

## Part II:

### Bonus:

#### 1) .m file (GUI Script):

```
function varargout = No_3_PID_Controller(varargin)
%
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @No_3_PID_Controller_OpeningFcn, ...
                  'gui_OutputFcn',  @No_3_PID_Controller_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(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

% --- Executes just before No_3_PID_Controller is made visible.
function No_3_PID_Controller_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
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_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

%%%
%%% 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 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', 'Gain', kp_NewStrVal);

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

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);

```

```

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

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);

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

% --- Executes on button press in simulatebutton.
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()

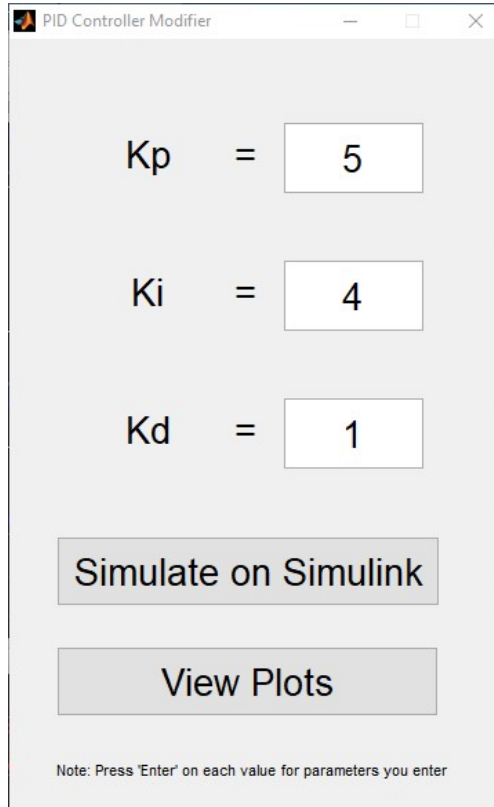
function myfunc
    a = sim('No_3_PID_Controller_Simulink', 'SimulationMode', 'normal'); %PID simulation
    b = a.get('Input_Data');
    c = a.get('Output_Data'); %import data
    assign('base', 'Input_Data', b);
    assign('base', 'Output_Data', c);

% --- Executes on button press in plotbutton.
function plotbutton_Callback(hObject, eventdata, handles)
sim('No_3_PID_Controller_Simulink');
x1=Input_Data(:, 1);
y1=Input_Data(:, 2);
x2=Output_Data(:, 1);
y2=Output_Data(:, 2);

```

```
figure('Name','Input Waveform','Position',[215,130,500,500])
plot(x1,y1);
grid on;
axis([min(x1) max(x1) min(y2)-0.5 max(y2)+0.5]);
figure('Name','Output Waveform','Position',[815,130,500,500])
plot(x2,y2);
grid on;
axis([min(x1) max(x1) min(y2)-0.5 max(y2)+0.5]);
```

## 2) .fig file (GUI Figure):



PID Controller Modifier

Kp = 5

Ki = 4

Kd = 1

Simulate on Simulink

View Plots

Note: Press 'Enter' on each value for parameters you enter

## 3) Waveforms:

