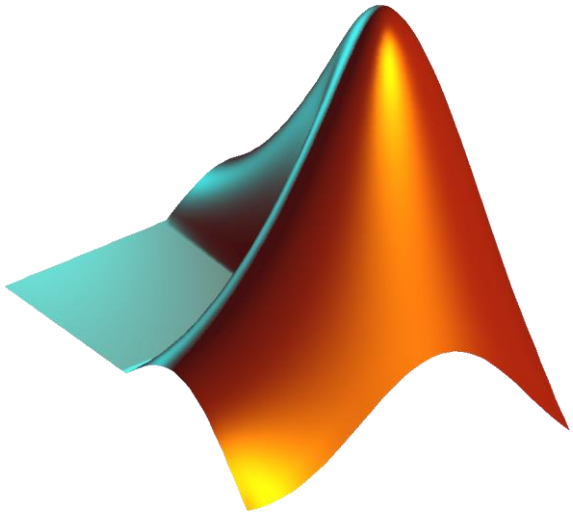
MATLAB PROJECT



Power Method
Least Squares Fitting
PID Controller & GUI

Name | Omar Ahmed Ali Hegazy **ID** | 111
Name | Mohamed Mostafa Helmy **ID** | 169
Name | Aya Saeed Ibrahim **ID** | 1
Name | Basma Mamdouh Gaber **ID** | 75
Name | Basant Islam AbdelHameed **ID** | 58
Name | Alaa Abdel-Wahab Said **ID** | 106
Name | Mohamed Magdy El-Said **ID** | 166
Name | Ahmed Mohamed Hemdan **ID** | 31

Names	Seat Numbers
اية سعيد ابراهيم محمد	١
احمد محمد حمدان محمود	٣١
بسمة ممدوح جابر حنفي	٧٥
بسنت اسلام عبدالحميد السيد	٥٨
علاء عبدالوهاب سعيد	١٠٦
عمر احمد علي حجازي	١١١
محمد مجدي السيد الورداني	١٦٦
محمد مصطفى حلمي	١٦٩

