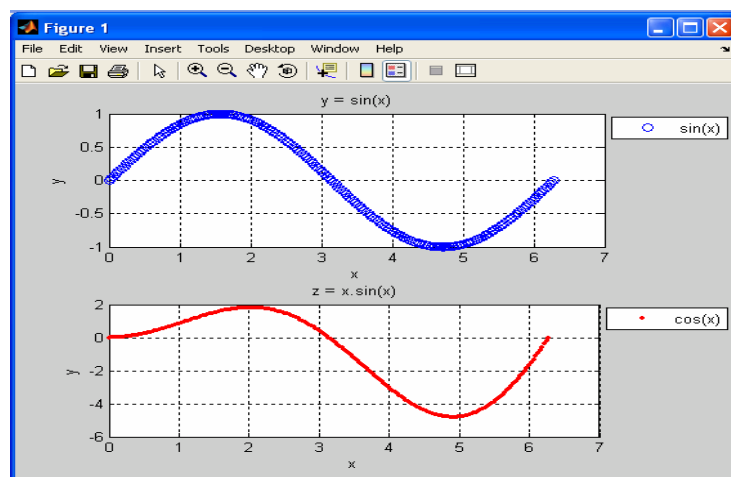
```
title('y = sin(x) and z = x.sin(x)');
```

```
% End of program
```



Chương trình vẽ 2 đồ thị $y = \sin(x)$ và $z = x \cdot \sin(x)$ trên 2 khung trên cùng một cửa sổ Figure:

```
%Beginning of program
x = 0:pi/100:2*pi;
y = sin(x);
z = x.*sin(x);
subplot(2,1,1) % lệnh cho phet tao ra nhung khung chua
plot(x,y,'linewidth',1.0,'linestyle','o','color','b')
grid
xlabel('x')
ylabel('y')
title('y = sin(x)')
legend('sin(x)',-1) % lech cho phet hien thi ten va mau do thi ve
subplot(2,1,2) % leanh cho pheut tao ra nhöông khung chöùa
plot(x,z,'linewidth',1.0,'linestyle','-', 'color','r')
grid % lệnh tạo mạng lưới
xlabel('x') % lệnh xlabel(x) : gian nhan truc x
ylabel('y')
title('z = x.sin(x)') % title( y = sin(x)): gian nhan cho do thi.
legend('cos(x)',-1) % lech cho phet hien thi ten va mau do thi ve
% End of program
```

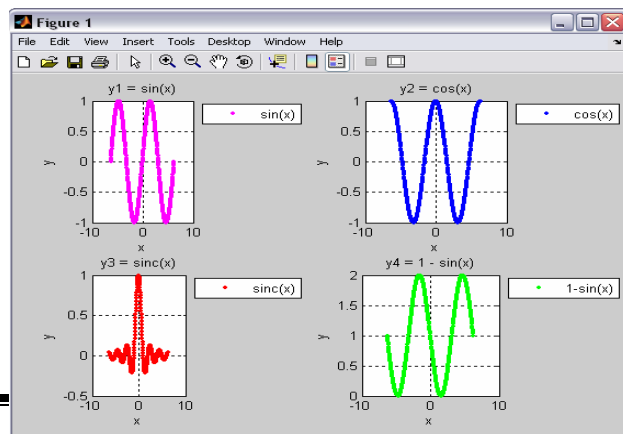


Chương trình vẽ 4 đồ thị trên 4 khung trên cùng một cửa sổ Figure:

```

%Beginning of program
x = -2*pi:pi/100:2*pi;
y1 = sin(x);
y2 = cos(x);
y3 = sinc(x);
y4 = 1 - sin(x);
subplot(2,2,1)
plot(x,y1,'linewidth',1.0,'linestyle','-', 'color','m')
grid
xlabel('x')
ylabel('y')
title('y1 = sin(x)')
legend('sin(x)',-1)
subplot(2,2,2)
plot(x,y2,'linewidth',1.0,'linestyle','-', 'color','b')
grid
xlabel('x')
ylabel('y')
title('y2 = cos(x)')
legend('cos(x)',-1)
subplot(2,2,3)
plot(x,y3,'linewidth',1.0,'linestyle','-', 'color','r')
grid
xlabel('x')
ylabel('y')
title('y3 = sinc(x)')
legend('sinc(x)',-1)
subplot(2,2,4)
plot(x,y4,'linewidth',1.0,'linestyle','-', 'color','g')
grid
xlabel('x')
ylabel('y')
title('y4 = 1 - sin(x)')
legend('1-sin(x)',-1)
% End of program

```





4. Tạo nét vẽ khác nhau dùng, màu vẽ khác nhau.

Dùng lệnh `>>help plot` để biết thêm cấu trúc hàm plot.

Màu được xác định bằng các tham số:

Mã	Màu	Mã	Màu
r	red	m	magenta
g	green	y	yellow
b	blue	k	black
c	cyan	w	white

Các dạng điểm đánh dấu xác định bằng:

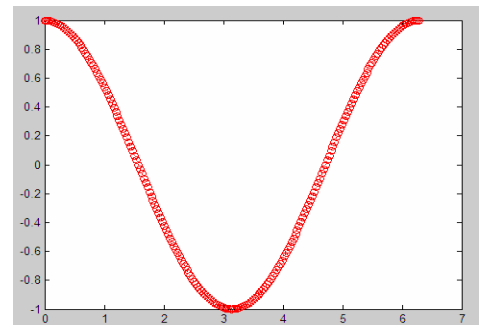
Mã	Kiểu đánh dấu	Mã	Kiểu đánh dấu
+	dấu cộng	.	điểm
o	vòng tròn	x	chữ thập
*	dấu sao	s	hình vuông
d	hạt kim cương	v	điểm tam giác hướng xuống
^	điểm tam giác hướng lên	<	tam giác sang trái
>	tam giác sang phải	h	lục giác
p	ngũ giác		

Các dạng đường thẳng xác định bằng:

Mã	Kiểu đường
-	đường liền
--	đường đứt nét
:	đường chấm chấm
-.	đường chấm gạch

Thực hiện lại đoạn code sau và lưu kết quả.

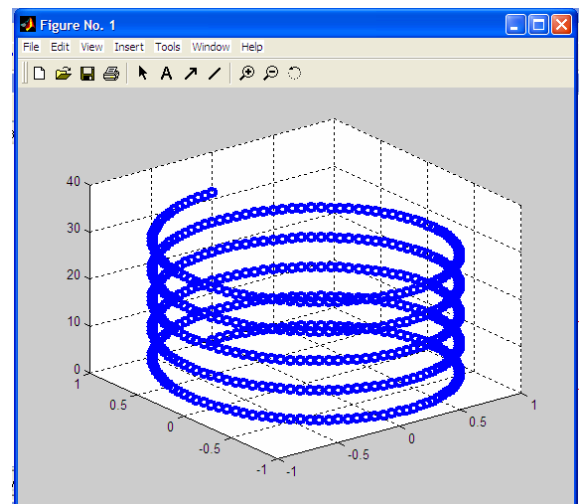
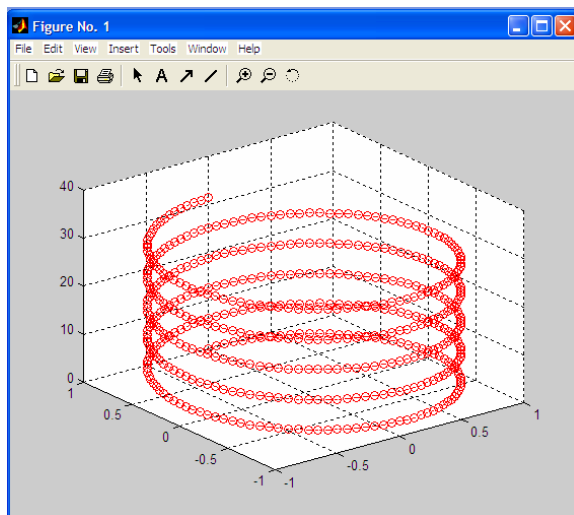
```
>> x = 0:pi/100:2*pi;
>> y = cos(x);
>> plot(x,y,'ro:')
```



II. Xử lý đồ họa ba chiều.

1. Hàm `plot3(x,y,z,s)`: dùng để vẽ các đường trong không gian 3 chiều, trong đó `x,y,z` là các vector điểm, `s` là chuỗi ký hiệu cho màu và nét vẽ. Thực hiện lại đoạn code sau và lưu kết quả.

```
>> t = 0:pi/50:10*pi;
>> x = sin(t);
>> y = cos(t);
>> z = t;
>> plot3(x,y,z,'ro:')
>> grid on
% có thể ứng lệnh sau:
>> plot3(x,y,z,'Linewidth',3.0,'LineStyle','o','color','b')
>> grid on
```



2. Hàm `[x,y] = meshgrid(x,y)` tạo hai vector điểm `x, y` kết hợp vẽ đồ thị ba chiều với hàm `surf(x,y,z,c)` và `mesh(x,y,z,c)`. Dùng để vẽ các vật thể 3D trong không gian 3 chiều.

Meshgrid: định lưới vẽ.

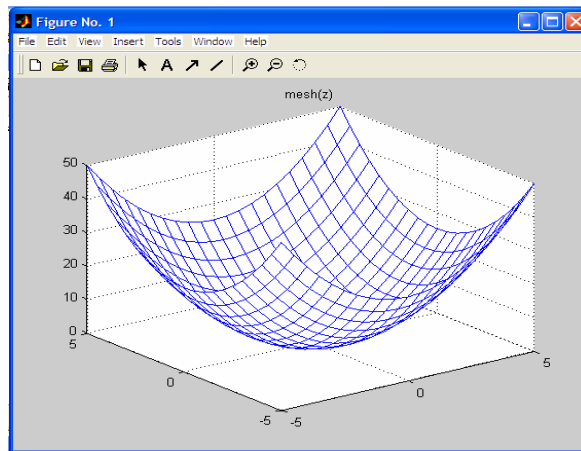
Mesh: vẽ mặt lưới 3D.

Vẽ mặt paraboloid $z = x^2 + y^2$ trong không gian 3 chiều.

Thực hiện lại đoạn code sau và lưu kết quả.

```
>> close all
>> t = -5:0.5:5; % cho t chạy từ -5 đến 5 với bước nhảy 0.5
>> [x,y] = meshgrid(t); % định lưới vẽ.
>> z = x.^2 + y.^2;
```

```
>> mesh(x,y,z,'Edgecolor','b'),title('mesh(z)') %ve mat luoi 3D
```



Sinh viên tự vẽ mặt $z = \frac{\sin(\sqrt{x^2 + y^2})}{\sqrt{x^2 + y^2}}$ trong không gian 3D và lưu kết quả.

Cách 1: Thực hiện chương trình trong m-file:

```
% ve khong gian 3 D
```

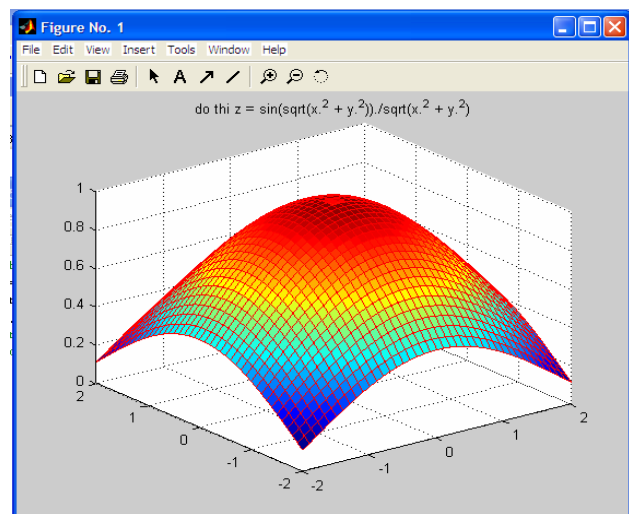
```
[x,y] = meshgrid(-2:0.1:2);
```

```
z = sin(sqrt(x.^2 + y.^2))./sqrt(x.^2 + y.^2);
```

```
surf(x,y,z,'Edgecolor','r'),title('do thi z = sin(sqrt(x.^2 + y.^2))./sqrt(x.^2 + y.^2)');
```

```
% End of program
```

Chạy chương trình cho ta kết quả :



Cách 2: đánh trực tiếp các lệnh trong cửa sổ Command Window:

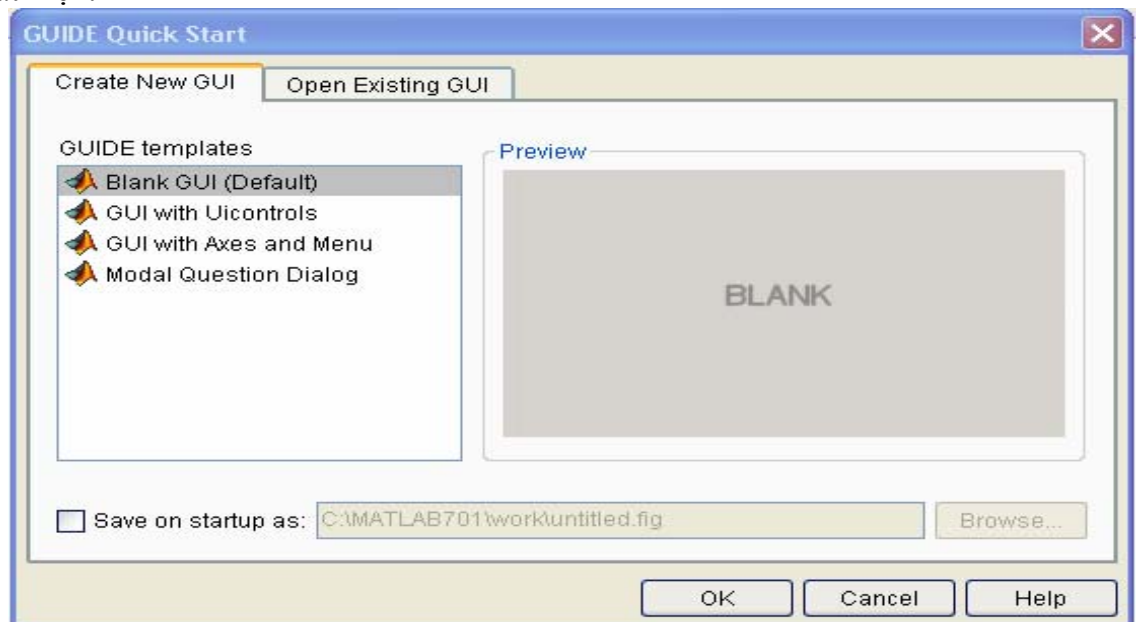
```
>> [x,y] = meshgrid(-2:0.1:2);  
>> z = sin(sqrt(x.^2 + y.^2))./sqrt(x.^2 + y.^2);  
>> surf(x,y,z,'Edgecolor','r'),title('đồ thị z = sin(sqrt(x.^2 + y.^2))./sqrt(x.^2 + y.^2)')
```

III. Thực hành tạo cửa sổ giao diện.

Tạo giao diện người sử dụng sử dụng công cụ Guide.

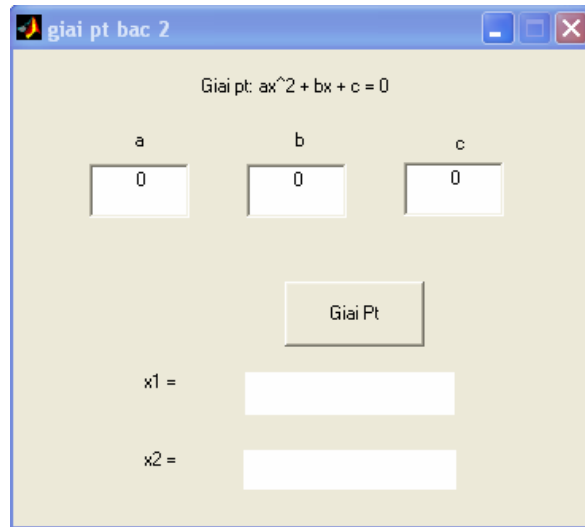
Sử dụng công cụ Guide mà matlab hỗ trợ để lập trình tạo giao diện.

Gõ lệnh guide trước dấu nhắc của cửa sổ lệnh, màn hình chính của guide xuất hiện.

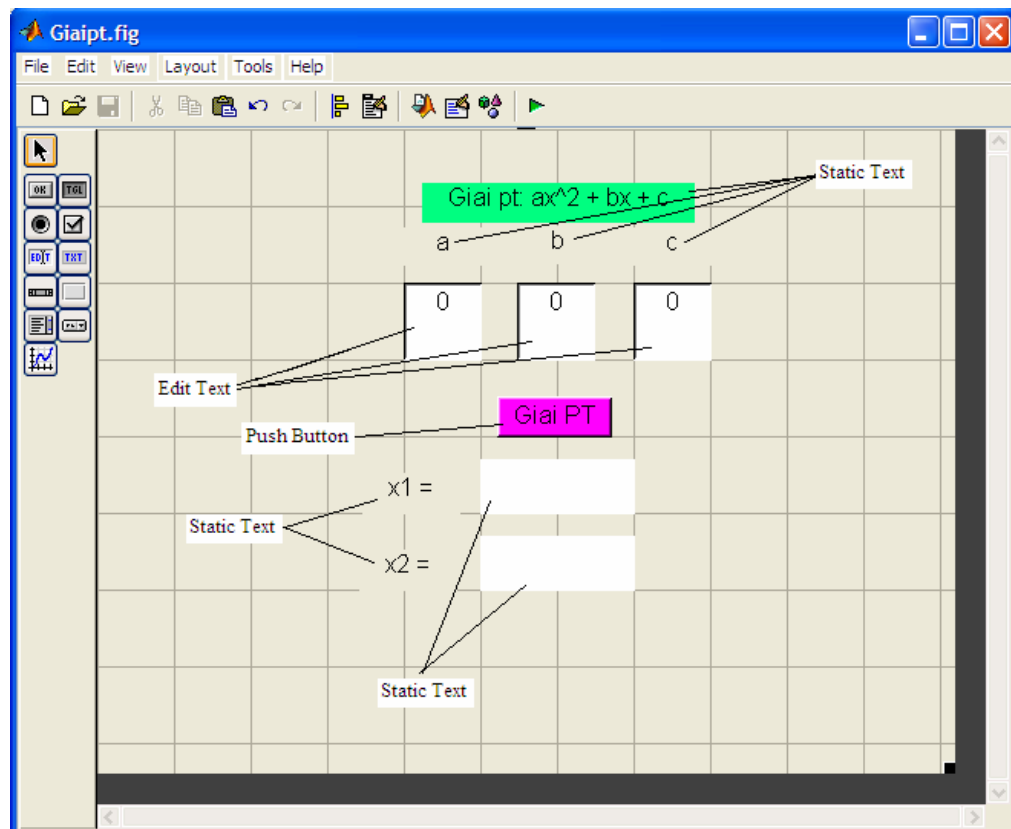


1. Thực hiện tạo giao diện giải phương trình bậc 2 $ax^2 + bx + c = 0$, với các hệ số a, b, c có thể thay đổi.

Với giao diện đề nghị như sau:



Trước tiên ta tạo giao diện guide như hình dõuôi nầy:



Để có được hình như trên ta cần tạo các đối tượng sau:

Edit Text (số lượng 3)				
STT	Fontsize	String	Tag	Color



1	12	0	Hesoa	White
2	12	0	Hesob	White
3	12	0	Hesoc	White

Static Text (số lượng 8)				
STT	Fontsize	String	Tag	color
1	12	Giai pt: $ax^2 + bx + c$	Text1	Tùy ý
2	12	a	Text2	Tùy ý
3	12	b	Text3	Tùy ý
4	12	c	Text4	Tùy ý
5	12	Bỏ trống	Ketquax1	White
6	12	Bỏ trống	Ketquax2	white
7	12	X1 =	Text7	Tùy ý
8	12	X2 =	Text8	Tùy ý

Push Button (số lượng 1)				
STT	Fontsize	String	Tag	Color
1	12	Giai PT	Giai_pt	Tùy ý

Sau khi tạo xong các đối tượng trên, ta nên save nó với tên (Giaipt.fig), rồi tiếp tục Chọn view → M-file editor ở trên cửa sổ của giao diện guide vừa tạo xong, khi đó sẽ xuất hiện một cửa sổ của M-file có chương trình của giao diện guide mà ta vừa tạo, ta soạn thảo thêm đoạn chương trình dưới vào cuối chương trình của m-file đó.

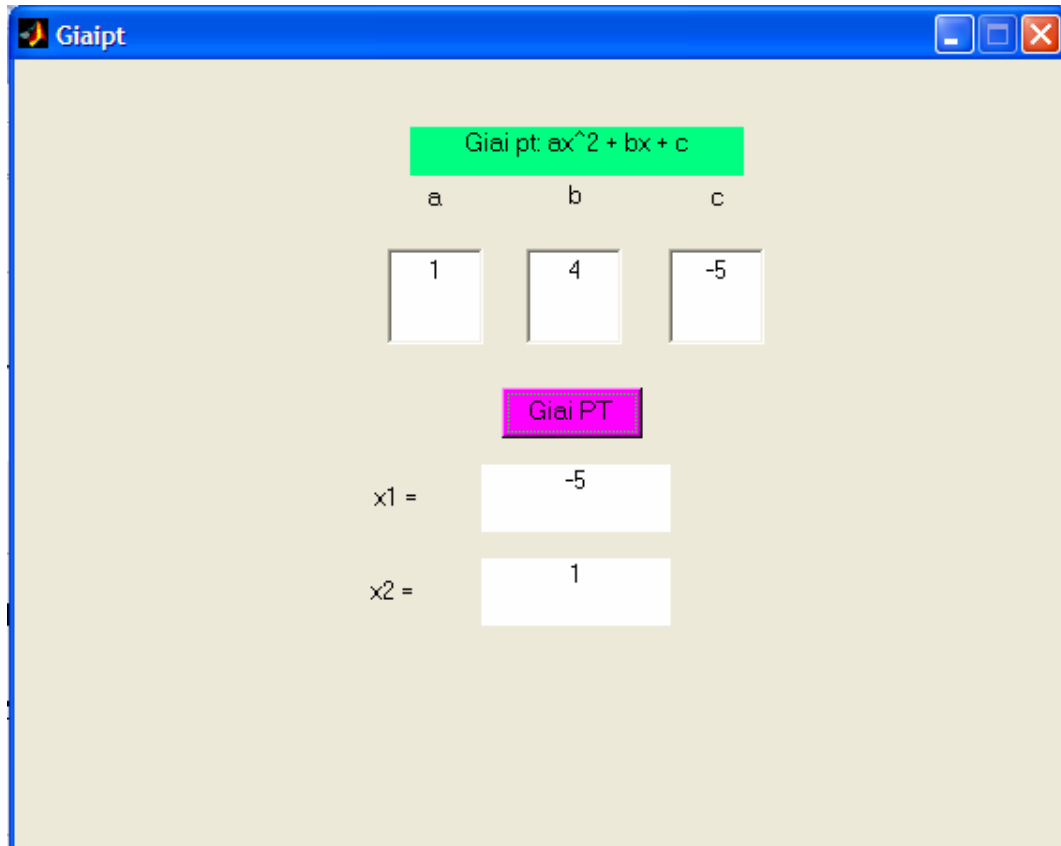
% Chương trình giải phương trình bậc 2 cần thêm vào :

```
function Giai_pt_Callback(hObject, eventdata, handles)
%code xử lý giải phương trình khi ta ấn nút Giai pt
hesoa= str2double(get(handles.hesoa,'string'))
hesob= str2double(get(handles.hesob,'string'))
hesoc= str2double(get(handles.hesoc,'string'))
x = roots([hesoa hesob hesoc]);
set(handles.ketquax1,'string',x(1))
set(handles.ketquax2,'string',x(2))
```



% --- Executes during object creation, after setting all properties.

Sau khi soạn xong đoạn chương trình đó, ta save và chạy chương trình và nhập các số a b c vào, tiếp tục nhấn vào nút Giải PT cho ta kết quả như hình dưới :



Chương trình trong M-file của giao diện:

%Begin of program

function varargout = giaiptbac2(varargin)

gui_Singleton = 1;

gui_State = struct('gui_Name', mfilename, ...

'gui_Singleton', gui_Singleton, ...

'gui_OpeningFcn', @giaiptbac2_OpeningFcn, ...

'gui_OutputFcn', @giaiptbac2_OutputFcn, ...

'gui_LayoutFcn', [], ...

'gui_Callback', []);

if nargin & isstr(varargin{1})

gui_State.gui_Callback = str2func(varargin{1});

end



```
if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function giaiptbac2_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
function varargout = giaiptbac2_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
function hesoa_CreateFcn(hObject, eventdata, handles)
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function hesoa_Callback(hObject, eventdata, handles)
function hesob_CreateFcn(hObject, eventdata, handles)
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function hesob_Callback(hObject, eventdata, handles)
function hesoc_CreateFcn(hObject, eventdata, handles)
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function hesoc_Callback(hObject, eventdata, handles)
% đoạn chương trình giải Phương trình bậc 2:
function Giai_pt_Callback(hObject, eventdata, handles)
hesoa= str2double(get(handles.hesoa,'string'))
hesob= str2double(get(handles.hesob,'string'))
hesoc= str2double(get(handles.hesoc,'string'))
x = roots([hesoa hesob hesoc]);
set(handles.ketquax1,'string',x(1))
set(handles.ketquax2,'string',x(2))
```

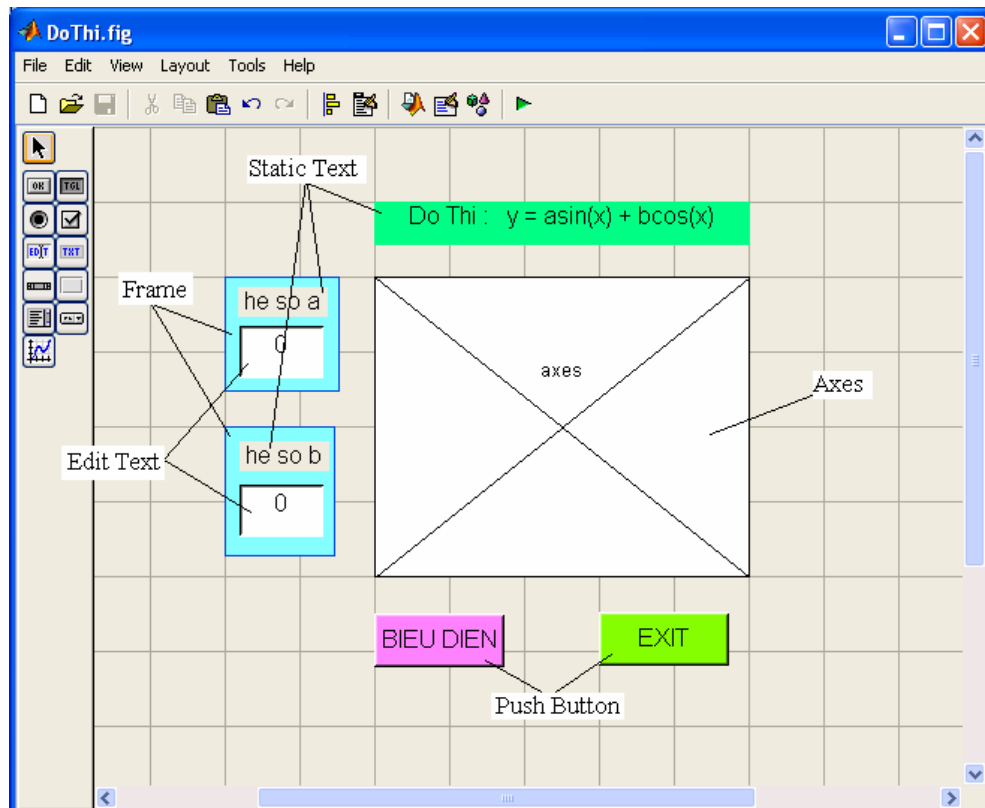


%En of proram

% --- Executes during object creation, after setting all properties.

2. Sinh viên thực hiện chương trình sau khi ta nhập các hệ số cho đồ thị và xuất dạng đồ thị $y = a \sin(x) + b \cos(x)$ ra màn hình.

Với giao diện đề nghị như sau:



Để có được hình như trên ta cần tạo các đối tượng sau:

Edit Text (số lượng 2)				
STT	Fontsize	String	Tag	Color
1	12	0	a	White
2	12	0	b	White

Static Text (số lượng 8)				
STT	Fontsize	String	Tag	color
1	12	Do Thi y = asin(x) + bcos(x)	Text1	Tùy ý
2	12	He so a	Text2	Tùy ý
3	12	He so b	Text3	Tùy ý



Axes (số lượng 1)				
STT	Xgrid	Ygrid	Tag	NextPlot
1	on	on	DoThi	replacechildren

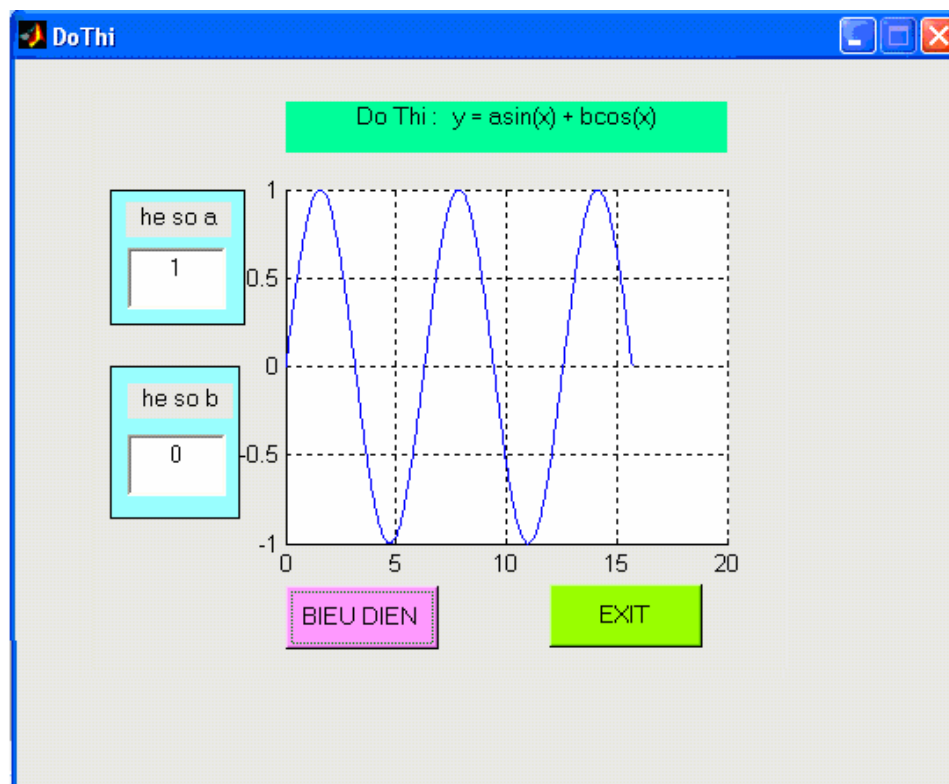
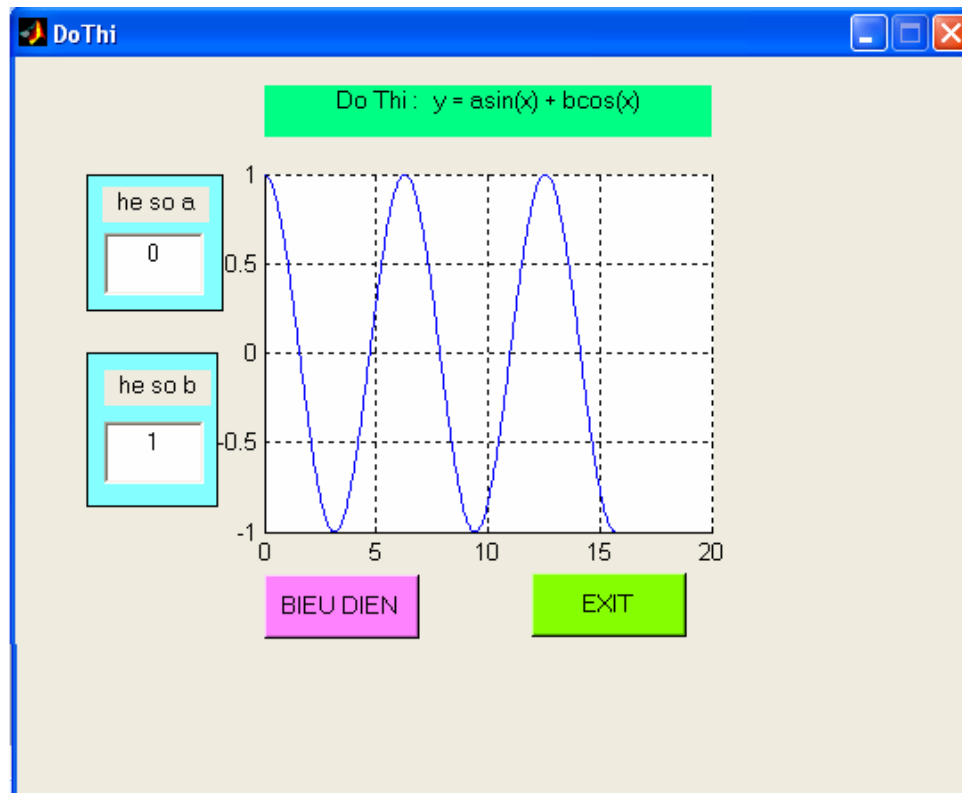
Push Button (số lượng 1)				
STT	Fontsize	String	Tag	Color
1	12	BIEU DIEN	DoThi	Tùy ý
2	12	EXIT	Close	Tùy ý

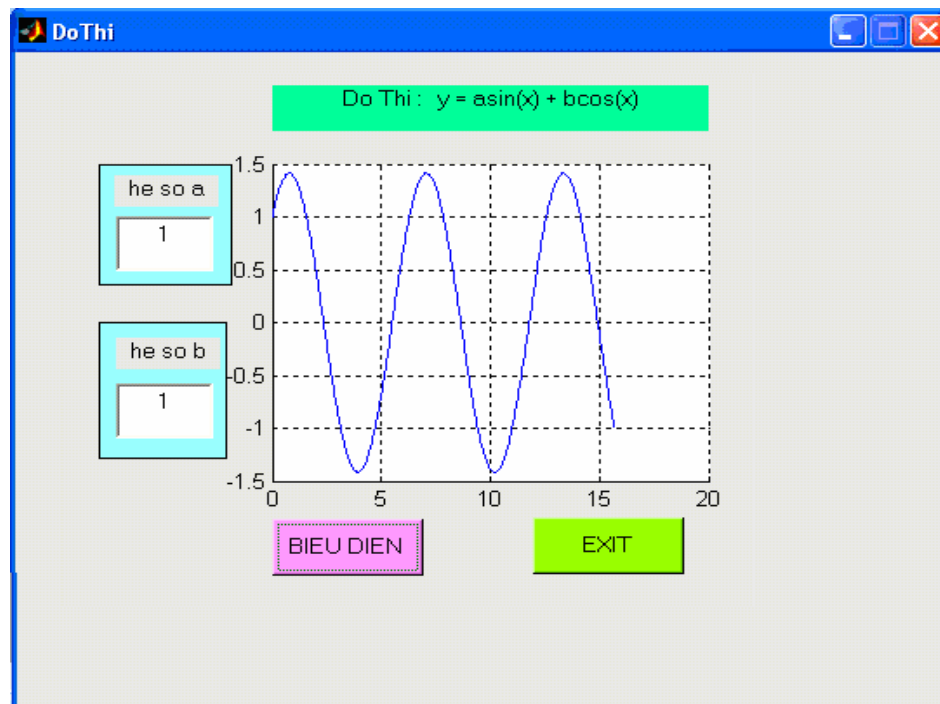
Cũng tương tự như bài 1, Sau khi tạo xong các đối tượng trên, ta nên save nó với tên (DotThi.fig), rồi tiếp tục Chọn view → M-file editor ở trên cửa sổ của giao diện guide vừa tạo xong, khi đó sẽ xuất hiện một cửa sổ của M-file có chương trình của giao diện guide mà ta vừa tạo, ta soạn thảo thêm đoạn chương trình dưới đây vào cuối chương trình của m-file đó.

```
% Doan chuong trinh thuc hien ve do thi
function DoThi_Callback(hObject, eventdata, handles)
a = str2double(get(handles.a,'string')) % cho phép nhập hệ số a ở Edit Text
b = str2double(get(handles.b,'string')) % cho phép nhập hệ số b ở Edit Text
x = 0:pi/100:5*pi;
y = a*sin(x) + b*cos(x);
plot(x,y,'b');
set(handles.axes,'string')
% End of program
```

Sau khi soạn xong đoạn chương trình đó, ta save và chạy chương trình và nhập lần lượt các hệ số a và b vào, nhấn vào nút BIEU DIEN cho ta kết quả như hình dưới :







Khi nhấn vào nút EXIT lập tức cửa sổ giao diện Guide sẽ đóng lại

Chương trình M-file tạo ra giao diện trên là :

```
% Begin of program:
function varargout = DoThi(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @DoThi_OpeningFcn, ...
    'gui_OutputFcn', @DoThi_OutputFcn, ...
    'gui_LayoutFcn', [] , ...
    'gui_Callback', []);
if nargin & isstr(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
```



```
% --- Executes just before DoThi is made visible.
function DoThi_OpeningFcn(hObject, eventdata, handles, varargin)

% Choose default command line output for DoThi
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = DoThi_OutputFcn(hObject, eventdata, handles)
% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes during object creation, after setting all properties.
function a_CreateFcn(hObject, eventdata, handles)
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function a_Callback(hObject, eventdata, handles)

function b_CreateFcn(hObject, eventdata, handles)

if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function b_Callback(hObject, eventdata, handles)

% --- Executes on button press in close.
function close_Callback(hObject, eventdata, handles)

% Doan chương trình thực hiện vẽ đồ thị
function DoThi_Callback(hObject, eventdata, handles)
a = str2double(get(handles.a,'string')) % cho phép nhập hệ số a ở Edit Text
b = str2double(get(handles.b,'string')) % cho phép nhập hệ số b ở Edit Text
x = 0:pi/100:5*pi;
y = a*sin(x) + b*cos(x);
```





```
plot(x,y,'b');  
set(handles.axes,'string')  
% End of proram
```

=====Hết Bài Tập Thực Hành Số 3=====



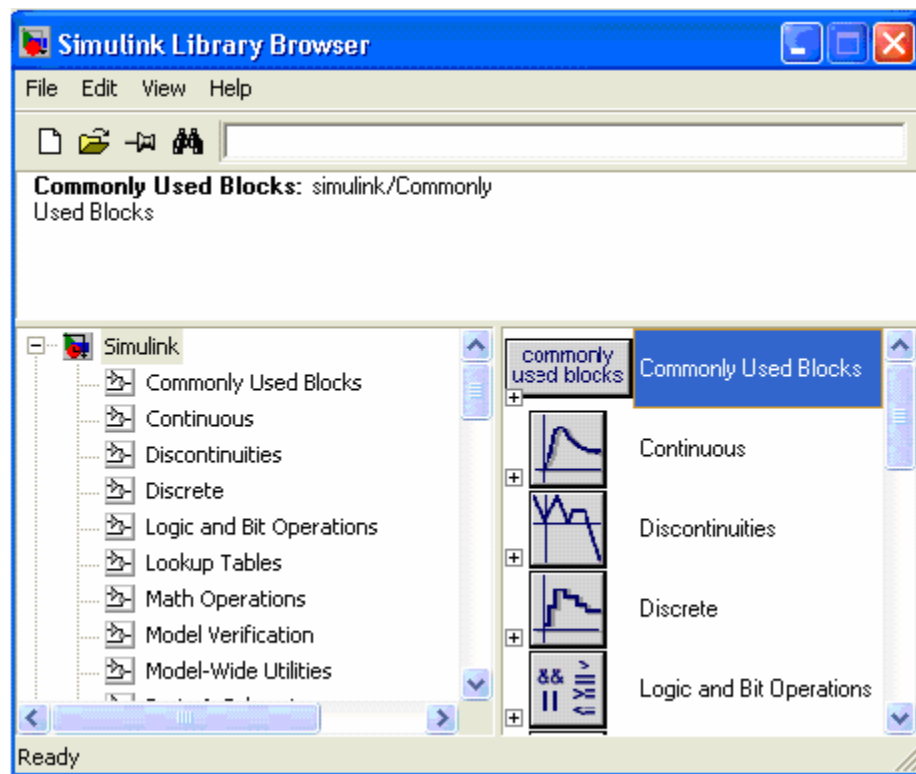
BÀI TẬP THỰC HÀNH SỐ 4

Simulink ứng dụng trong các ngành kỹ thuật

I. Thực hiện các thao tác trên màn hình chính của simulink.

Cài đặt giao diện của Simulink :

- Simulink Library Browser : cửa sổ chính.
- Simulink library : cửa sổ chứa các thư viện.
- Simulink parameters : cửa sổ hiệu chỉnh các thông số của mô hình simulink.
- Model : cửa sổ cho phép soạn thảo mô hình simulink
- Scope : cửa sổ hiển thị đồ thị mô phỏng của mô hình simulink.
- Làm việc với simulink, khởi động simulink bằng lệnh
 `>>simulink`
- Cửa sổ chính Simulink Library Browser xuất hiện như hình:



II. Thực hiện các thao tác trên màn hình soạn thảo mô hình.

Tạo mô hình simulink : Tạo mô hình simulink gồm các bước là

- Khởi tạo Simulink.
- Mở cửa sổ mới model với lệnh file-new-model.
- Soạn thảo mô hình với các khối từ thư viện simulink library.
- Chỉnh sửa dữ liệu trong mô hình.
- Liên kết các khối trong mô hình.

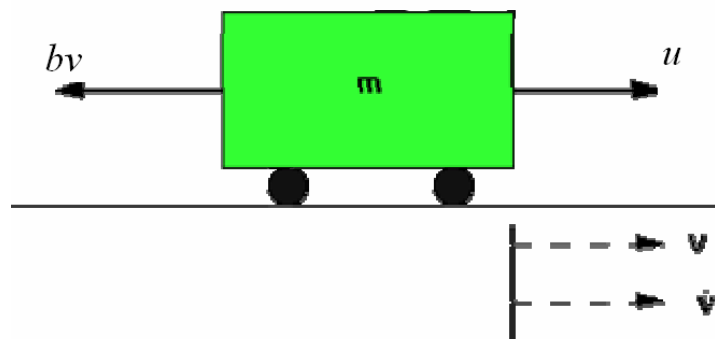
- Lưu mô hình lên đĩa với lệnh file-save từ cửa sổ model với phần đuôi mặc định là .mdl.
- Chạy mô hình với lệnh simulation-start từ cửa sổ model.

Soạn thảo mô hình :

- Chọn khối : click chuột lên khối đó.
- Hủy bỏ khối : Click chuột lên khối đó và nhấn phím delete.
- Di chuyển khối : Drag chuột kéo khối đến vị trí mong muốn
- Liên kết các khối : Drag chuột từ ngõ ra của khối này đến ngõ vào của khối khác.
- Với các đường khuỷu : Drag chuột kết hợp với phím Ctrl.
- Chọn toàn bộ mô hình : Drag chuột bao quanh mô hình.

Các thư viện cơ bản của simulink :

- *continuous* : thư viện chứa các khối cho phép xây dựng các mô hình của các hệ liên tục.
- *discontinuities* : thư viện chứa các khối cho phép xây dựng các mô hình của các hệ gián đoạn.
- *discrete* : thư viện chứa các khối cho phép xây dựng các mô hình của các hệ rời rạc.
- *user-defined-function* : thư viện chứa các khối cho phép người sử dụng dùng hàm hoặc S-function tự tạo để xây dựng các mô hình theo ý muốn.
- *sinks* : thư viện chứa các khối xuất tín hiệu của mô hình.
- *sources* : thư viện chứa các khối nhập dữ liệu của mô hình.
- *Matlab operations* : thư viện chứa các khối toán học.
- *Subsystem* : thư viện chứa các khối cho phép tạo các hệ con của mô hình.

III. Xây dựng mô hình hệ thống xe tải:

Mô hình xe tải cho bởi phương trình:

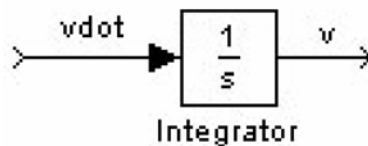
$$m \frac{dv}{dt} = u - bv \text{ hay } \frac{dv}{dt} = \frac{1}{m}(u - bv)$$

Trong đó m là khối lượng xe, u là lực tác động của động cơ (ngõ vào của mô hình), b là hệ số ma sát và v là vận tốc đạt được (ngõ ra của mô hình).

Xuất phát của việc xây dựng các mô hình hệ thống từ các phương trình vi phân tương ứng là các khối tích phân (Integrator). Nếu trong phương trình mô tả hệ thống có vi phân bậc n thì ta sẽ đặt vào mô hình n khối tích phân, do quan

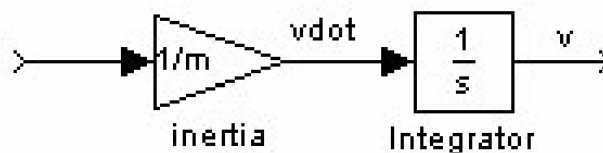
$$\text{hệ } \int \frac{dv}{dt} = v$$

- Mở một cửa sổ mô hình mới.
- Đặt vào mô hình khối 'Integrator' từ thư viện 'Continuous' và kẻ các đường thẳng nối đến ngõ vào và ngõ ra của khối này.
- Đặt nhãn 'vdot' (dv/dt) cho cho đường nối đến ngõ vào và 'v' cho đường nối đến ngõ ra bằng cách nhấp đúp chuột ngay phía trên các đường này.



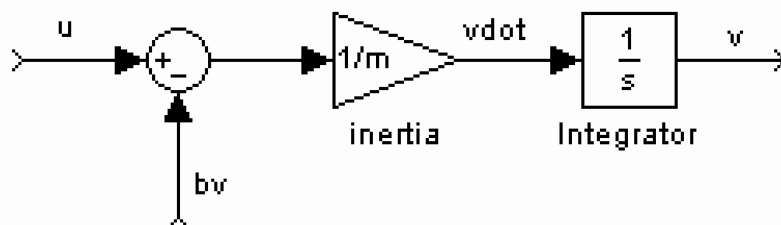
Từ phương trình hệ thống ta thấy dv/dt (vdot) bằng tích của thành phần (1/m) và thành phần tổng (u-bv), nên ta thêm khối 1/m trước khối tích phân:

- Đặt vào khối 'Gain' trong thư viện *Math*.
- Nhấp đúp chuột vào khối này để thay đổi độ lợi thành 1/m.
- Đặt nhãn 'inertia' cho khối này để tượng trưng cho quán tính của xe (nhấp đúp vào nhãn 'Gain' bên dưới khối).



Bây giờ ta đặt khối tổng với 2 ngõ vào '+-', ngõ vào '+' sẽ được nối với u, ngõ vào '-' sẽ được nối với thành phần bv để được (u-bv)

- Đặt vào khối 'Sum' trong thư viện *Math*
- Nhấp đúp vào khối này để đổi ngõ vào từ '++' sang '+-'

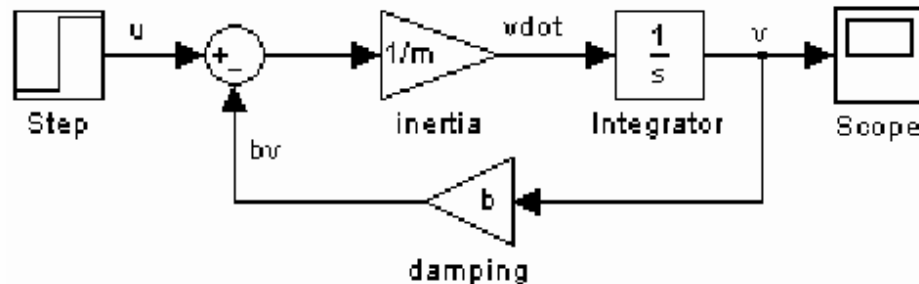


Để được thành phần bv ta chỉ cần đặt thêm khối 'Gain' với độ lợi b:

- Đặt khối 'Gain' có độ lợi b
- Đặt nhãn là 'damping' tượng trưng cho thành phần lực cản của xe.

Đến đây việc xây dựng mô hình xe tải với ngõ vào u và ngõ ra v coi như hoàn thành. Tuy nhiên, để mô phỏng mô hình này, ta cần đặt thêm khối 'Step' vào u và hiển thị v trên khối 'Scope'

- Đặt khối 'Step' trong thư viện *sources* biên độ u ngay ngõ vào.
- Đặt khối 'Scope' trong thư viện *sinks* ngay ngõ ra v .



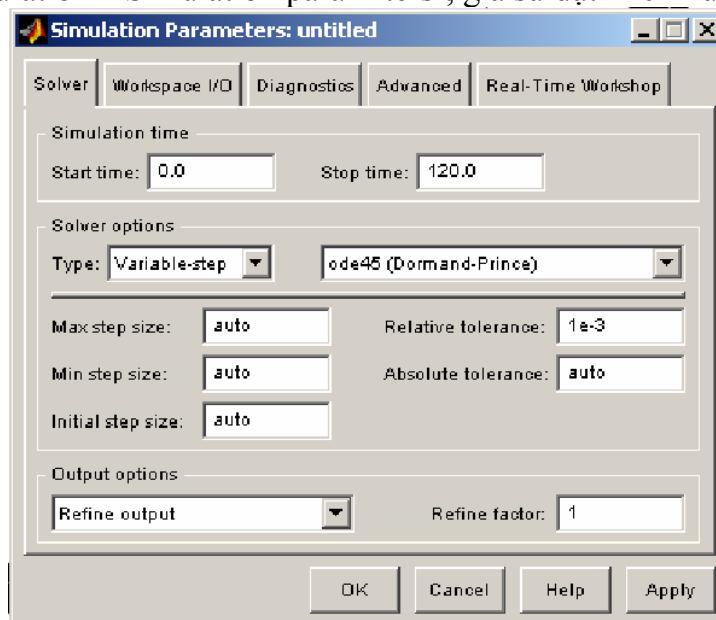
Nhớ rằng m , b và u là các biến cần được gán trị trước khi mô phỏng.

```
>>m=1000
```

```
>>b=50
```

```
>>u=500
```

Thời gian mô phỏng hệ thống tùy thuộc vào thông số 'Stop time' trong menu 'Simulation→Simulation paramrters', giả sử đặt 120 như Hình.



Để chạy mô phỏng, ta có thể thực hiện bằng 1 trong các cách:

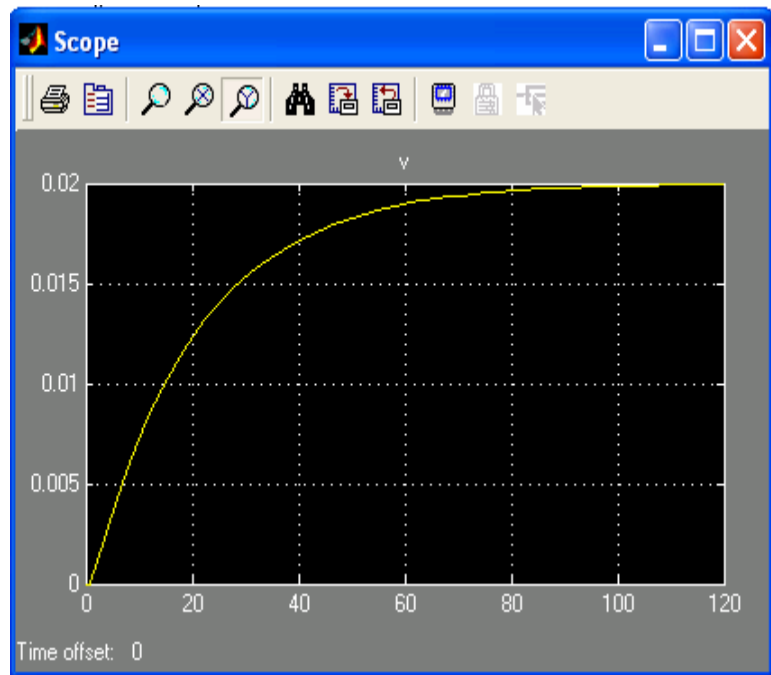
- nhấp chuột vào biểu tượng  trên menubar của mô hình
- chọn: Simulation → Start
- Ctrl-T

Nhấp đúp vào 'Scope' để xem kết quả mô phỏng.

Sinh viên thực hiện kiểm tra và lưu kết quả.

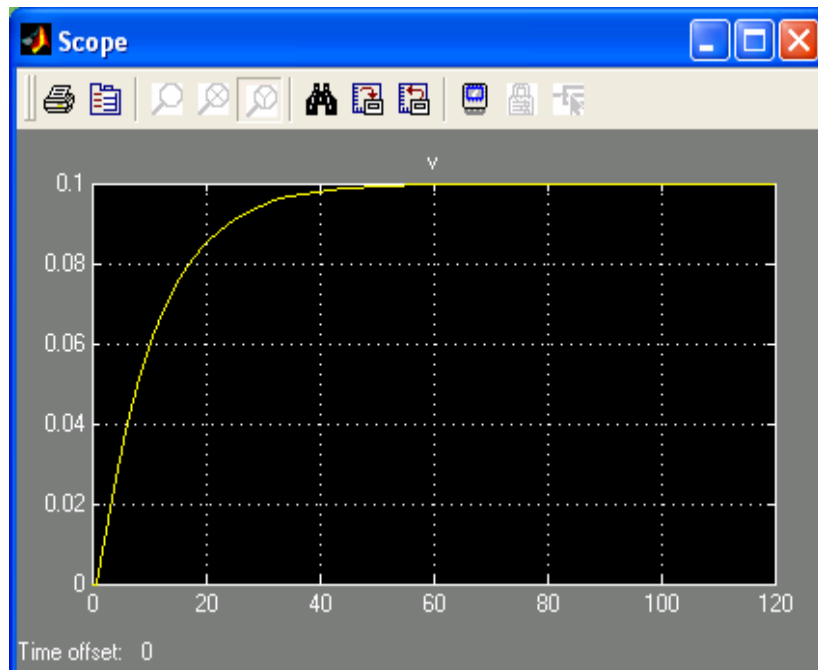
Khi ta nhập các giá trị và chạy mô phỏng được kết quả là hình dưới

```
>> u = 500;  
>> m = 1000;  
>> b = 50;
```



Khi ta thay đổi các giá trị và chạy mô phỏng được kết quả là hình dưới:

```
>> m = 100;  
>> b = 10;  
>> u = 1000;
```



IV. Xây dựng mô hình hệ thống điều khiển vị trí motor DC:

Hệ thống điều khiển vị trí motor DC cho bởi phương trình vi phân sau:

$$\frac{d^2\theta}{dt^2} = \frac{1}{J} \left(K_t i - b \frac{d\theta}{dt} \right)$$

$$\frac{di}{dt} = \frac{1}{L} \left(-Ri + V - K_e \frac{d\theta}{dt} \right)$$

Trong đó:

$J = 0.01 \text{ Kg}\cdot\text{m}^2/\text{s}^2$

$b = 0.1 \text{ Mms}$

$K = K_e = K_t = 0.01 \text{ Nm/A}$

$R = 10 \text{ ohm}$

$L = 0.5 \text{ H}$

V

θ

i

motor.

là moment quán tính của rotor

là hệ số ma sát của các bộ phận cơ khí

là hằng số sức điện động

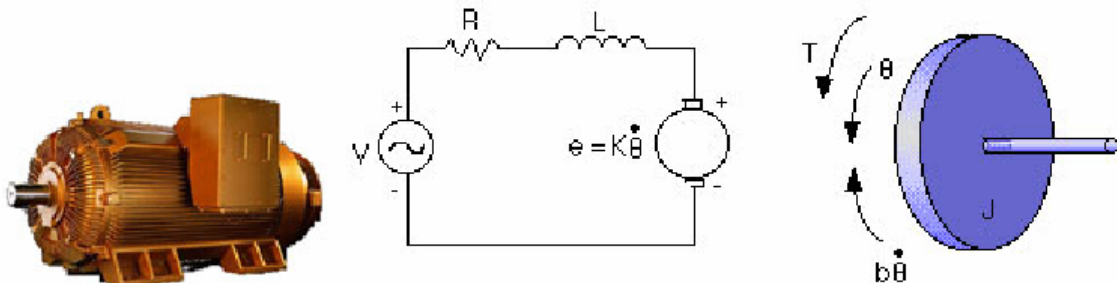
là điện trở dây quấn

là hệ số tự cảm

là điện áp đặt lên cuộn dây của motor

là vị trí trục quay (ngõ ra của mô hình)

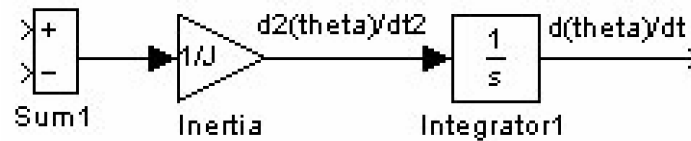
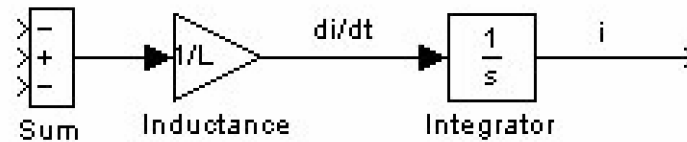
là dòng điện chạy trong cuộn dây của



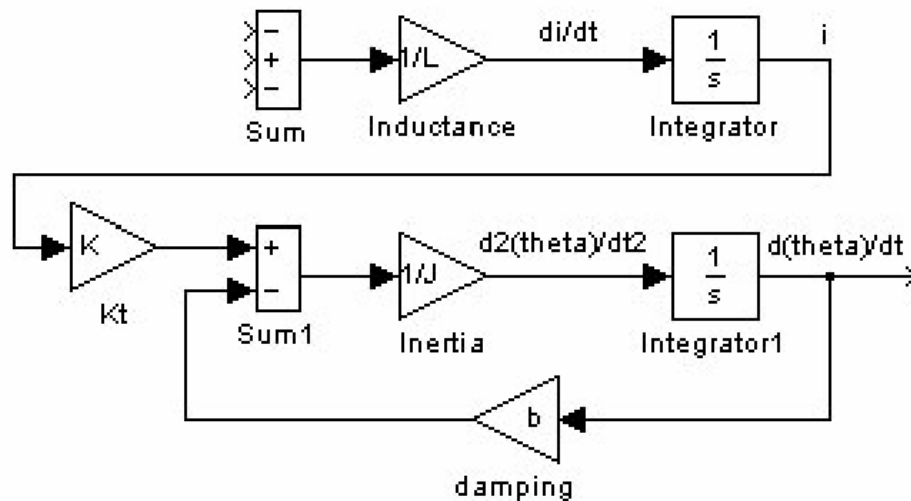
Mô hình toán hệ điều khiển vị trí motor DC

Quan sát từng phương trình mô tả hệ thống ta thấy cấu trúc của chúng cũng tương tự như phương trình của câu trước. Sinh viên lần lượt thực hiện trên từng phương trình để được hình :

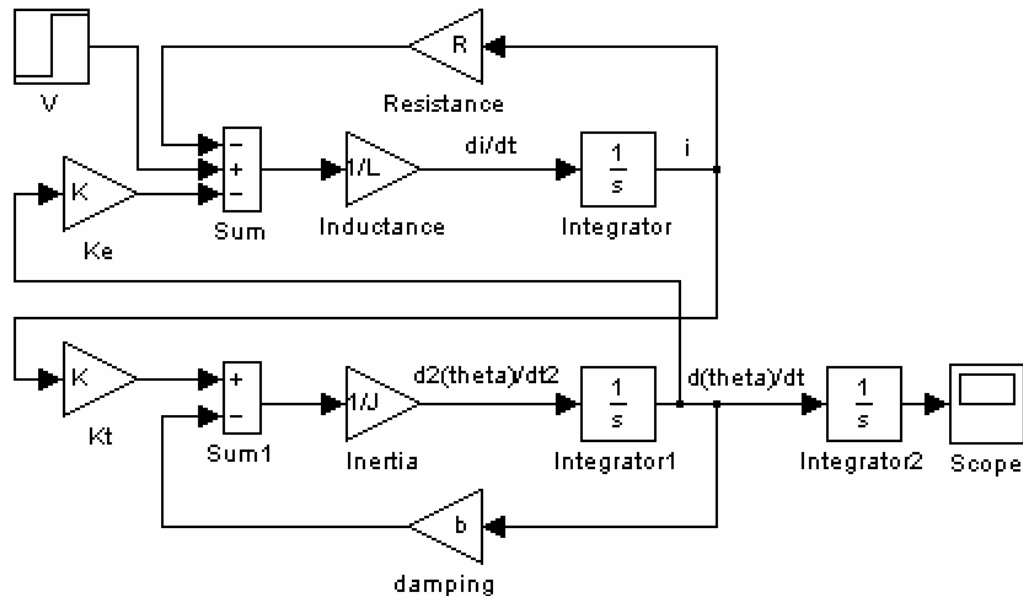
Tương tự như bài trên ta vào các thư viện lần lượt lấy ra các khối như hình dưới: vào thư viện Neural Network Blockset /Net Input Functions chọn khối netsum , sau đó nhấp đúp vào khối này để đổi số 2 sang '+-' trong ô "List of signs" ta được khối sum 1 như hình dưới. Tương tự làm như vậy nếu ta thay số 2 thành '- + -' thì ta được khối Sum như hình dưới.



Sau khi lấy ra các khối, ta thay đổi tên và thông số cho khối cũng như đường kết nối giữa các khối như hình. Kết hợp 2 phương trình:



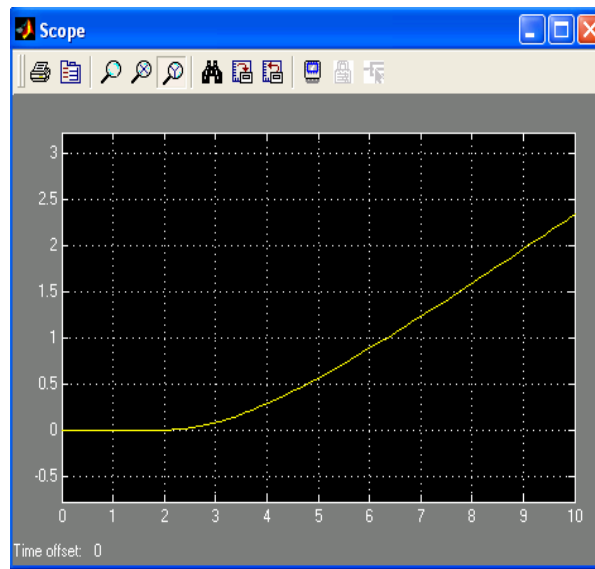
Đặt vào mô hình khối 'Step' để làm tín hiệu tham khảo, khối 'Scope' để quan sát đáp ứng. Từ mô hình Simulink hoàn chỉnh Sinh viên hãy gán trị cho tất cả các thông số của mô hình, thực hiện mô phỏng và quan sát đáp ứng. Sinh viên thực hiện kiểm tra và lưu kết quả.