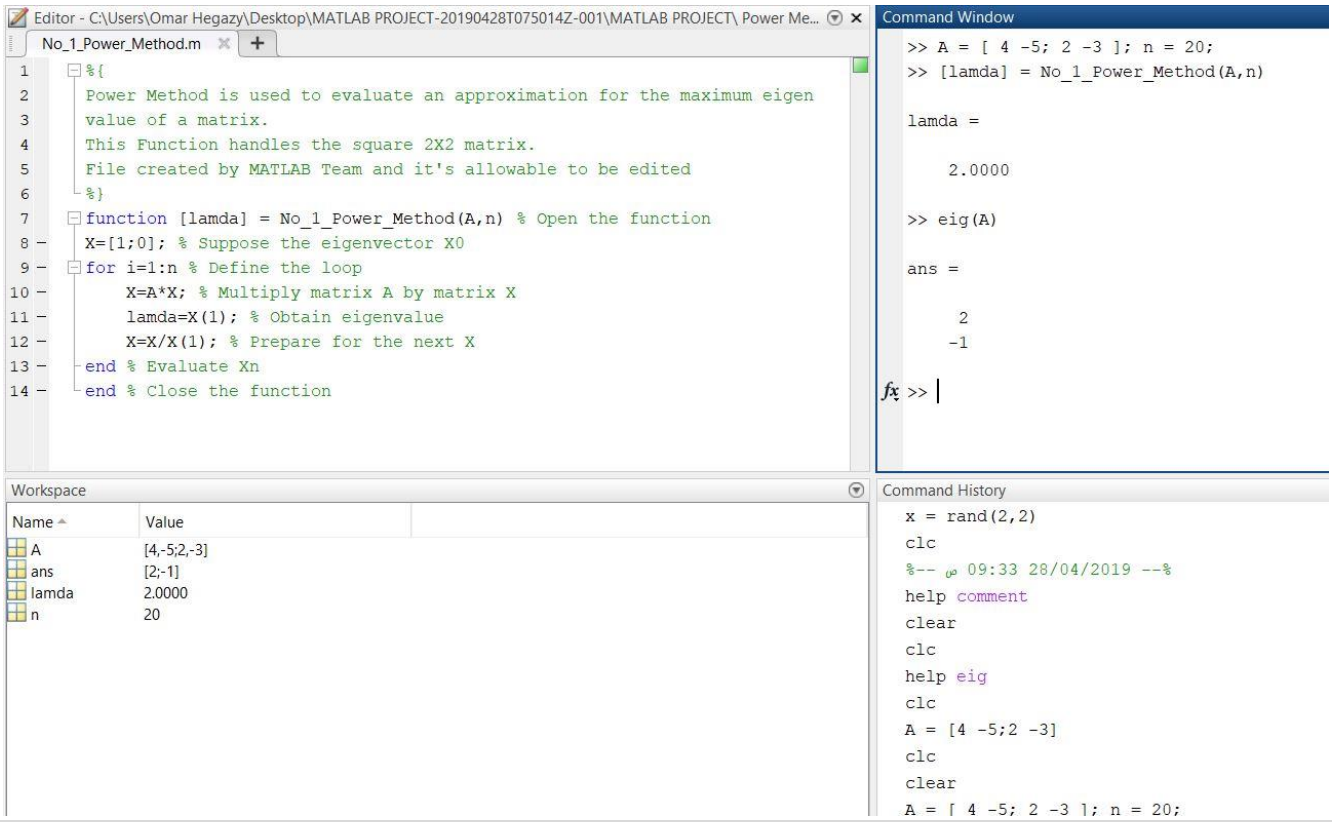
Part 1 | PROGRAMMING

A-Power Method

The Source Code:

```
%{
Power Method is used to evaluate an approximation for the
maximum eigen
value of a matrix.
This Function handles the square 2X2 matrix.
File created by MATLAB Team and it's allowable to be
edited
%}
function [lamda] = No_1_Power_Method(A,n) % Open the
function
X=[1;0]; % Suppose the eigenvector X0
for i=1:n % Define the loop
    X=A*X; % Multiply matrix A by matrix X
    lamda=X(1); % Obtain eigenvalue
    X=X/X(1); % Prepare for the next X
end % Evaluate Xn
end % Close the function
```

Code Test with eig() function:



B-Least Squares Fitting

The Source Code:

```
%{
Least Squares Method 'or Linear Least Squares Regression'
is used to fit
a dependent variable 'y for example' equal to some
functions of the
independent variables'x & u & v for example'.
the functions are : y1=a0+a1*x1+a2*u1+a3*v1
                    y2=a0+a1*x2+a2*u2+a3*v2
                    y3=a0+a1*x3+a2*u2+a3*v3
                    ........................
                    yn=a0+a1*xn+a2*un+a3*vn
in matrix form :
X = [1,x1,u1,v1 ; 1,x2,u2,v2 ; ..... ; 1,xn,un,vn]
a = [a0 ; a1 ; a2 ; a3]
Y = [y1 ; y2 ; ..... ; yn]
This Function fits y equal to a function of x only to
obtain a line
equation that has two constants (a0 & a1) in the 'a'
matrix that can be
evaluated by the equation : a = inv(X'*X)*X'*Y
File created by MATLAB Team and it's allowable to be
edited
%}
function [a0,a1]=No_2_Least_Squares_Method(x,y)
X = [ones(length(x),1) x]; %first column treated as all
ones since x_1=1
Y = y; %column vector for proper dimension during
multiplication
a = inv(X'*X)*X'*Y;  % Least Squares Estimator -
equivalent code X\y
plot ( x , y , '*'); % original data
a0=a(1);%intersection of the line
a1=a(2);%slope of the line
hold on;
plot( x , a0+a1*x , 'r-' ); %Fitted line
legend('Data',['y=' num2str(a0) '+' num2str(a1) 'x'])
xlabel('X values');
ylabel('Y values');
end
```

Code Test with Plot Diagram:

Editor - C:\Users\Omar Hegazy\Desktop\MATLAB PROJECT-20190428T075014Z-001\MATLAB PROJECT\ ...