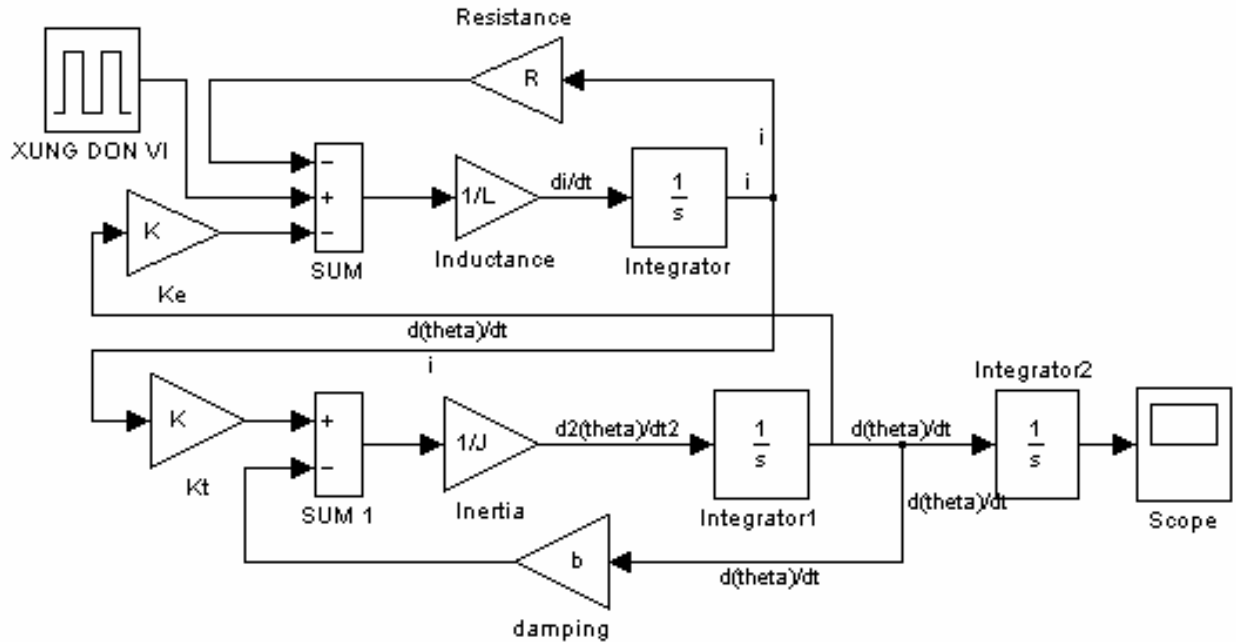
Mô hình Simulink hoàn chỉnh của hệ điều khiển vị trí motor DC

Gán các giá trị vào cho các tham số và chạy mô phỏng cho ta kết quả như hình dưới:

```
>> J = 0.1;
>> b = 0.05;
>> K = 0.1;
>> Kt = 0.1;
>> Ke = 0.1;
>> R = 5;
>> L = 0.2;
>> V = 220;
```

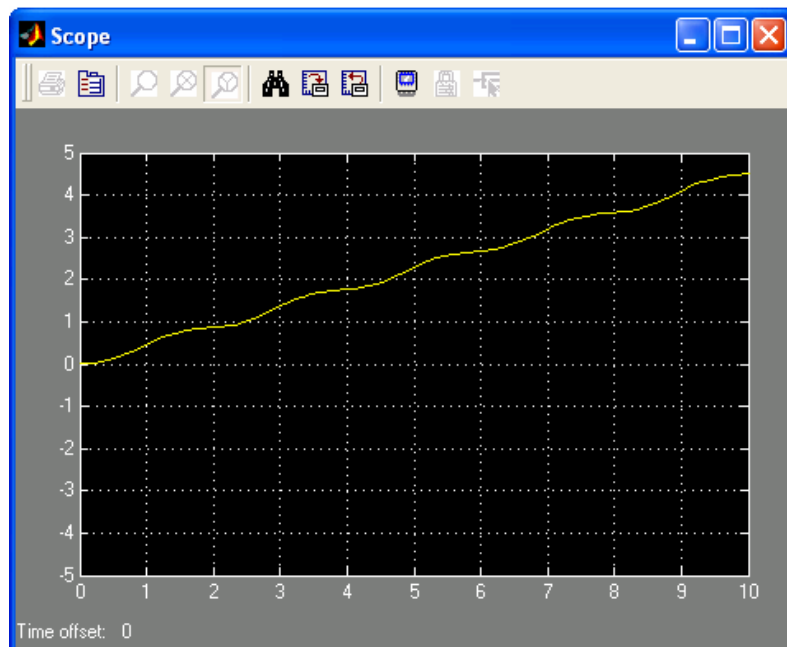


Hãy thay đổi ngõ vào mô hình bằng khối tạo **xung vuông**. Sinh viên thực hiện kiểm tra và lưu kết quả.



Gán các giá trị vào cho các tham số và chạy mô phỏng cho ta kết quả như hình dưới:

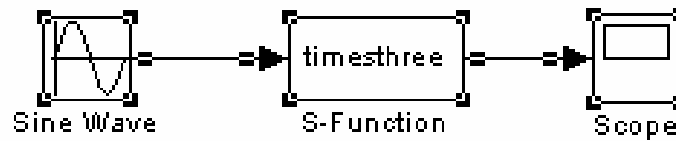
```
>> V = 240;
>> L = 0.5;
>> R = 5;
>> Ke = 0.1;
>> Kt = 0.1;
>> K = 0.1;
>> b = 0.02;
>> J = 0.01;
```



V. Xây dựng các S m-file function.

Để hoàn thành tốt phần thí nghiệm này sinh viên cần tham khảo lại phần lý thuyết S-function.

1. Three Time function



Xây dựng mô hình simulink như hình, S m-file function có tác dụng nhân 3 lần tín hiệu ngõ vào

Mã nguồn tham khảo S m-file function

```

function [sys,x0,str,ts] = timesthree(t,x,u,flag)
switch flag,
case 0
    [sys,x0,str,ts]=mdlInitializeSizes;
case 3
    sys=mdlOutputs(t,x,u);
case { 1, 2, 4, 9 }
    sys=[];
otherwise
    error(['Unhandled flag = ',num2str(flag)]);
end
function [sys,x0,str,ts] = mdlInitializeSizes()
sizes = simsizes;
sizes.NumContStates = 0;
sizes.NumDiscStates = 0;
sizes.NumOutputs = 1;
sizes.NumInputs = 1;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1;
sys = simsizes(sizes);
str = [];
x0 = [];
ts = [-1 0];
function sys = mdlOutputs(t,x,u)
sys = u * 3;
% end mdlOutputs
Sinh viên thực hiện kiểm tra và lưu kết quả
  
```

=====Hết Bài Tập Số 4=====



BÀI THÍ NGHIỆM 1

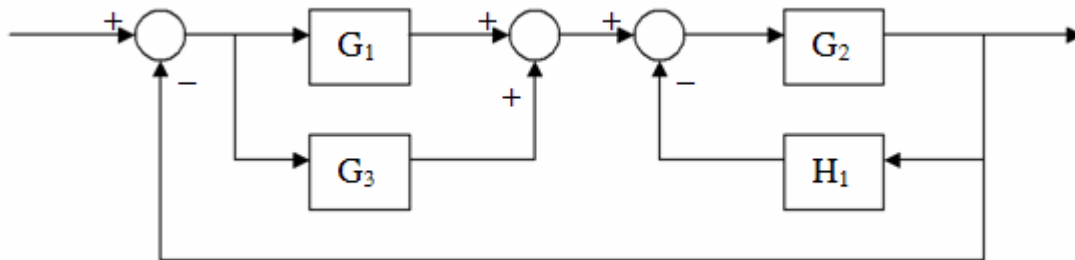
ỨNG DỤNG MATLAB KHẢO SÁT HỆ THỐNG DÙNG GIẢN ĐỒ BODE

Câu .1. Tìm hàm truyền tương đương của hệ thống:▪ **Mục đích:**

Giúp sinh viên làm quen với các lệnh cơ bản để kết nối các khối trong một hệ thống.

▪ **Thí nghiệm:**

Bằng cách sử dụng các lệnh cơ bản **conv**, **tf**, **series**, **parallel**, **feedback** ở phần phụ lục chương 2 (trang 85) trong sách *Lý thuyết điều khiển tự động* (Phần phụ lục này có kèm theo cuối bài hướng dẫn), tìm biểu thức hàm truyền tương đương $G(s)$ của hệ thống sau :



$$G1 = \frac{s+1}{(s+3)(s+5)}, G2 = \frac{s}{s^2+2s+8}, G3 = \frac{1}{s}, H1 = s+2$$

▪ **Hướng dẫn:**

Trên cửa sổ commad windown lần lượt đánh các lệnh trước dấu nhảy như sau:

```
>> G1 = tf([1 1],conv([1 3],[1 5])) % nhập hàm truyền G1
```

Transfer function:

$$s + 1$$

$$s^2 + 8s + 15$$

```
>> G2 = tf([1 0],[1 2 8]) % nhập hàm truyền G2
```

Transfer function:





```
s
-----
s^2 + 2 s + 8

>> G3 = tf(1,[1 0])           % nhập hàm truyền G3

Transfer function:
1
-
s

>> H1 = tf([1 2],1)           % nhập hàm truyền H1

Transfer function:
s + 2

>> G13 = parallel(G1,G3)       % G13 = G1 + G3 , 2 hàm truyền song song

Transfer function:
2 s^2 + 9 s + 15
-----
s^3 + 8 s^2 + 15 s

>> G2H1 = feedback(G2,H1)      % G2H1 = G2 / (1 + G2*H1) của hệ hồi tiếp âm.

Transfer function:
s
-----
2 s^2 + 4 s + 8

>> Gh = series(G13,G2H1)       % Gh = G13*G2H1 hệ thống nối tiếp

Transfer function:
2 s^3 + 9 s^2 + 15 s
-----
2 s^5 + 20 s^4 + 70 s^3 + 124 s^2 + 120 s
```





```
>> Gk = feedback(Gh,1)           % Gk = Gh/(1+ Gh*H) , voi H(s) = 1
```

Transfer function:

$$2s^3 + 9s^2 + 15s$$

% kết quả hàm truyền cần tìm.

$$2s^5 + 20s^4 + 72s^3 + 133s^2 + 135s$$

Câu .2.a. Khảo sát hệ thống dùng biểu đồ Bode:

▪ Mục đích:

Từ biểu đồ Bode của hệ hở $G(s)$, ta tìm được tần số cắt biên, độ dự trữ pha, tần số cắt pha, độ dự trữ biên của hệ thống hở. Dựa vào kết quả tìm được để xét tính ổn định của hệ thống hồi tiếp âm đơn vị với hàm truyền vòng hở là $G(s)$.

▪ Thí nghiệm:

Khảo sát hệ thống phản hồi âm đơn vị có hàm truyền vòng hở:

$$G(s) = \frac{K}{(s+0.2)(s^2+8s+20)}$$

a. Với $K = 10$, vẽ biểu đồ Bode biên độ và pha hệ thống trên trong khoảng tần số (0.1, 100).

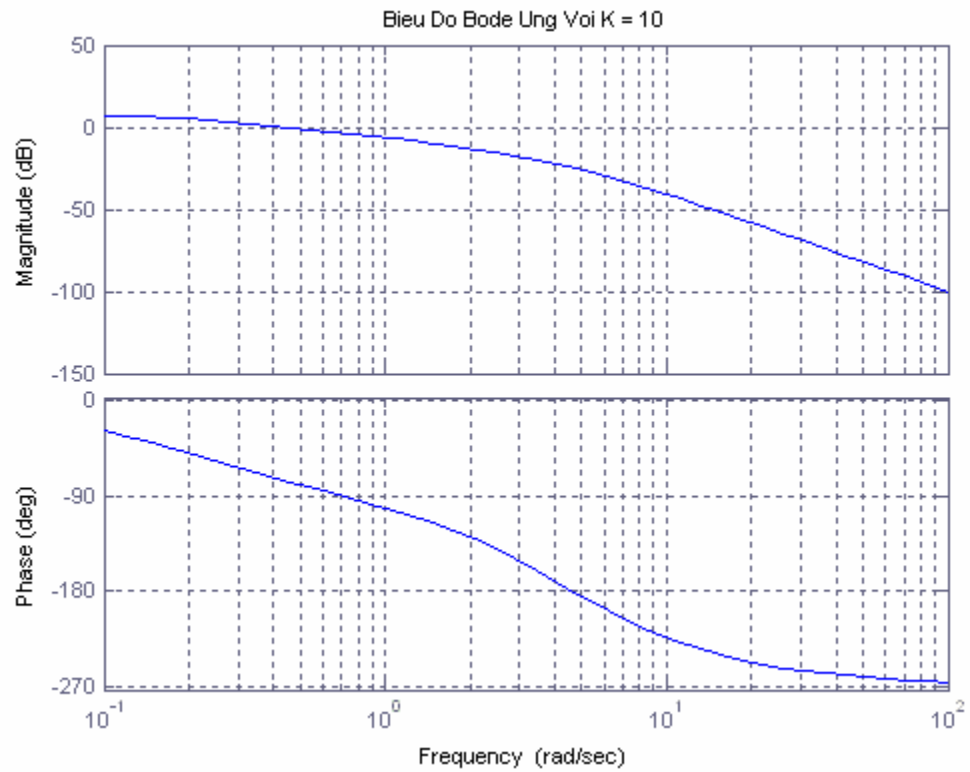
HD:

Nhập lệnh vẽ biểu đồ Bode của $G(s)$ khi $K=10$ như sau:

```
>> TS = 10 ;
>> MS = conv([1 0.2],[1 8 20]) ;
>> G = tf(TS,MS) ;
>> bode(G,{0.1,100})
>> grid on
```

Kết quả ta có hình như sau:

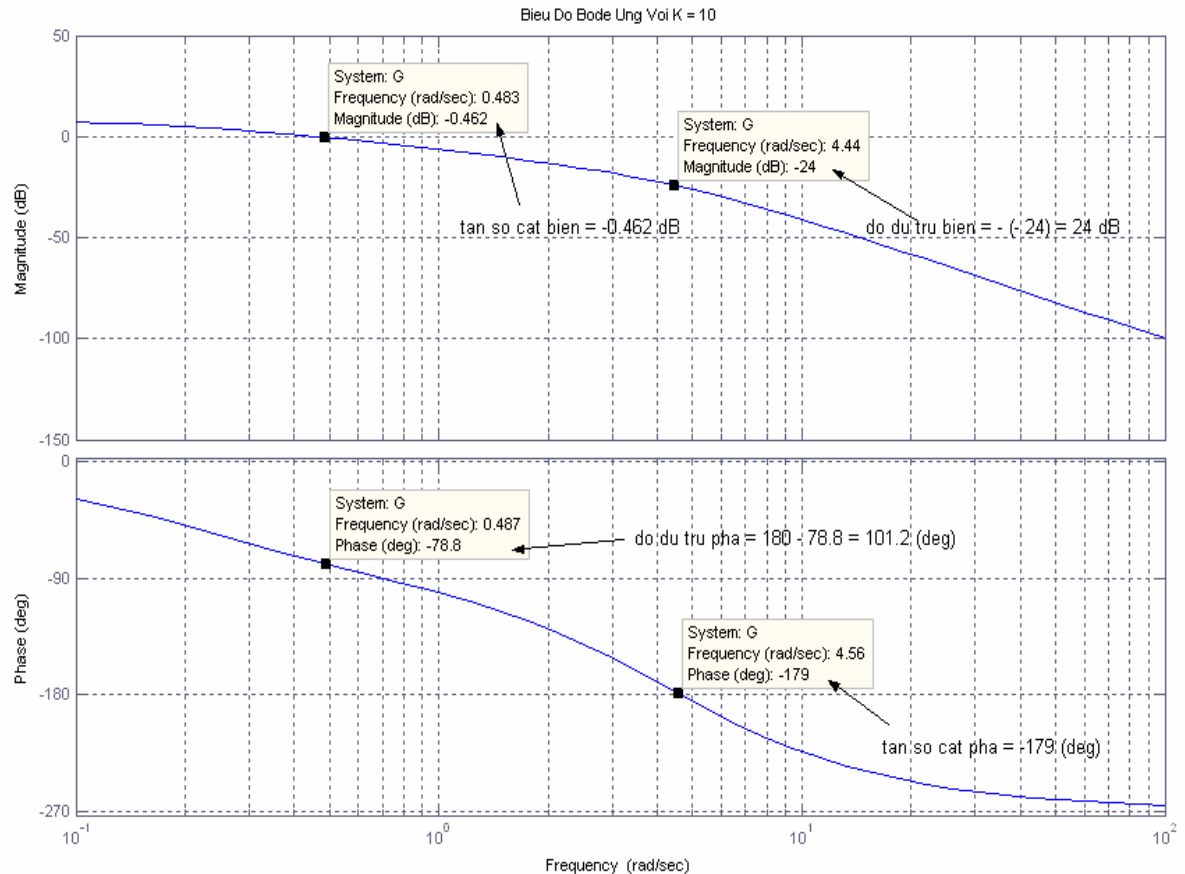




b. Dựa vào biểu đồ Bode, tìm tần số cắt biên, độ dự trữ pha, tần số cắt pha, độ dự trữ biên của hệ thống

HD:

Kết quả cần tìm như hình vẽ:



Tần số cắt biên $\omega_c = -0.462$ dB, tần số cắt pha = -179°

Độ dự trữ biên $GM = 24$ dB, độ dự trữ pha $\phi_M = 101.2^\circ$

c. Hệ thống trên có ổn định không, giải thích.

HD:

Hệ thống trên có ổn định. Tại vì Độ dự trữ biên $GM = 24$ dB, độ dự trữ pha $\phi_M = 101.2^\circ$ đều dương (Theo lý thuyết điều khiển tự động).

d. Vẽ đáp ứng quá độ của hệ thống trên với đầu vào hàm nấc đơn vị trong khoảng thời gian $t = 0 \div 10$ s để minh họa kết luận ở câu c

HD:

Để vẽ đáp ứng của hệ kín ta nhập các lệnh sau:



```
>> TS = 10 ;  
>> MS = conv([1 0.2],[1 8 20]) ;  
>> G = tf(TS,MS) ;  
>> Gk = feedback(G,1)
```

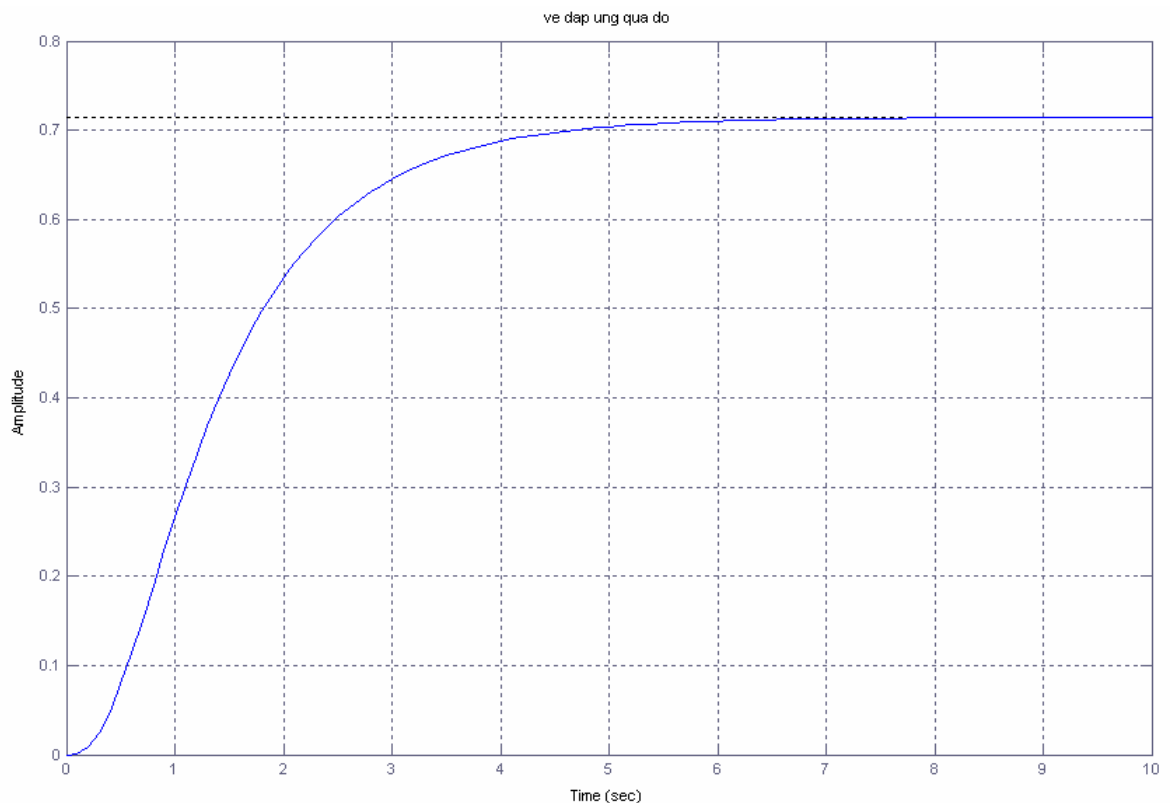
Transfer function:

10

 $s^3 + 8.2 s^2 + 21.6 s + 14$

```
>> step(Gk,10)  
>> grid on  
>> title('ve dap ung qua do')
```

Kết quả có được: ta thấy hệ thống ổn định.





Câu .2.b. Với $K = 400$, thực hiện lại các yêu cầu ở câu a \rightarrow d.

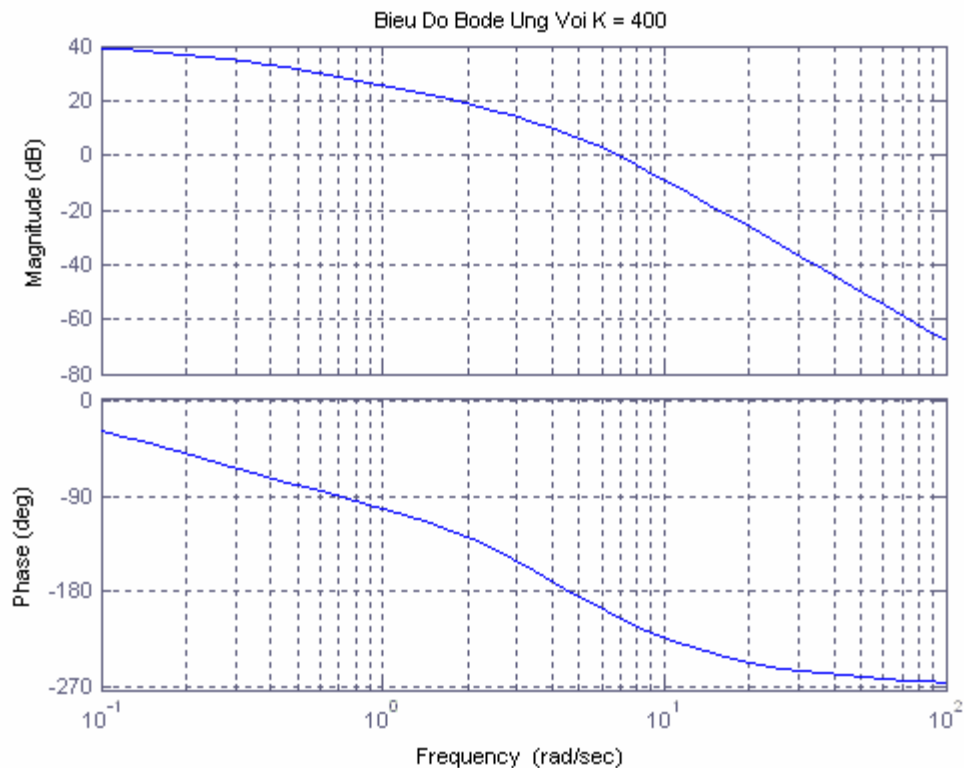
a. Với $K = 400$, vẽ biểu đồ Bode biên độ và pha hệ thống trên trong khoảng tần số $(0.1, 100)$.

HD:

Nhập lệnh vẽ biểu đồ Bode của $G(s)$ khi $K=10$ như sau:

```
>> TS = 400 ;  
>> MS = conv([1 0.2],[1 8 20]) ;  
>> G = tf(TS,MS) ;  
>> bode(G,{0.1,100})  
>> grid on
```

Kết quả ta có hình như sau:

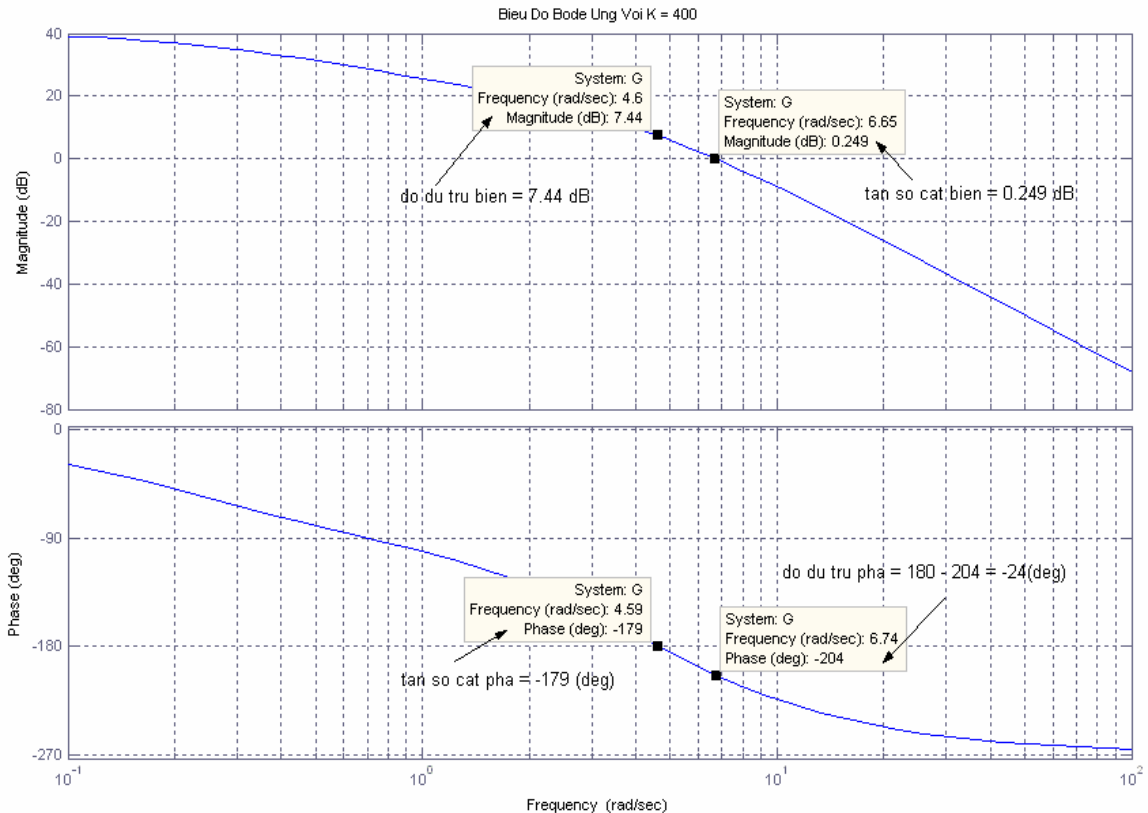


b. Dựa vào biểu đồ Bode, tìm tần số cắt biên, độ dự trữ pha, tần số cắt pha, độ dự trữ biên của hệ thống

HD:

Kết quả cần tìm như hình vẽ:





Tần số cắt biên $\omega_c = 0.249$ dB, tần số cắt pha $= -179^\circ$

Độ dự trữ biên $GM = -7.44$ dB, độ dự trữ pha $\phi M = 180 - 204 = -24^\circ$

c. Hệ thống trên có ổn định không ? giải thích.

HD:

Hệ thống trên không ổn định. Tại vì Độ dự trữ biên $GM = 7.44$ dB dương, còn độ dự trữ pha $\phi M = -24^\circ < 0$ âm (Theo lý thuyết điều khiển tự động).

d. Vẽ đáp ứng quá độ của hệ thống trên với đầu vào hàm nấc đơn vị trong khoảng thời gian $t = 0 \div 10$ s để minh họa kết luận ở câu c

HD:

Để vẽ đáp ứng của hệ kín ta nhập các lệnh sau:

```
>> TS = 400 ;
>> MS = conv([1 0.2],[1 8 20]) ;
>> G = tf(TS,MS) ;
>> Gk = feedback(G,1)
```

Transfer function:

400

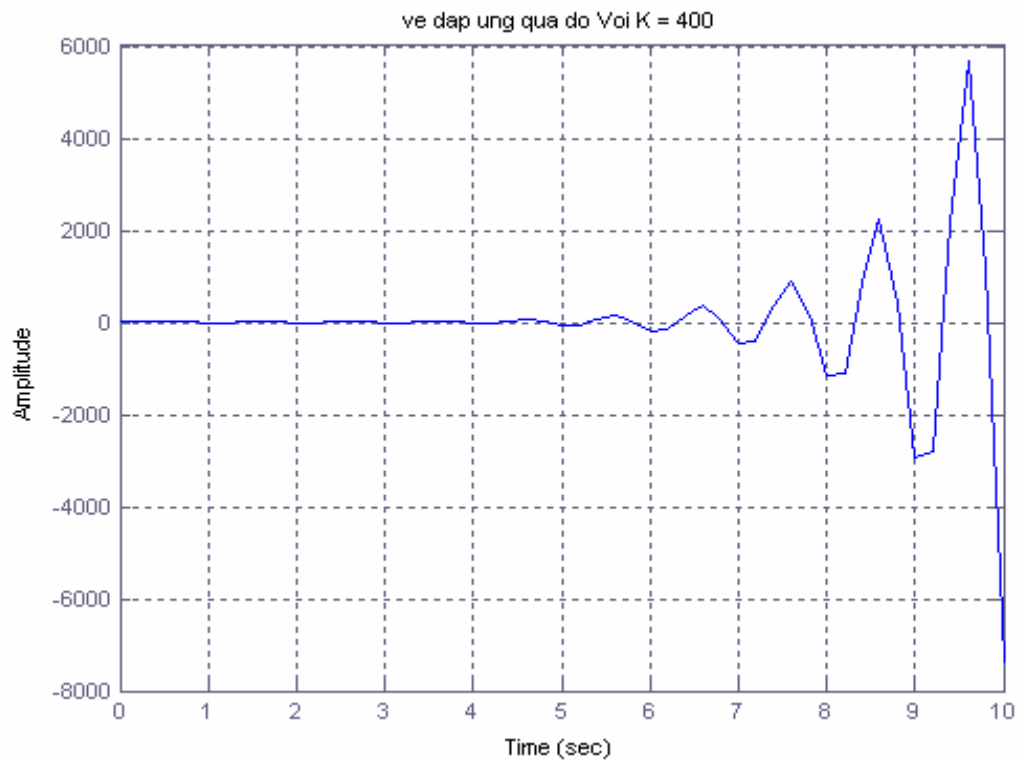




$$s^3 + 8.2 s^2 + 21.6 s + 404$$

```
>> step(Gk,10)
>> grid on
>> title('ve dap ung qua do')
```

Kết quả có được: ta thấy hệ thống ổn định.



----- HẾT BÀI 1 -----



**BÀI THÍ NGHIỆM 2****ỨNG DỤNG MATLAB KHẢO SÁT HỆ THỐNG BẰNG PHƯƠNG PHÁP****QUỖ ĐẠO NGHIỆM SỐ****Câu .1. Khảo sát hệ thống dùng phương pháp QĐNS:****▪ Mục đích:**

Khảo sát đặc tính của hệ thống tuyến tính có hệ số khuếch đại K thay đổi, tìm giá trị giới hạn K_{gh} của K để hệ thống ổn định.

▪ Thí nghiệm:

Hệ thống hồi tiếp âm đơn vị có hàm truyền vòng hở:

$$G(s) = \frac{K}{(s+3)(s^2+8s+20)}, \quad K \geq 0$$

a. Vẽ QĐNS của hệ thống. Dựa vào QĐNS, tìm K_{gh} của hệ thống, chỉ rõ giá trị này trên QĐNS. Lưu QĐNS này thành file *.bmp để viết báo cáo.

HD:

Khi nhập hàm truyền cho G ta không nhập tham số K trong lệnh tf. Dùng lệnh **grid on** để kẻ lưới:

```
>> TS = 1                % nhập tu so cua G(s) khong chua K
>> MS = conv([1 3],[1 8 20]) % nhập mau so cua G(s)
>> G = tf(TS,MS)         % nhập ham truyen G(s)
>> rlocus(G)             % vẽ quỹ đạo nghiệm số
>> grid on               % kẻ lưới
```

- Gain : giá trị độ lợi K tại vị trí nhấp chuột (giá trị K cần tìm).
- Pole : cực của hệ thống vòng kín tương ứng với giá trị K
- Damping : hệ số tắt ξ
- Overshoot : độ vọt lố
- Frequency : tần số dao động tự nhiên ω_n
- (A) : vòng tròn các điểm có cùng tần số dao động tự nhiên $\omega_n = 4$
- (B) : đường thẳng các điểm có cùng hệ số tắt $\xi = 0.68$
- (C) : đường thẳng các điểm có cùng $t_{xl} = \frac{4}{\xi\omega_n} = 4 \Rightarrow \xi\omega_n = 1$

Để tìm K_{gh} ta nhấp chuột vào vị trí cắt nhau giữa QĐNS với trục ảo. Lúc này, giá trị K sẽ hiển thị lên như trên hình vẽ sau:

→ $K_{gh} = K_a = 42.9$





b. Tìm K để hệ thống có tần số dao động tự nhiên $\omega_n = 4$

HD:

Muốn tìm K để hệ thống có tần số dao động tự nhiên $\omega_n = 4$, ta nhấp chuột vào vị trí giao điểm của QĐNS với vòng tròn $\omega_n = 4$ (vòng tròn (A)). Chọn giao điểm gần trục ảo (giao điểm M) để giá trị K này làm hệ thống có tính dao động.

$$\rightarrow K_b = 50.6$$

c. Tìm K để hệ thống có hệ số tắt $\xi = 0.7$

HD:

Để hệ thống có $\xi = 0.7$ ta nhấp chuột tại vị trí giao điểm (N) của QĐNS với đường thẳng $\xi = 0.7$ (đường thẳng (B)). Ta có thể chọn gần đúng đường thẳng $\xi = 0.68$ như ở trên hình vẽ.

$$\rightarrow K_c = 22.4$$

d. Tìm K để hệ thống có độ vọt lố $POT = 25\%$ HD:

$$\text{Tương tự cho } POT = \exp\left(-\frac{\xi\omega}{\sqrt{1-\xi^2}}\right) = 25\% \Rightarrow \xi = 0.4$$

$$\rightarrow K_d = 77.9$$

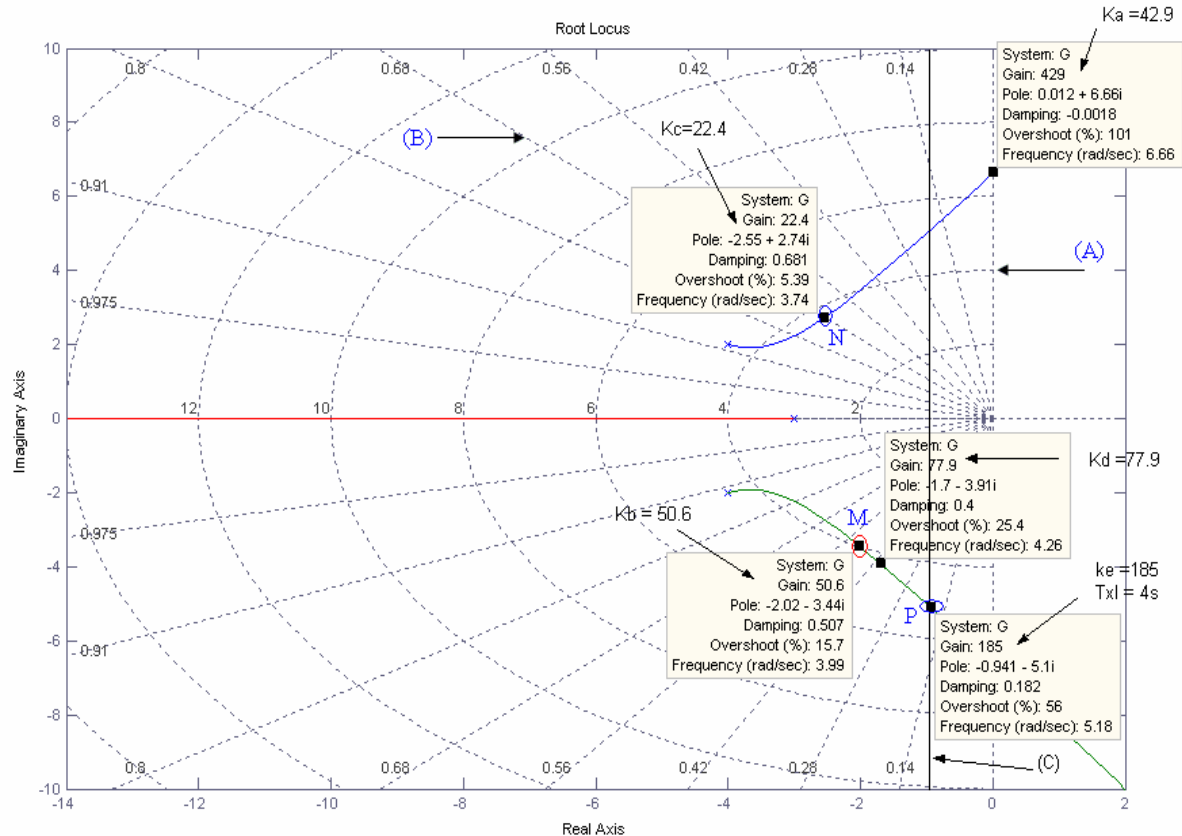
e. Tìm K để hệ thống có thời gian xác lập (tiêu chuẩn 2%) $t_{xl} = 4s$

HD:

$t_{xl} = \frac{4}{\xi\omega_n} = 4s \Rightarrow \xi\omega_n = 1$. Do đó muốn tìm K để hệ thống có $t_{xl} = 4$ ta nhấp chuột vào vị trí giao điểm (P) của QĐNS với đường thẳng $\xi\omega_n = 1$ (đường thẳng (C)).

$$\rightarrow K_e = 185$$





Câu .2. Đánh giá chất lượng của hệ thống:

Mục đích:

Khảo sát đặc tính quá độ của hệ thống với đầu vào hàm nấc để tìm độ vọt lố và sai số xác lập của hệ thống.

Thí nghiệm:

Với hệ thống như ở phần III.1 :

a. Với giá trị $K = K_{gh}$ tìm được ở trên, vẽ đáp ứng quá độ của hệ thống vòng kín với đầu vào hàm nấc đơn vị. Kiểm chứng lại đáp ứng ngõ ra có dao động không?

HD:

```
>> TS = 1;
>> MS = conv([1 3],[1 8 20]);
>> G = tf(TS,MS)
```

Transfer function:



1

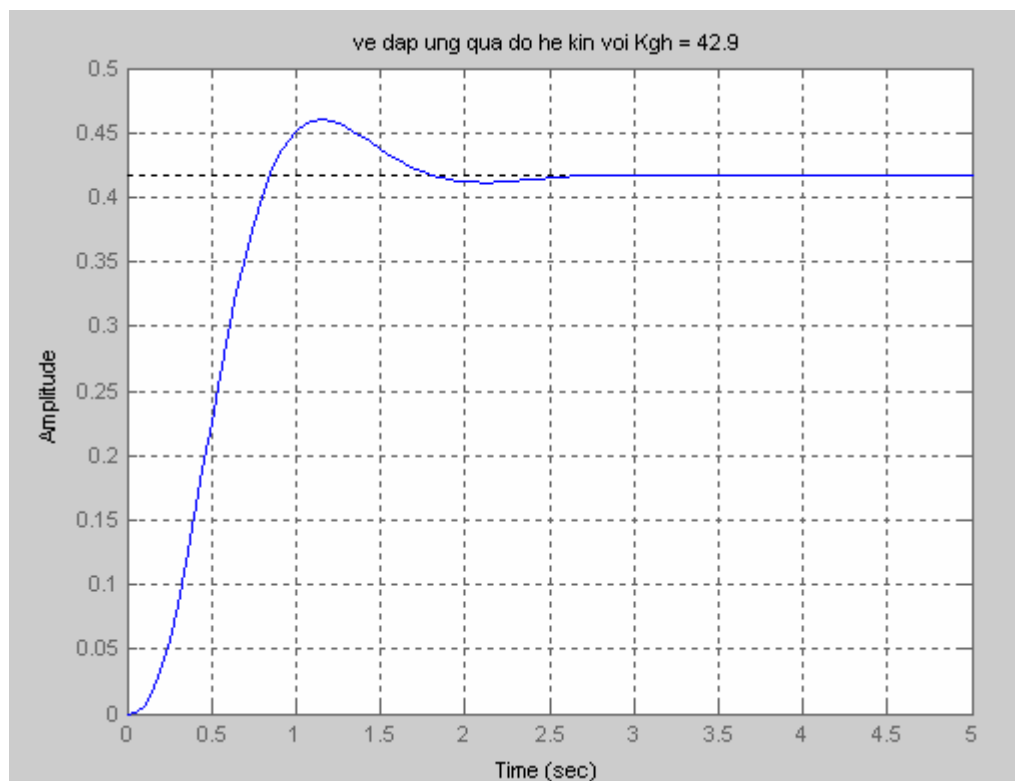
```
-----  
s^3 + 11 s^2 + 44 s + 60  
>> Gk= feedback(42.9*G,1)
```

Transfer function:
42.9

```
-----  
s^3 + 11 s^2 + 44 s + 102.9
```

```
>> step(Gk,5)  
>> grid on
```

Kết quả có được: ta thấy đáp ứng ngõ ra không dao động.



b. Với giá trị K tìm được ở câu d. phần **III.1.**, vẽ đáp ứng quá độ của hệ thống vòng kín với đầu vào hàm nấc đơn vị trong khoảng thời gian $t = 0 \div 5s$. Từ hình vẽ, tìm độ vọt lố và sai số xác lập của hệ thống. Kiểm chứng lại hệ thống có $POT = 25\%$ không? Lưu hình vẽ này để viết báo cáo.



HD:

```
>> TS = 1;
>> MS = conv([1 3],[1 8 20]);
>> G = tf(TS,MS)
```

Transfer function:

1

 $s^3 + 11s^2 + 44s + 60$

```
>> Gk= feedback(77.9*G,1)
```

Transfer function:

77.9

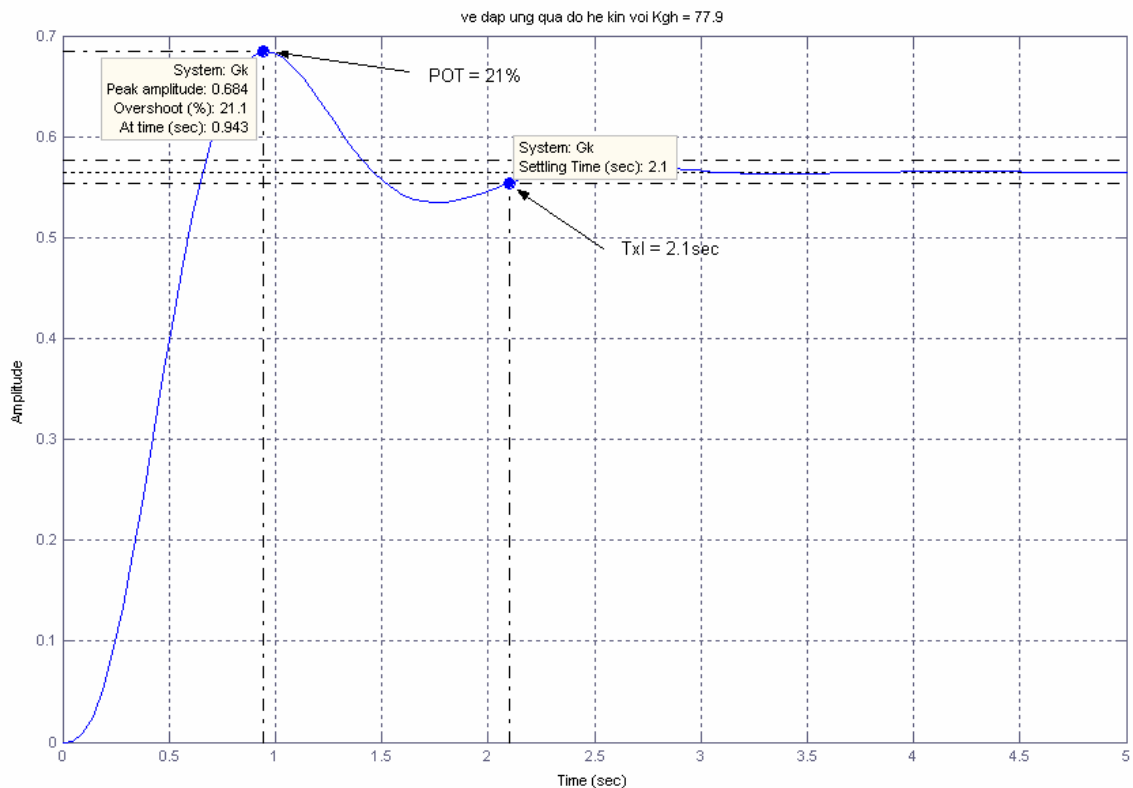
 $s^3 + 11s^2 + 44s + 137.9$

```
>> step(Gk,5)
```

```
>> grid on
```

```
>> title('ve dap ung qua do he kin voi Kgh = 77.9')
```

Kết quả có được: POT = 21% < 25%





c. Với giá trị K tìm được ở câu e. phần **III.1**, vẽ đáp ứng quá độ của hệ thống vòng kín với đầu vào hàm nấc đơn vị trong khoảng thời gian $t = 0 \div 5s$. Từ hình vẽ, tìm độ vọt lố và sai số xác lập của hệ thống. Kiểm chứng lại hệ thống có $t_{xl} = 4s$ không? Lưu hình vẽ này để viết báo cáo.