No_1_Power_Method.m | No_2_Least_Squares_Method.m

```
1   %{...%}
19  function [a0,a1]=No_2_Least_Squares_Method(x,y)
20  X = [ones(length(x),1) x]; %first column treated as all ones since x
21  Y = y; %column vector for proper dimension during multiplication
22  a = inv(X'*X)*X'*Y;  % Least Squares Estimator - equivalent code X\y
23  plot ( x , y , '*'); % original data
24  a0=a(1);%intersection of the line
25  a1=a(2);%slope of the line
26  hold on;
27  plot( x , a0+a1*x , 'r-' ); %Fitted line
28  legend('Data',['y=' num2str(a0) '+' num2str(a1) 'x'])
29  xlabel('X values');
30  ylabel('Y values');
31  end
```

Workspace

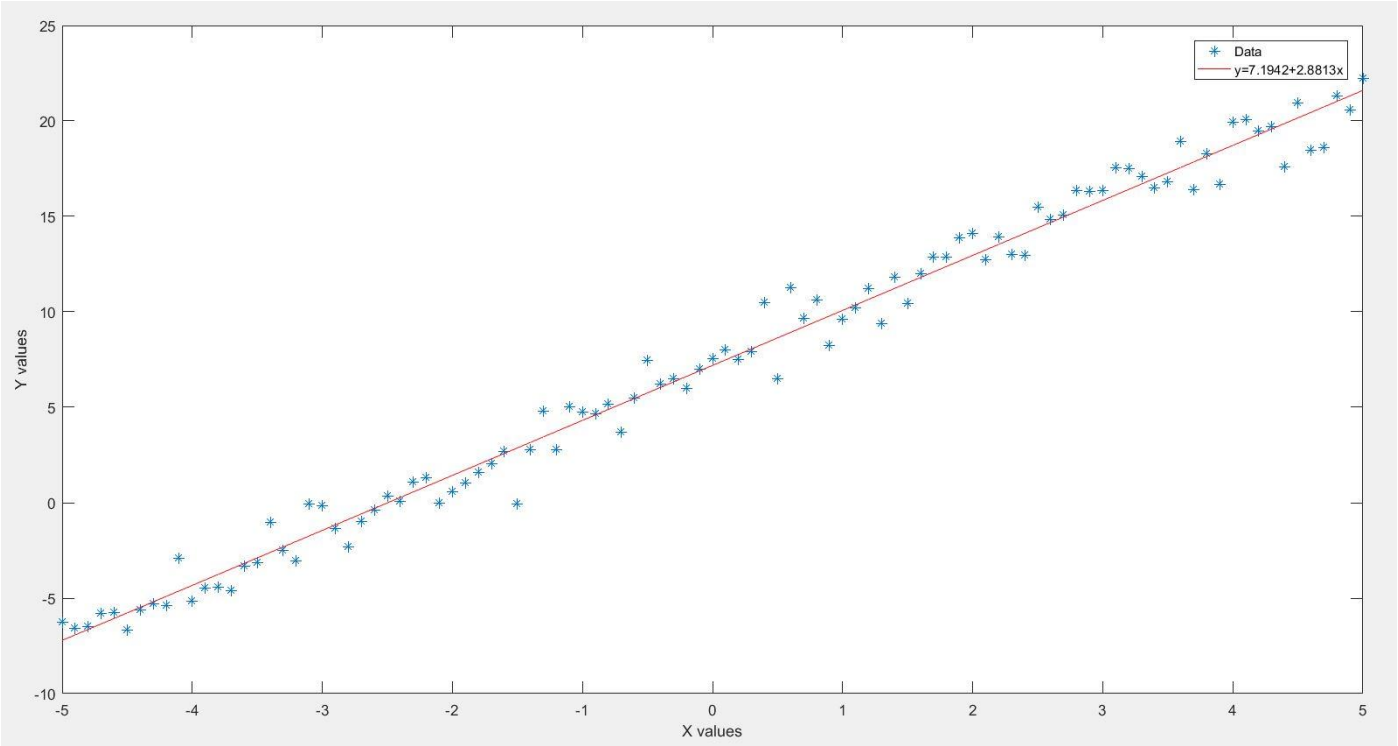
Name	Value
a0	7.1942
a1	2.8813
x	*101x1 double*
y	*101x1 double*

Command Window

```
>> x = (-5 :0.1:5)' ;
>> y = 7.3 + 2.9*x + randn(size(x));
>> [a0,a1]=No_2_Least_Squares_Method(x,y)

a0 =

    7.1942


a1 =

    2.8813

fx >>
```