HD:

```
>> TS = 1;  
>> MS = conv([1 3],[1 8 20]);  
>> G = tf(TS,MS)
```

Transfer function:

1

 $s^3 + 11 s^2 + 44 s + 60$

```
>> Gk= feedback(185*G,1)
```

Transfer function:

185

 $s^3 + 11 s^2 + 44 s + 245$

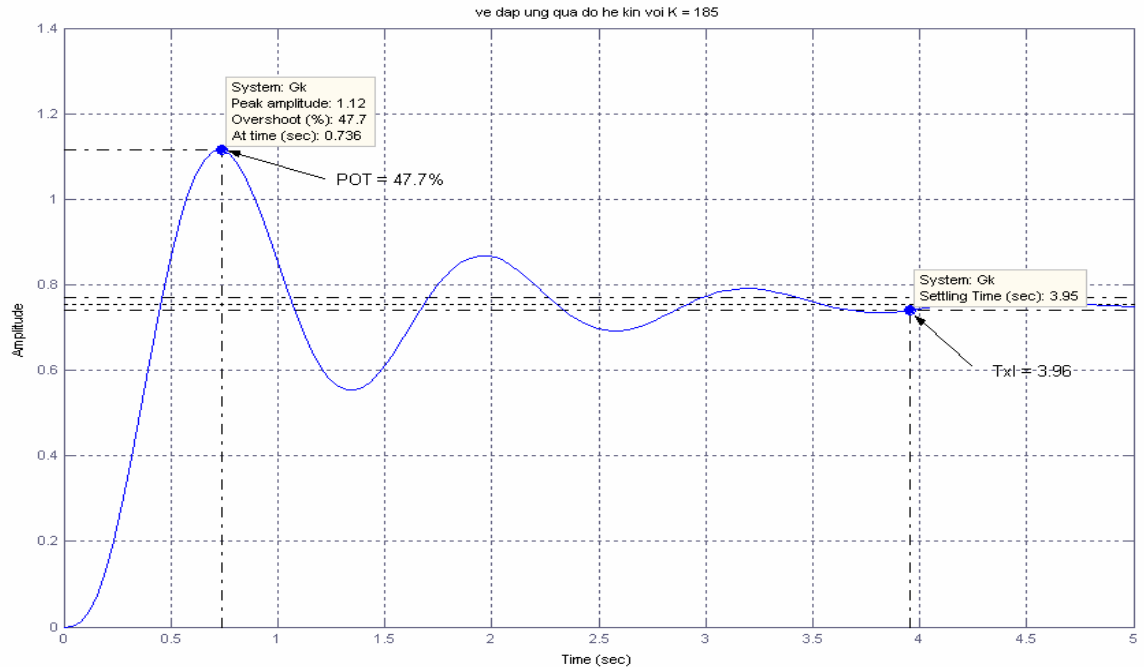
```
>> step(Gk,5)
```

```
>> grid on
```

```
>> title('ve dap ung qua do he kin voi K = 185')
```

Kết quả có được: Ta thấy $T_{xl} = 3.96 \approx 4sec$





d. Vẽ 2 đáp ứng quá độ ở câu b. và c. trên cùng 1 hình vẽ. Chú thích trên hình vẽ đáp ứng nào là tương ứng với K đó. Lưu hình vẽ này để viết báo cáo.

HD:

```
>> TS = 1;
```

```
>> MS = conv([1 3],[1 8 20]);
```

```
>> G = tf(TS,MS)
```

% tạo hàm truyền G

Transfer function:

1

 $s^3 + 11s^2 + 44s + 60$

```
>> Gk1 = feedback(77.9*G,1)
```

% tạo hàm truyền hệ kín Gk1

Transfer function:

77.9

 $s^3 + 11s^2 + 44s + 137.9$

```
>> Gk2 = feedback(185*G,1)
```

% tạo hàm truyền hệ kín Gk2





Transfer function:

185

$$s^3 + 11s^2 + 44s + 245$$

```
>> step(Gk1,5) % vẽ đáp ứng quá độ hệ kín 1
```

```
>> grid on % tạo kẻ lưới
```

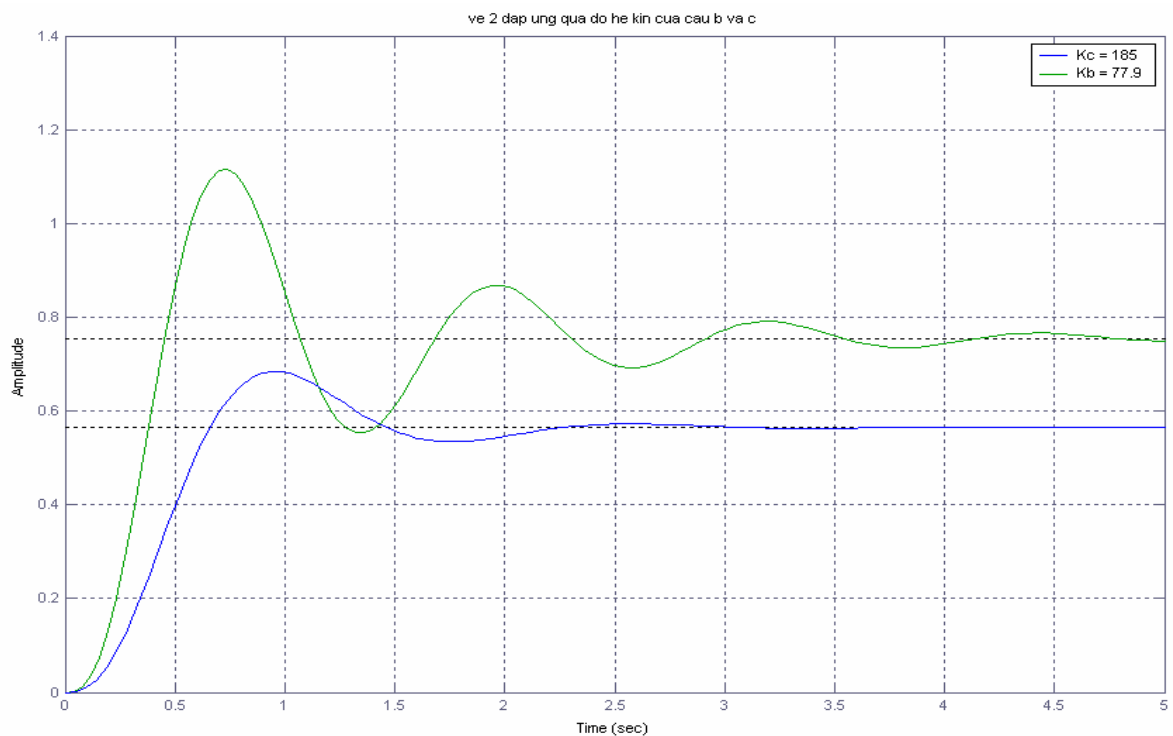
```
>> hold on % giữ lại đồ thị vẽ trước
```

```
>> step(Gk2,5) % vẽ đáp ứng quá độ hệ kín 2
```

```
>> title('ve 2 dap ung qua do he kin cua cau b va c')
```

```
>> legend('Kc = 185','Kb = 77.9') % tạo chú thích cho đáp ứng
```

Kết quả có được :



----- HẾT BÀI 2 -----



BÀI THÍ NGHIỆM 3

ỨNG DỤNG SIMULINK MÔ PHỎNG VÀ ĐÁNH GIÁ CHẤT LƯỢNG HỆ THỐNG

Câu .1. Khảo sát mô hình hệ thống điều khiển nhiệt độ:

Câu.1.a. Khảo sát hệ hở, nhận dạng hệ thống theo mô hình Ziegler-Nichols:

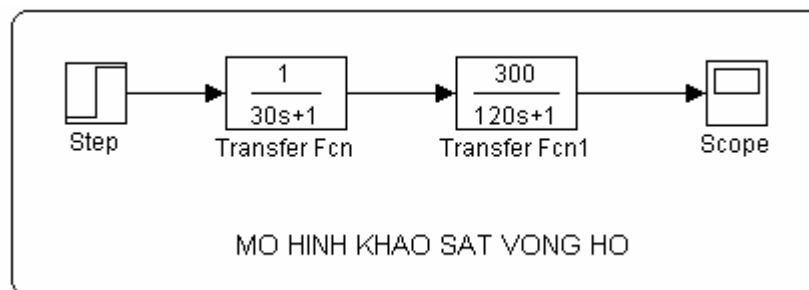
■ Mục đích:

Đặc trưng của lò nhiệt là khâu quán tính nhiệt. Từ khi bắt đầu cung cấp năng lượng đầu vào cho lò nhiệt, nhiệt độ của lò bắt đầu tăng lên từ từ. Để nhiệt độ lò đạt tới giá trị nhiệt độ cần nung thì thường phải mất một khoảng thời gian khá dài. Đây chính là đặc tính quán tính của lò nhiệt. Khi tuyến tính hoá mô hình lò nhiệt, ta xem hàm truyền của lò nhiệt như là một khâu quán tính bậc 2 hoặc như là một khâu quán tính bậc nhất nối tiếp với khâu trễ. Trong bài thí nghiệm này ta xem mô hình lò nhiệt như là một khâu quán tính bậc 2.

Trong phần này, sinh viên sẽ khảo sát khâu quán tính bậc 2 cho trước. Dùng phương pháp Ziegler-Nichols nhận dạng hệ thống sau đó xây dựng lại hàm truyền. So sánh giá trị các thông số trong hàm truyền vừa tìm được với khâu quán tính bậc 2 cho trước này.

■ Thí nghiệm:

Dùng **SIMULINK** xây dựng mô hình hệ thống lò nhiệt vòng hở như sau:



Step : là tín hiệu hàm nấc thể hiện phần trăm công suất cung cấp cho lò nhiệt.

Giá trị của hàm nấc từ $0 \rightarrow 1$ tương ứng công suất cung cấp $0\% \rightarrow 100\%$

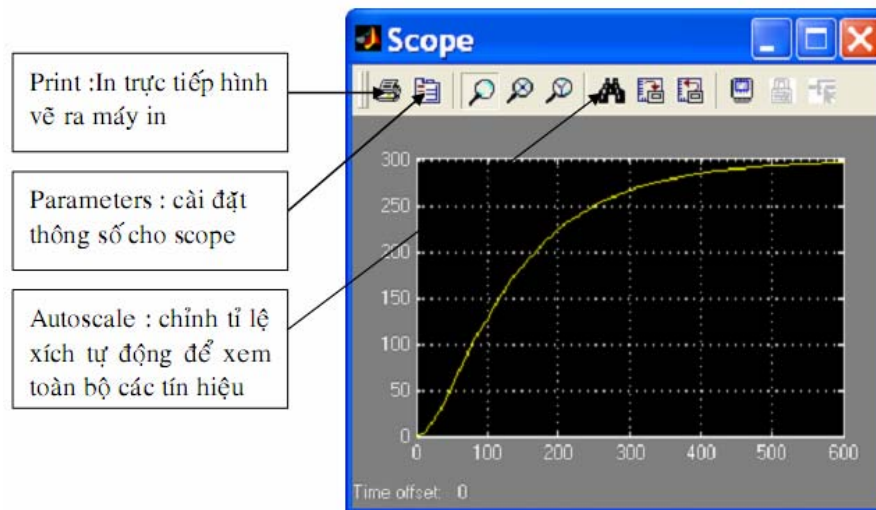
Transfer Fcn - Transfer Fcn1 : mô hình lò nhiệt tuyến tính hóa.

Chỉn giá trị của hàm nấc bằng 1 để công suất cung cấp cho lò là 100% (Step time = 0, Initial time = 0, Final time = 1). Chỉn thời gian mô phỏng Stop time = 600s. Mô phỏng và vẽ quá trình quá độ của hệ thống trên.

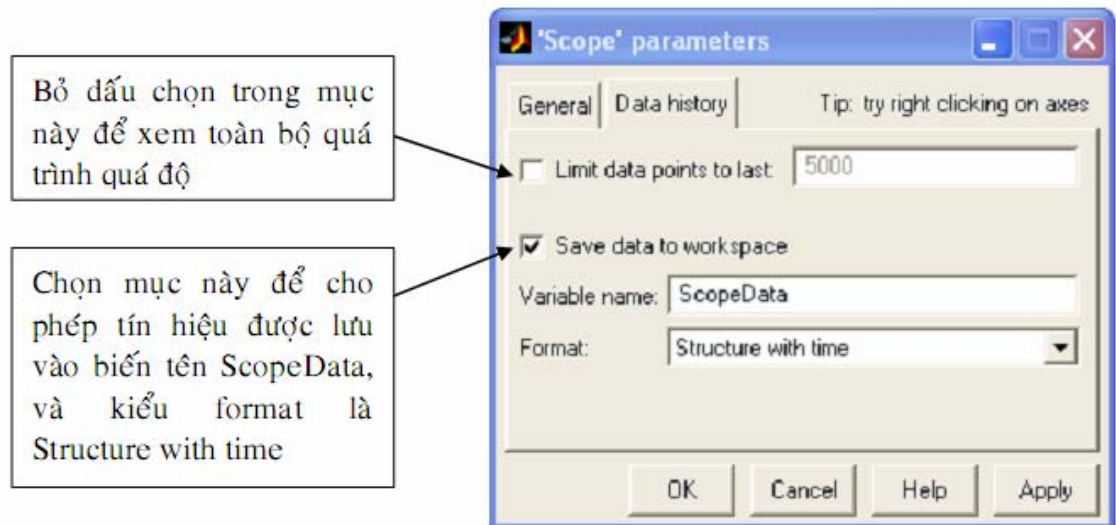
■ Hướng dẫn:

Sau khi chạy xong mô phỏng, để xem quá trình quá độ của tín hiệu ta double click

vào khối Scope. Cửa sổ Scope hiện ra như sau:



Vì cửa sổ Scope chỉ có thể xem đáp ứng hoặc in trực tiếp ra máy in nhưng không lưu hình vẽ thành file *.bmp được nên ta phải chuyển **Scope** này sang cửa sổ **Figure** để lưu. Thực hiện điều này bằng cách nhấp chuột vào ô **Parameters**. Cửa sổ **Parameters** hiện ra, nhấp chuột vào trang **Data history** và tiến hành cài đặt các thông số như hình bên dưới:

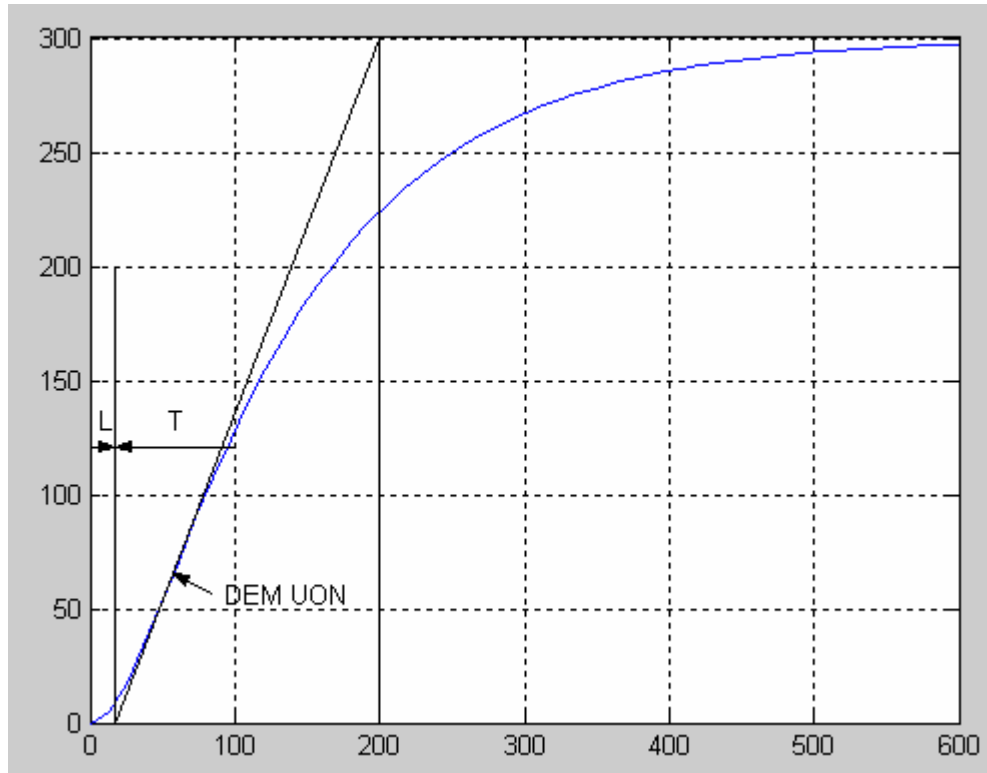


Tiến hành chạy mô phỏng lại để tín hiệu lưu vào biến ScopeData. Chú ý nếu sau khi khai báo mà không tiến hành chạy mô phỏng lại thì tín hiệu sẽ không lưu vào biến ScopeData mặc dù trên cửa sổ Scope vẫn có hình vẽ. Sau đó, vào cửa sổ **Command Window** nhập lệnh sau:

```
>> plot(ScopeData.time,ScopeData.signals.values)
>> grid on
```

%vẽ đáp ứng
%ke lưới

Lúc này cửa sổ **Figure** hiện ra với hình vẽ giống như hình vẽ ở cửa sổ **Scope**. Vào menu **Insert/ Line**, **Insert/ Text** để tiến hành kẻ tiếp tuyến và chú thích cho hình vẽ. Kết quả cuối cùng như hình bên dưới :



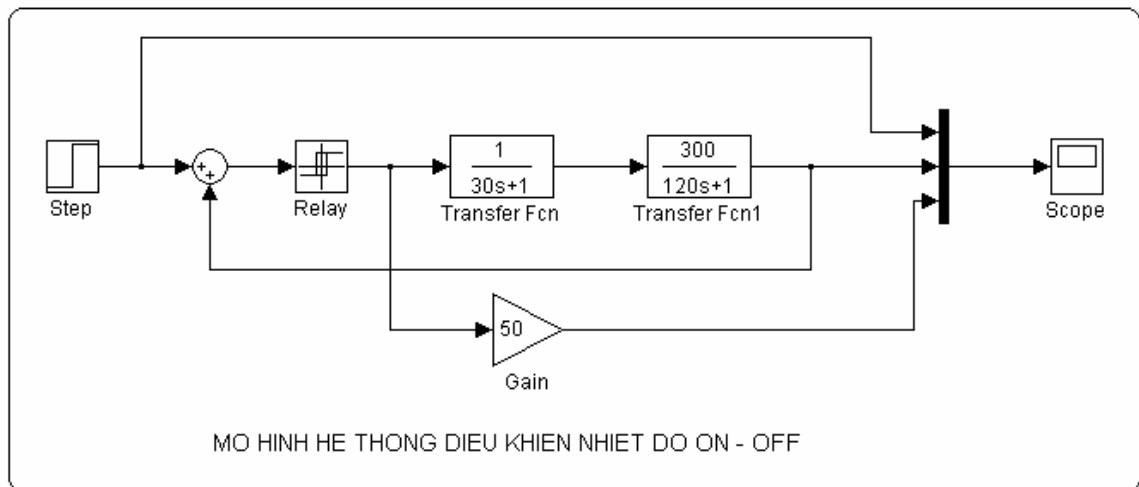
Vào menu [File]→[Export] để lưu thành file *.bmp như ở **Bài thí nghiệm 1**.

Câu .2. Khảo sát mô hình điều khiển nhiệt độ ON-OFF:**■ Mục đích:**

Khảo sát mô hình điều khiển nhiệt độ ON-OFF, xét ảnh hưởng của khâu rơle có trễ.

■ Thí nghiệm:

Xây dựng mô hình hệ thống điều khiển nhiệt độ ON-OFF như sau:



Trong đó:

- Tín hiệu đặt đầu vào hàm nấc $u(t) = 100$ (nhiệt độ đặt 100°C)
- Khối **Relay** là bộ điều khiển ON-OFF.
- Giá trị độ lợi ở khối Gain = 50 dùng để khuếch đại tín hiệu ngõ ra khối **Relay** để quan sát cho rõ. Lưu ý rằng giá trị này không làm thay đổi cấu trúc của hệ thống mà chỉ hỗ trợ việc quan sát tín hiệu.