Command History

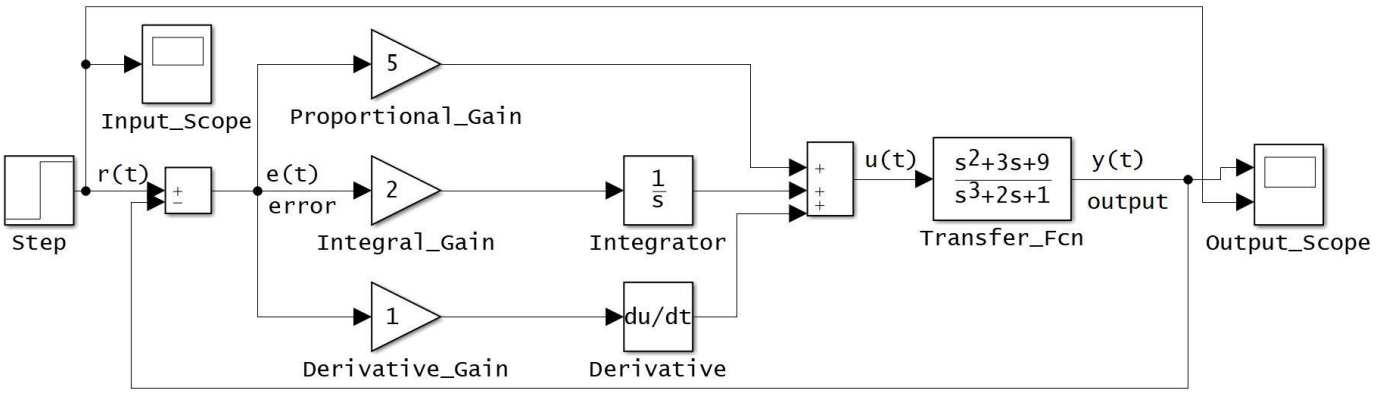
```
clc
x = -5 :0.1:5 ;
y = 7.3 + 2.9*x + randn(size(x));
No_2_Least_Squares_Method
clc
x = -5 :0.1:5 ;
y = 7.3 + 2.9*x + randn(size(x));
[a0,a1]=No_2_Least_Squares_Method(x,y)
clc
x = (-5 :0.1:5)' ;
y = 7.3 + 2.9*x + randn(size(x));
[a0,a1]=No_2_Least_Squares_Method(x,y)
```