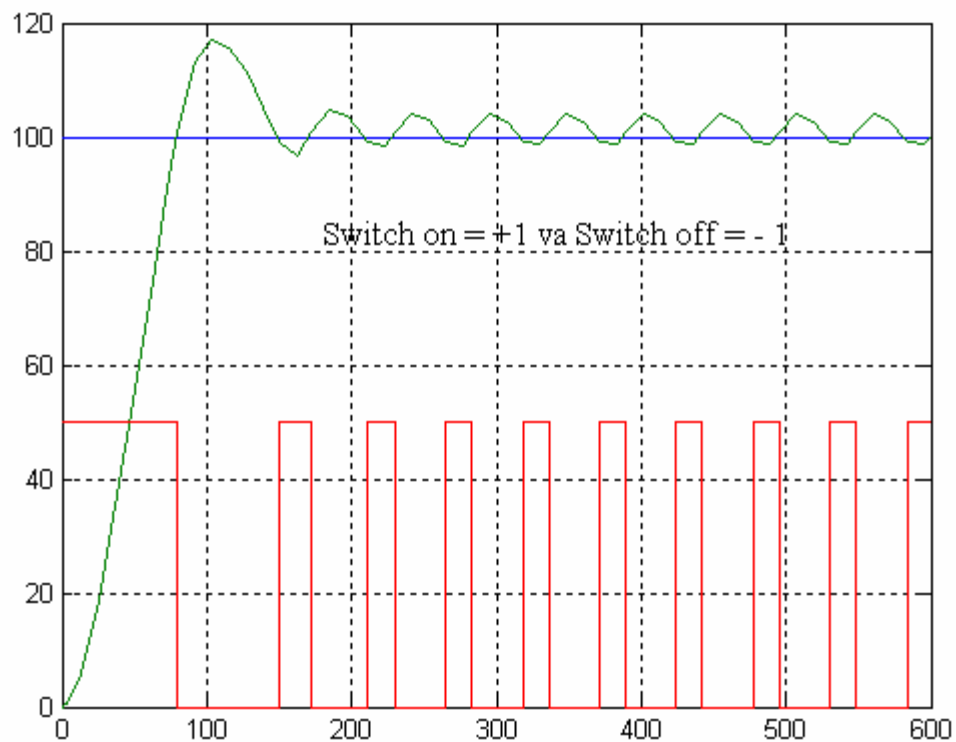
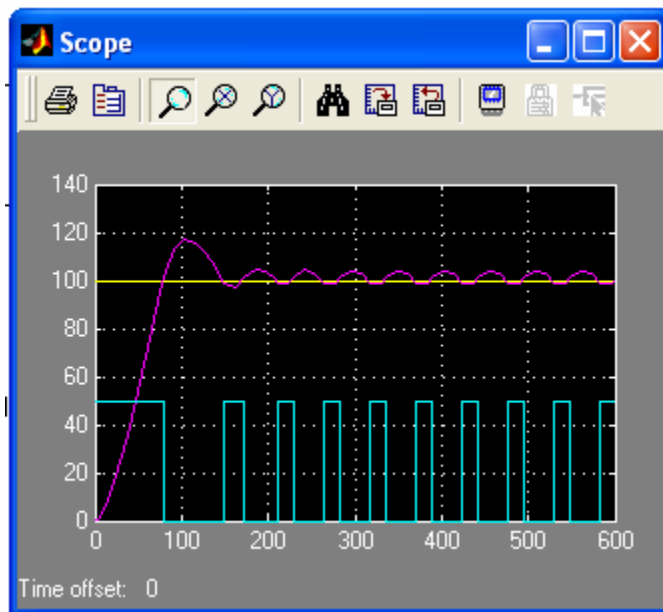
a. Chính thời gian mô phỏng **Stop time** = 600s để quan sát được 5 chu kỳ điều khiển. Khảo sát quá trình quá độ của hệ thống với các giá trị của khâu Relay theo bảng sau:



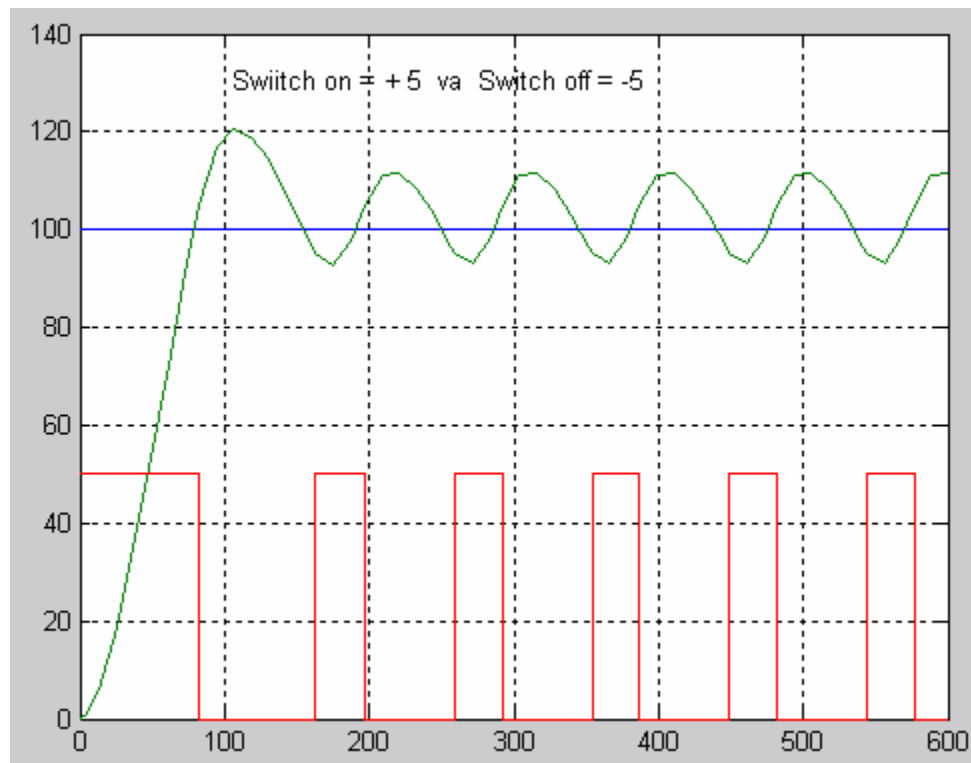
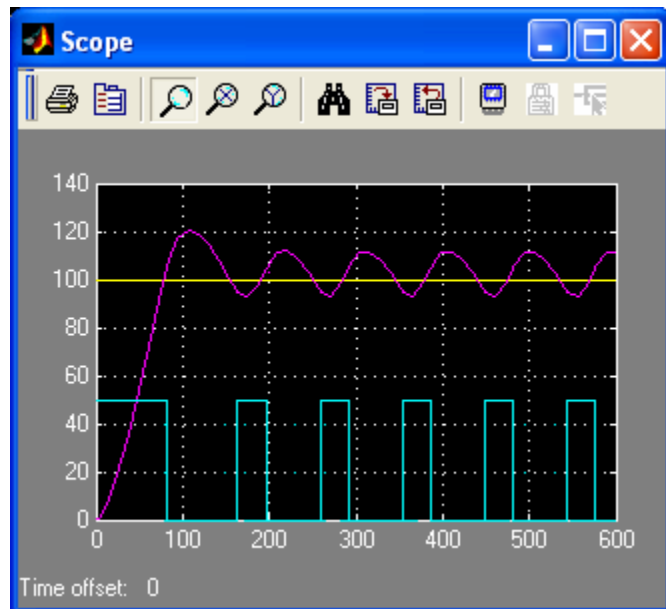
Vùng trễ (Switch on /off point)	Ngõ ra cao (Output when on)	Ngõ ra thấp (Output when off)
+1 / -1	1 (công suất 100%)	0 (công suất 0%)
+5 / -5	1 (công suất 100%)	0 (công suất 0%)
+10 / -10	1 (công suất 100%)	0 (công suất 0%)
+20 / -20	1 (công suất 100%)	0 (công suất 0%)



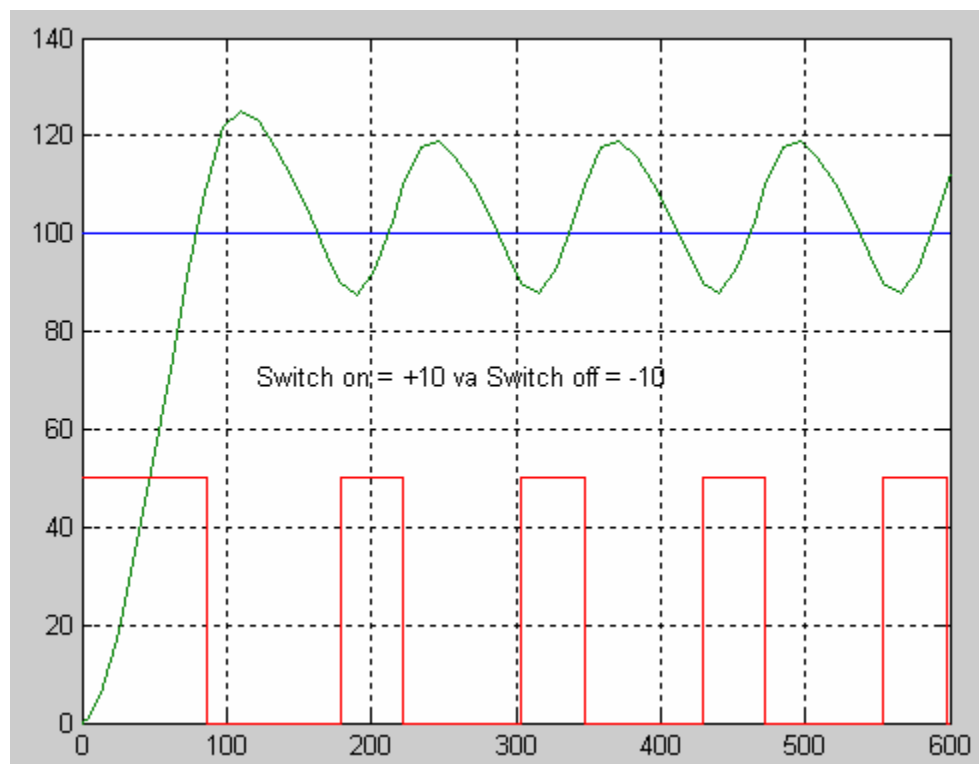
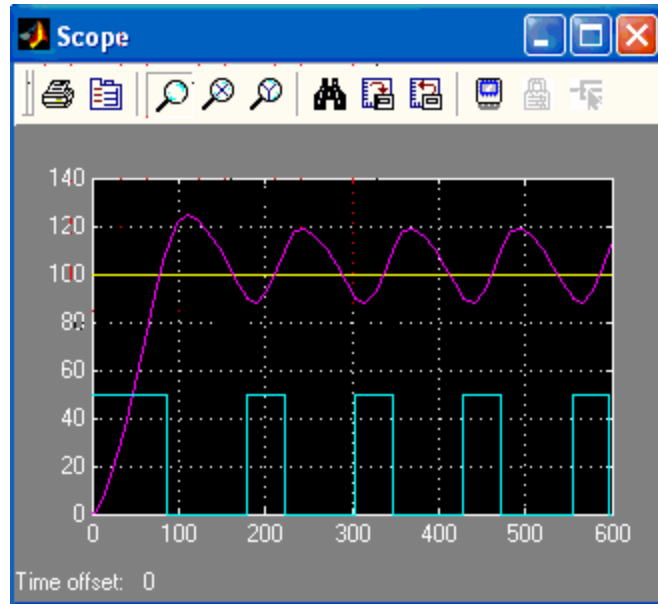
➤ Với Switch on = +1 , Switch off = -1 ta có kết quả sau:



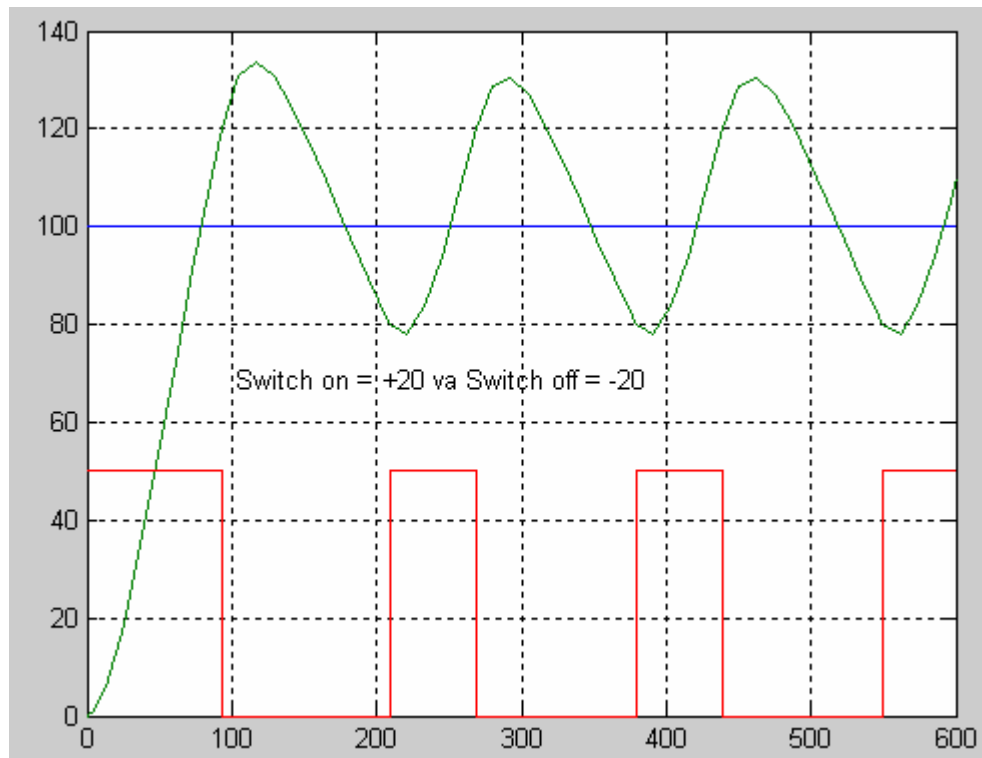
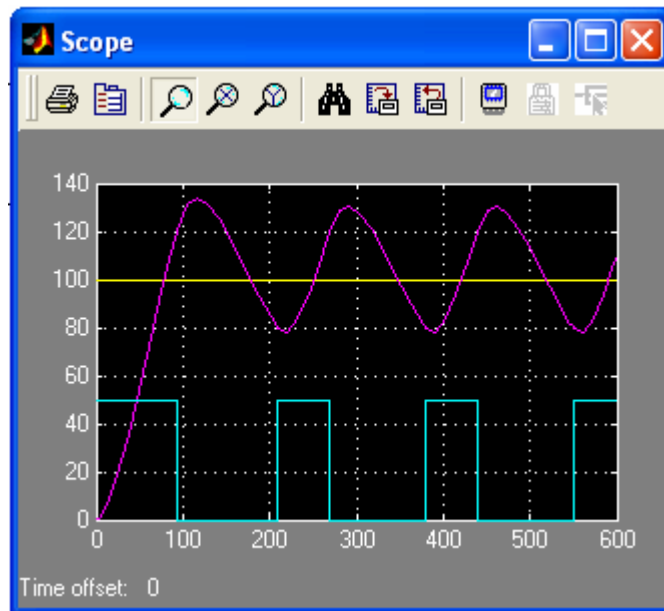
➤ Với Switch on = +5 , Switch off = -5 :



➤ Với Switch on = +10 , Switch off = -10, ta có kết quả sau :



➤ Với Switch on = +20 , Switch off = -20, ta có kết quả sau :



b. Tính sai số ngõ ra so với tín hiệu đặt và thời gian đóng ngắt ứng với các trường hợp của khâu **Relay** ở câu a theo bảng sau:



■ Hướng dẫn:

Khi điều khiển ON-OFF, ngõ ra của hệ thống có dạng dao động quanh giá trị đặt, sai

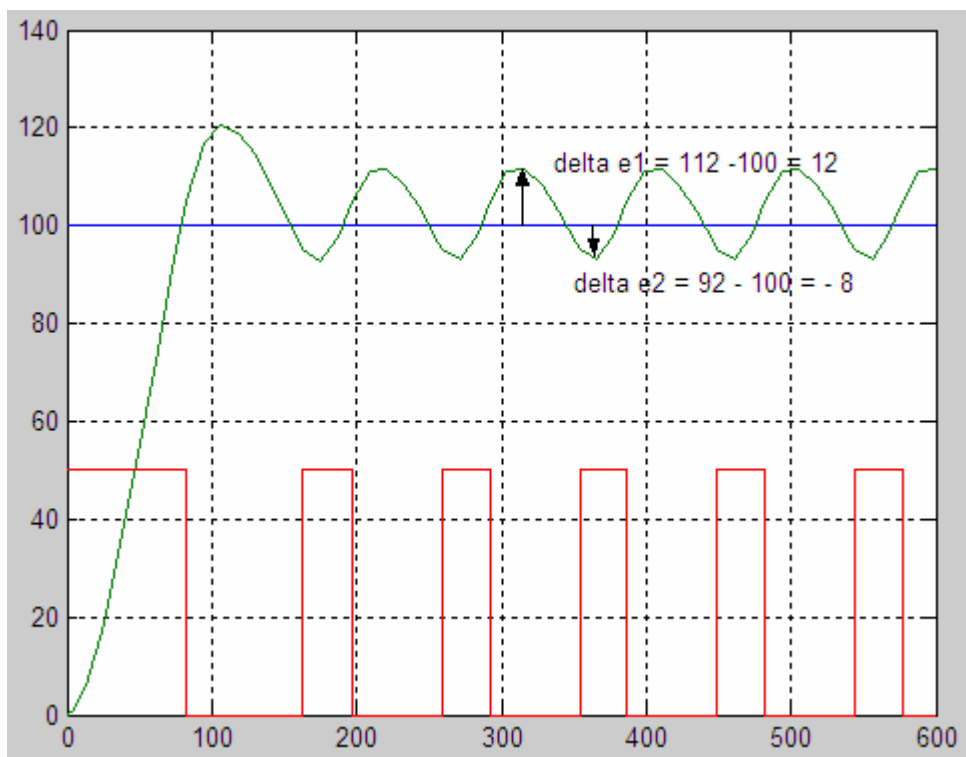
số của nó được đánh giá qua biên độ của sai lệch nhiệt độ: $\Delta e = \text{phan hoi}$ - Đặt khi hệ thống có dao động ổn định. Báo cáo hai giá trị sai lệch dương $\Delta e1$ (lớn hơn) và âm $-\Delta e2$ (nhỏ hơn) so với tín hiệu đặt. Giá trị vùng trễ phải lựa chọn sao cho dung hòa sai số ngõ ra và chu kỳ đóng ngắt. Nếu vùng trễ nhỏ thì sai số ngõ ra nhỏ nhưng chu kỳ đóng ngắt sẽ tăng lên làm giảm tuổi thọ bộ điều khiển ON-OFF.

Vùng trễ	$\Delta e1$	$-\Delta e2$	Chu kỳ đóng ngắt(s)
+1 / -1	4	2	50s
+5 / -5	12	8	≈ 100 s
+10 / -10	18	12	120s
+20 / -20	30	21	175s

Nhận xét sự ảnh hưởng của vùng trễ đến sai số ngõ ra và chu kỳ đóng ngắt của khâu **Relay** (khoảng thời gian ngõ ra khâu **Relay** thay đổi 1 chu kỳ). Vùng trễ càng nhỏ thì sai số xác lập ở ngõ ra càng nhỏ. Nhưng chu kỳ đóng ngắt sẽ nhiều (vì thời gian đóng ngắt của một chu kỳ nhỏ).

c. Lưu quá trình quá độ của trường hợp vùng trễ (+5 / -5) để viết báo cáo. Trên hình vẽ chỉ rõ 2 sai số $+\Delta e1$ / $-\Delta e2$ quanh giá trị đặt và chu kỳ đóng ngắt.





Câu 3. Khảo sát mô hình điều khiển nhiệt độ dùng phương pháp Ziegler-Nichols (điều khiển PID):

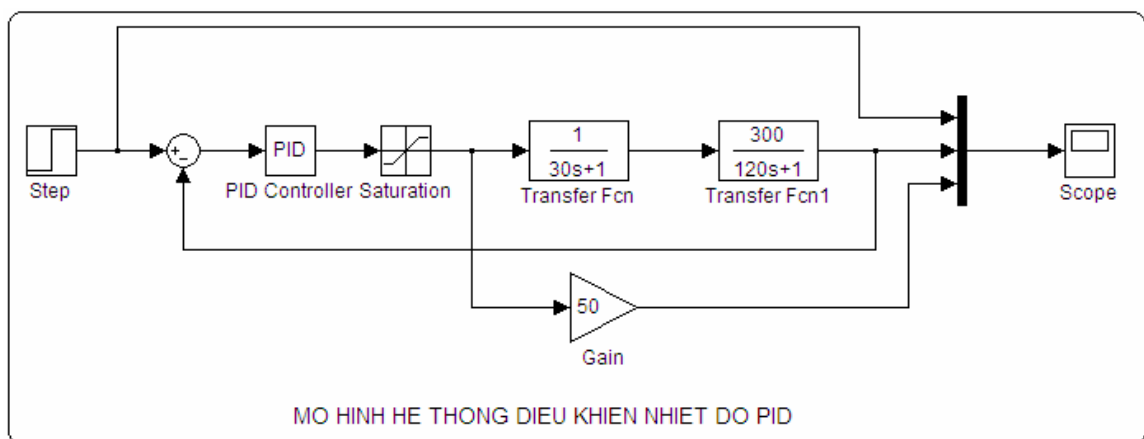
■ Mục đích:

Khảo sát mô hình điều khiển nhiệt độ dùng bộ điều khiển PID, các thông số của bộ

PID được tính theo phương pháp Ziegler-Nichols. Từ đó so sánh chất lượng của hệ thống ở 2 bộ điều khiển PID với bộ điều khiển ON-OFF.