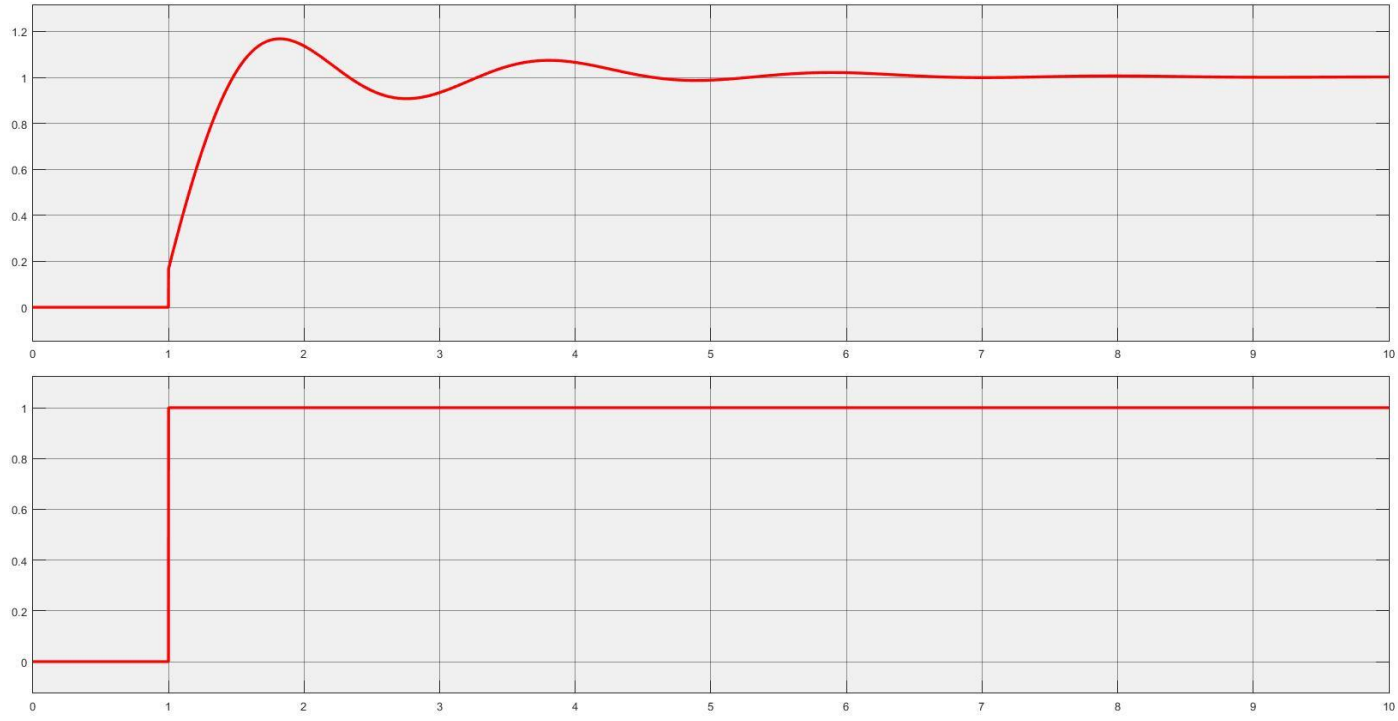
Part 2 | SIMULINK

PID Controller

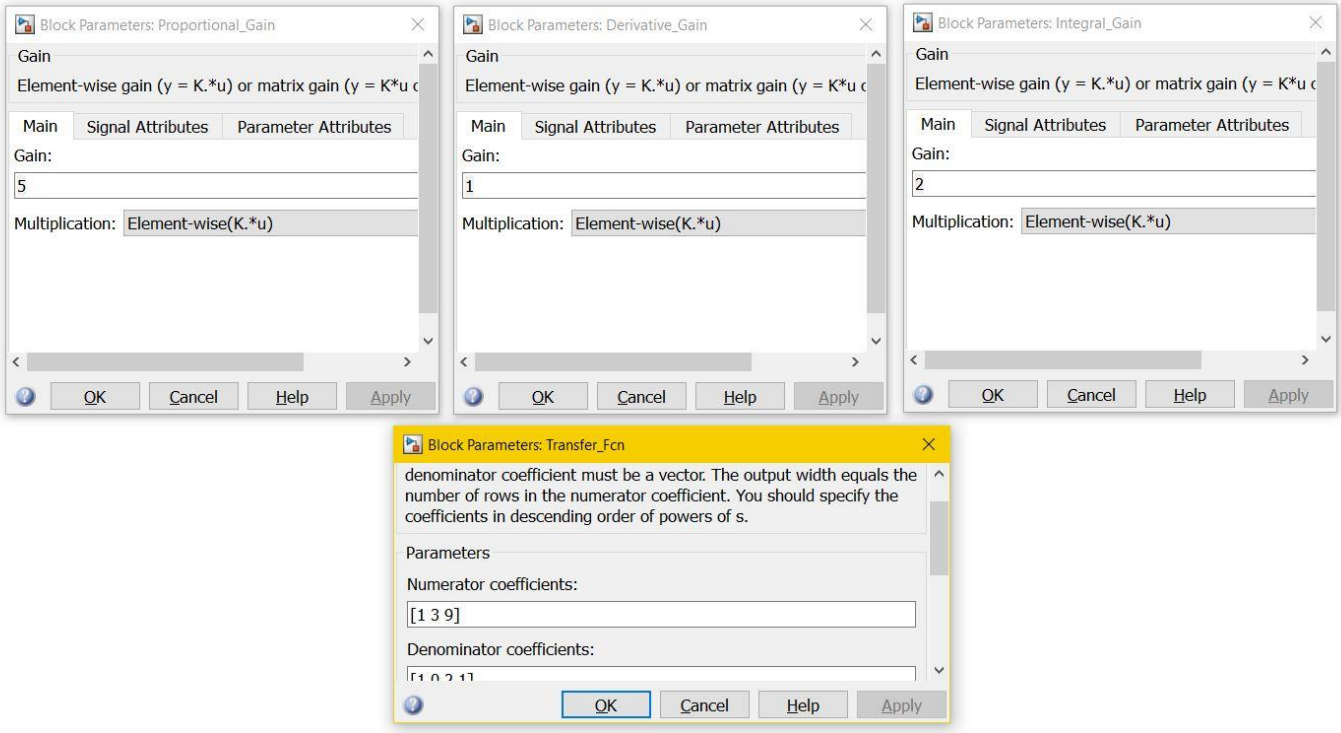
Simulink model:



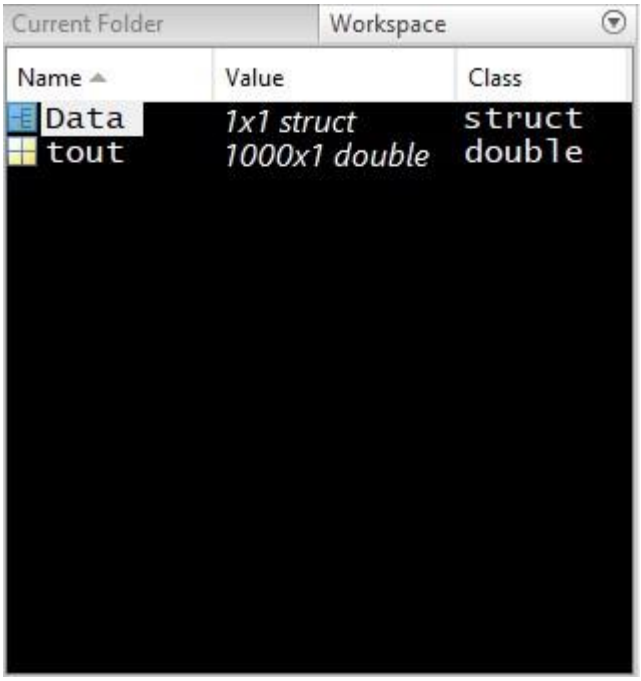
Output & Input Scopes:



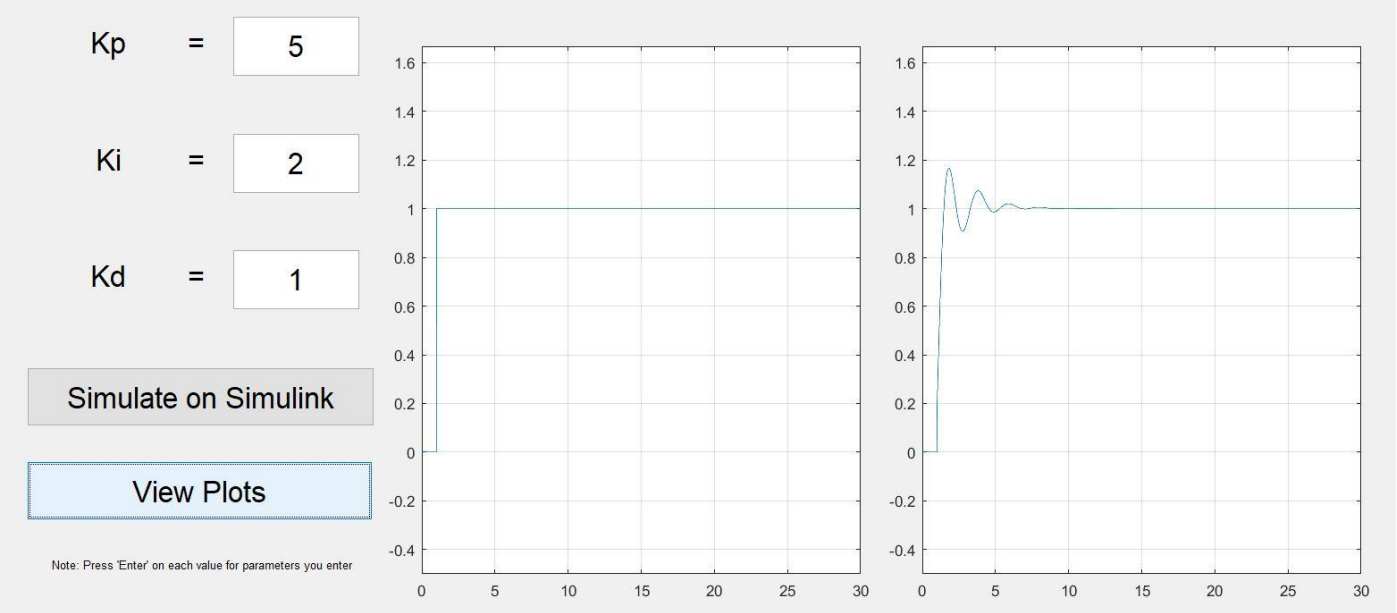
Modified Simulink Parameter:



Workspace:



Bonus:



Edited parts of the associated code:

Function Change No. 1

```
function No_3_PID_Controller_Modified_OpeningFcn(hObject,
eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see
GUIDATA)
% varargin   command line arguments to untitled1 (see
VARARGIN)

%%%                 %%%
%%% Start Changes %%%
%%%                 %%%

model_open(handles)
% Choose default command line output for final
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% Now we can use the figure, as required.
%  Set model parameters to match GUI settings

model_open(handles)

%%%                 %%%
%%%  End Changes  %%%
%%%                 %%%
% Choose default command line output for
No_3_PID_Controller_Modified
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);
```

```
% --- Outputs from this function are returned to the command
line.
function varargout =
No_3_PID_Controller_Modified_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see
VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see
GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;
%end function No_3_PID_Controller_Modified

%%%                %%%
%%% Start Changes %%%
%%%                %%%

% Ensure that the Simulink model is open
function model_open(handles)
% Make sure the diagram is still open
  if
isempty(find_system('Name','No_3_PID_Controller_Simulink')),
      % check whether our Simulink model is opened or not
    open_system('No_3_PID_Controller_Simulink');
  end
  %endfunction model_open

%%%                %%%
%%%  End Changes  %%%
%%%                %%%
```

Function Change No. 2

```
function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see
GUIDATA)

% Ensure model is open
model_open(handles)

% Get the new value
kp_NewStrVal = get(hObject,'String');
kp_NewVal = str2double(kp_NewStrVal);


% Set the Gain parameter of the Kp Gain Block to the new value
set_param('No_3_PID_Controller_Simulink/Proportional_Gain','Ga
in',kp_NewStrVal);
```

Function Change No. 3

```
function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see
GUIDATA)

% Ensure model is open
model_open(handles)

% Get the new value
ki_NewStrVal = get(hObject,'String');
ki_NewVal = str2double(ki_NewStrVal);


% Set the Gain parameter of the Ki Gain Block to the new value
set_param('No_3_PID_Controller_Simulink/Integral_Gain','Gain',
ki_NewStrVal);
```

Function Change No. 4

```
function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see
GUIDATA)

% Ensure model is open
 model_open(handles)

 % Get the new value
 kd_NewStrVal = get(hObject,'String');
 kd_NewVal = str2double(kd_NewStrVal);

 % Set the Gain parameter of the Kd Gain Block to the new
value

set_param('No_3_PID_Controller_Simulink/Derivative_Gain','Gain
',kd_NewStrVal);
```

Function Change No. 5,6

```
function simulatebutton_Callback(hObject, eventdata, handles)
% hObject    handle to simulatebutton (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see
GUIDATA)
myfunc()

function myfunc
    a =
sim('No_3_PID_Controller_Simulink','SimulationMode','normal');
    b = a.get('Data');
    assignin('base','Data',b);
% --- Executes on button press in plotbutton.
function plotbutton_Callback(hObject, eventdata, handles)
sim('No_3_PID_Controller_Simulink');
axes(handles.axes1)
x=Data.time;
y1=Data.signals(1).values;
y2=Data.signals(2).values;
plot(x,y1);
grid on;
axis([min(x) max(x) min(y2)-0.5 max(y2)+0.5]);
axes(handles.axes2)
x=Data.time;
y1=Data.signals(1).values;
y2=Data.signals(2).values;
plot(x,y2);
grid on;
axis([min(x) max(x) min(y2)-0.5 max(y2)+0.5]);
```