■ Thí nghiệm:

Xây dựng mô hình hệ thống điều khiển nhiệt độ PID như sau:



Trong đó:

- Tín hiệu đặt đầu vào hàm nấc $u(t) = 100$ (tượng trưng nhiệt độ đặt 100°C)
- Khâu bảo hòa **Saturation** có giới hạn là **upper limit** = 1, **lower limit** = 0 (tượng trưng ngõ ra bộ điều khiển có công suất cung cấp từ 0% đến 100%).
- Bộ điều khiển PID có các thông số cần tính toán.
- **Transfer Fcn – Transfer Fcn1** : mô hình lò nhiệt tuyến tính hóa.

a. Tính giá trị các thông số K_P , K_I , K_D của khâu PID theo phương pháp Ziegler-Nichols từ thông số L và T tìm được ở phần câu 1.a

■ Hướng dẫn:

Cách tính các thông số K_P , K_I , K_D của khâu PID theo phương pháp Ziegler-Nichols như sau:

$$PID(s) = K_P + \frac{K_I}{s} + K_D \quad \text{với} \quad K_P = \frac{1.2 * T}{L * K} = \frac{1.2 * 175}{25 * 175} = 0,028$$

$$K_I = \frac{K_P}{2 * L} = \frac{0,028}{2 * 25} = 5,6 * 10^{-6} \quad K_D = 0,5 K_P * L = 0,5 * 0,028 * 25 = 0,35$$

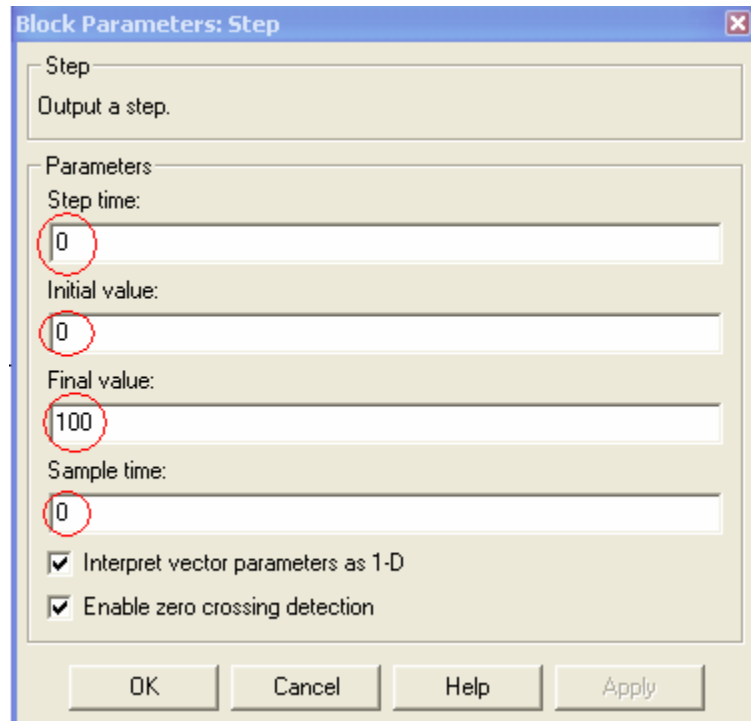
Trong đó $L = 25$, $T = 175$, K là các giá trị đã tìm được ở phần **câu.1.a**. Chú ý,

giá trị K đã cho trước ở mô hình hàm truyền lò nhiệt $K = 300$.

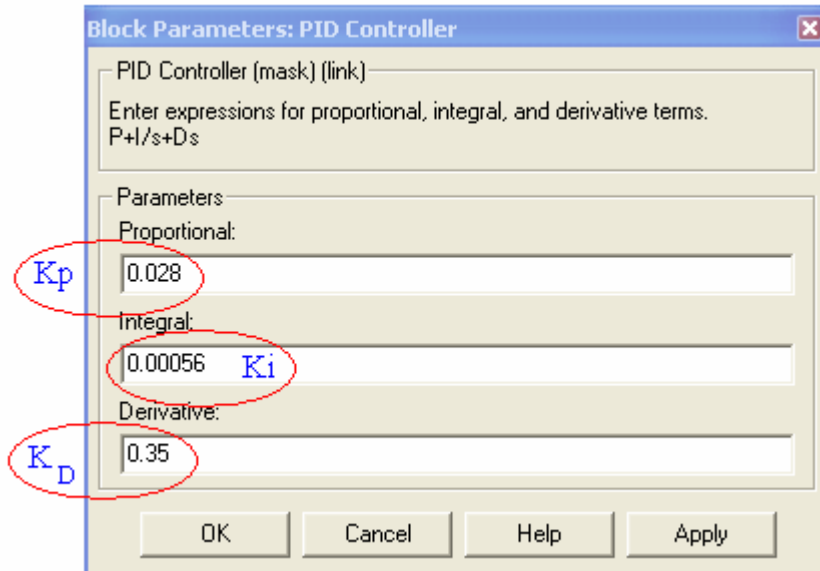
b. Chạy mô phỏng và lưu đáp ứng của các tín hiệu ở Scope để viết báo cáo. Có thể chọn lại **Stop time** cho phù hợp. Trong hình vẽ phải chú thích rõ tên các tín hiệu.

■ **Hướng dẫn:**

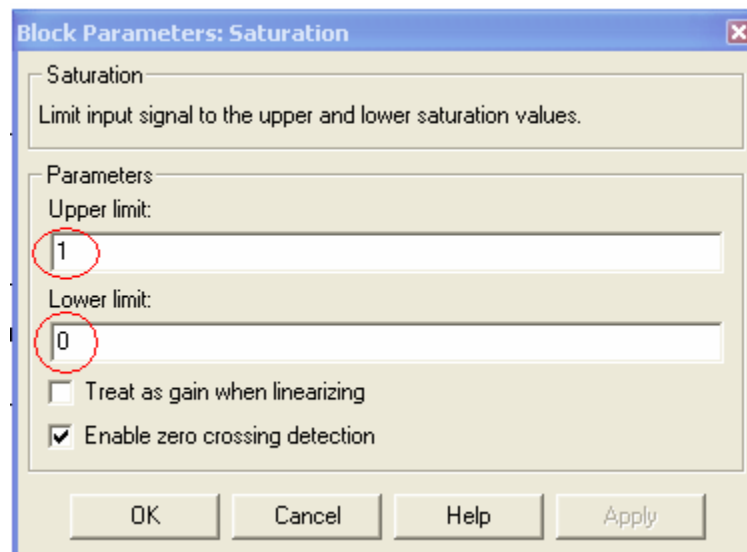
Tín hiệu đặt đầu vào hàm nấc $u(t) = 100$ (tương trưng nhiệt độ đặt 100°C):



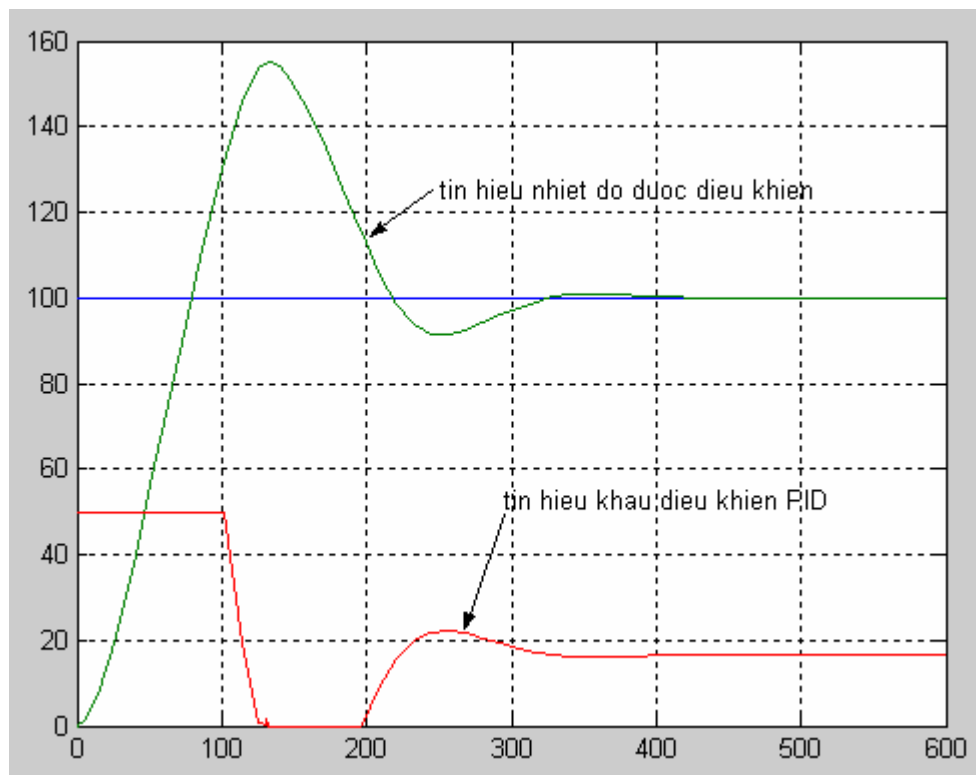
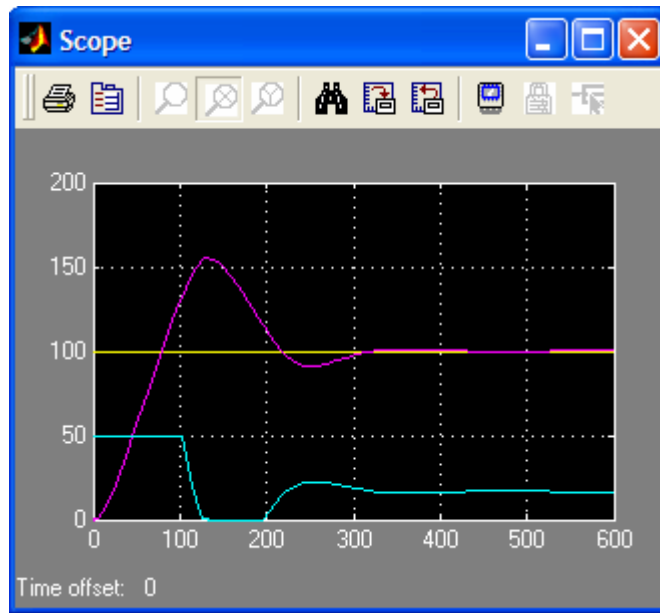
Sau đó ta nhập các thông số cho bộ điều khiển PID như sau:



Tiếp theo nhập khâu bão hòa **Saturation** có giới hạn là **upper limit = 1, lower limit = 0**



Chạy mô phỏng ta được kết quả



c. Nhận xét về chất lượng ngõ ra ở 2 phương pháp điều khiển PID và ON-OFF.

➤ Đối với phương pháp điều khiển ON – OFF thì độ vọt lố ở ngõ ra có giảm nhưng có sai số xác lập lớn, vì hệ chỉ cân bằng động quanh giá trị nhiệt độ đặt, và thay đổi theo tải.



➤ Đối với phương pháp điều khiển PID nếu giá trị của K càng lớn thì sai số sẽ giảm nhưng sẽ xuất hiện dao động và độ vọt lố.

-----**HẾT BÀI 3**-----



BÀI THÍ NGHIỆM 5

HỆ THỐNG ĐIỀU KHIỂN NHIỆT ĐỘ

Phần 1. Điều khiển ON -OFF :


▪ Mục đích :


Khảo sát hệ thống điều khiển nhiệt độ theo nguyên tắc điều khiển ON - OFF.


▪ Thí nghiệm :

Mở cửa lò nếu lò còn nóng thì dùng quạt để hạ nhiệt độ trong lò xuống dưới 40 °C.


Thực hiện quá trình cài đặt thông số cho bộ điều khiển OMRON E5AZ



+ Phím : cho phép người sử dụng di chuyển qua lại giữa các chế độ : *Initial Setting level* (chế độ cài đặt thông số bộ điều khiển) , *Operation Level* (chế độ hoạt động) , *Adjustment Level* , *Advanced Function Setting Level* .

+ phím : di chuyển con trỏ trong 1 chế độ.


+ 2 phím : tăng giảm các giá trị .

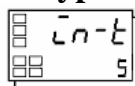
➤ *Initial Setting Level:*

Trước tiên ta phải vô chế độ cài đặt *Initial Setting Level* : ấn và giữ phím  hơn 5s

Ấn phím  để lựa chọn thông số ,  tăng giảm các thông số .

▪ **Input type** : cảm biến ngõ vào (mặc định là 5) :

Press  keys to select input type.




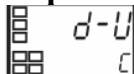
Input type 5

do ta sử dụng cảm biến có thông số là (K)

Input type	K	J
Temperature range	-200 to 1300°C	-20 to 850°C
Setting number	5	7



▪ **Temperature Unit** : đơn vị đo nhiệt độ là °C

Press  keys to select input type.





Temperature unit

- **Sp upper limit** : nhiệt độ giới hạn trên **100 °C**

Press  
keys to select
input type.





- **Sp lower limit** : nhiệt độ giới hạn dưới **0 °C**

Press  
keys to select
input type.



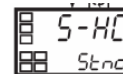
- **Select the control system** : hệ thống điều khiển

Press  
keys to select
input type.

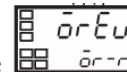


ON/OFF

- **Standard or heating/cooling** : kiểu điều khiển **Standard**





- **Direct/reverse operation** : chọn kiểu điều khiển **reverse**






- **Alarm 1 ÷ 3**: chọn 0



Press  
keys to select
input type.



➤ **Operation Level :**

Đề vào chế độ hoạt động *Operation Level* từ chế độ *Initial Setting Level* ta ấn giữ phím  hơn 1s . Sau khi vào chế độ hoạt động *Operation Level* ta Ấn   thiết lập nhiệt độ đặt là **70°C**.

Ấn phím  : -Ấn   chọn run  và ấn  cho phép hệ thống bắt đầu hoạt động.

- Ấn   chọn stop  và ấn  hệ thống tạm dừng

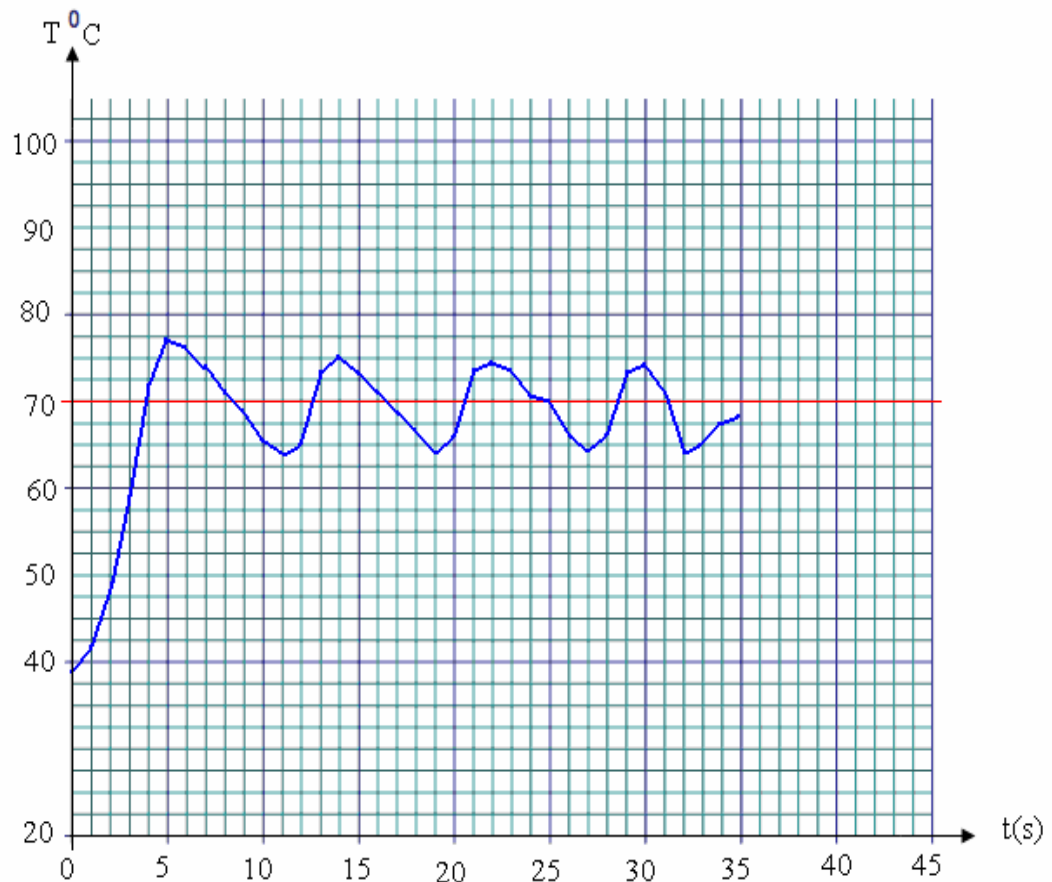
Cho hệ thống hoạt động và ghi quá trình tăng nhiệt trong 35 phút. Ta có kết quả ở Bảng 1

T(min)	0	1	2	3	4	5	6	7	8	9	10	11
--------	---	---	---	---	---	---	---	---	---	---	----	----



$T^{\circ}C$	39	42	48	59	72	77	76	74	72	69	66	64
T(min)	12	13	14	15	16	17	18	19	20	21	22	23
$T^{\circ}C$	66	73	75	73	71	69	67	64	66	73	74	73
T(min)	24	25	26	27	28	29	30	31	32	33	34	35
$T^{\circ}C$	71	70	66	64	66	73	74	71	69	66	64	67

Bảng 1 : Điều khiển ON – OFF



QTQĐ nhiệt độ lò

Từ kết quả ghi theo bảng 1 ta vẽ QTQĐ nhiệt độ lò như hình trên.

Từ hình vẽ QTQĐ:

- Xác định vùng trễ của khâu rơle (Vùng trễ của bộ điều khiển sẽ tính bằng chênh lệch nhiệt độ giữa hai lần đóng ngắt liên tiếp)

Vùng trễ = 8s

- Sai số thực tế. Sai số thực tế là $\pm 4^{\circ}C$

Phần 2 .Điều khiển tuyến tính :



▪ **Mục đích :**



Khảo sát hệ thống điều khiển tuyến tính nhiệt độ với hiệu chỉnh PID

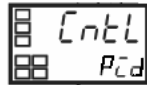
▪ **Thí nghiệm :**



Mở cửa lò để hạ nhiệt độ xuống dưới 40°C .

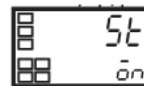
Thực hiện các bước tương tự ở chế độ ON/OFF :nhưng ta thay đổi 1 số thông số ở chế độ *Initial Setting Level* là .

▪ **Select the control system :hệ thống điều khiển PID**

Press  
keys to select
input type.





Press  
keys to select
input type.



▪ **ST : chế độ tự dò tìm ON**

▪ **control period : chu kỳ điều khiển ngõ ra 20 s**

Press  
keys to select
input type.



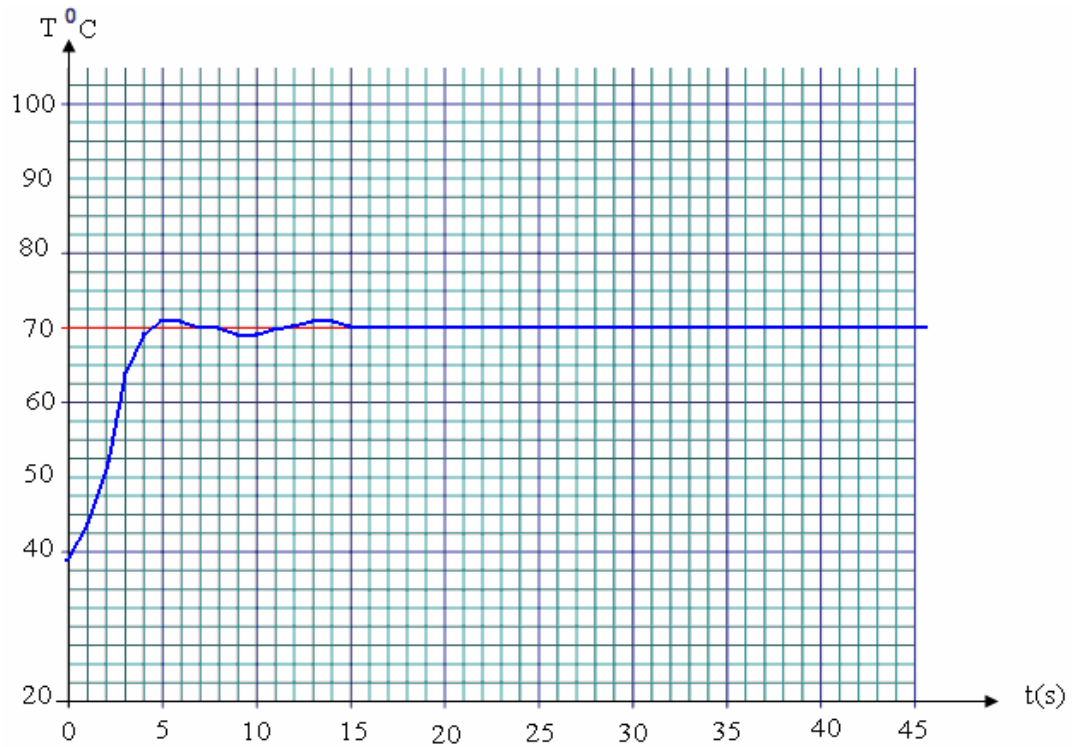
Các bước còn lại thực hiện tương tự như phần **ON/OFF**

Vẫn giữ nhiệt độ đặt là **70°C** đo như phần **III.1** trong 30 phút.

Báo cáo kết quả ghi theo Bảng 2 và vẽ QTQĐ nhiệt độ lò trong 30 phút.

T(min)	0	1	2	3	4	5	6	7	8	9	10	11
T^0C	39	43	52	63	69	71	71	70	70	69	69	70
T(min)	12	13	14	15	16	17	18	19	20	21	22	23
T^0C	70	71	71	70	70	70	70	70	70	70	70	70
T(min)	24	25	26	27	28	29	30	31	32	33	34	35
T^0C												

Bảng 2 : Điều khiển tuyến tính (PID)



QTQĐ nhiệt độ lò

Kết quả ghi theo bảng 2 và vẽ QTQĐ nhiệt độ lò như hình trên.

Từ hình vẽ QTQĐ:

- Đánh giá thời gian quá độ : thời gian quá độ

- Độ vọt lố : $POT = \frac{C_{\max} - C_{xl}}{C_{xl}} \times 100 = \frac{71 - 70}{70} \times 100 = 1,43\%$

- Sai số xác lập : $e_{xl} = 0$

- Nhận xét và so sánh về chất lượng điều khiển của bộ điều khiển tuyến tính (PID) so với bộ điều khiển ON-OFF :

Với điều khiển ON – OFF sai số xác lập lớn do hệ chỉ cân bằng động quanh nhiệt độ đặt và thay đổi theo tải.

Còn với điều khiển PID: tín hiệu điều khiển theo dạng điều khiển độ rộng xung, theo phương pháp tỉ tích phân tỉ lệ do đó mà ngõ ra hầu như không có sai số xác lập.

----- HẾT BÀI 5 -----



