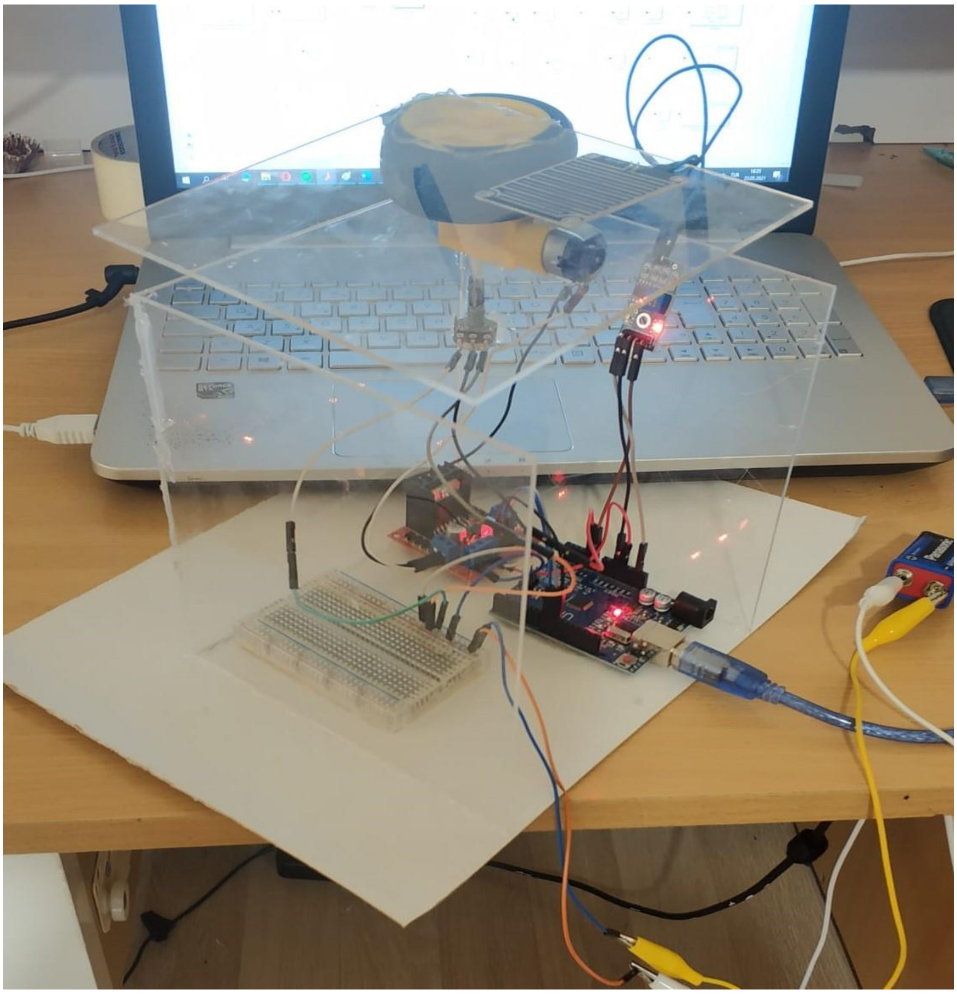
MEE 428 REAL TIME CONTROL
FINAL PROJECT

MY PROPOSED SYSTEM :

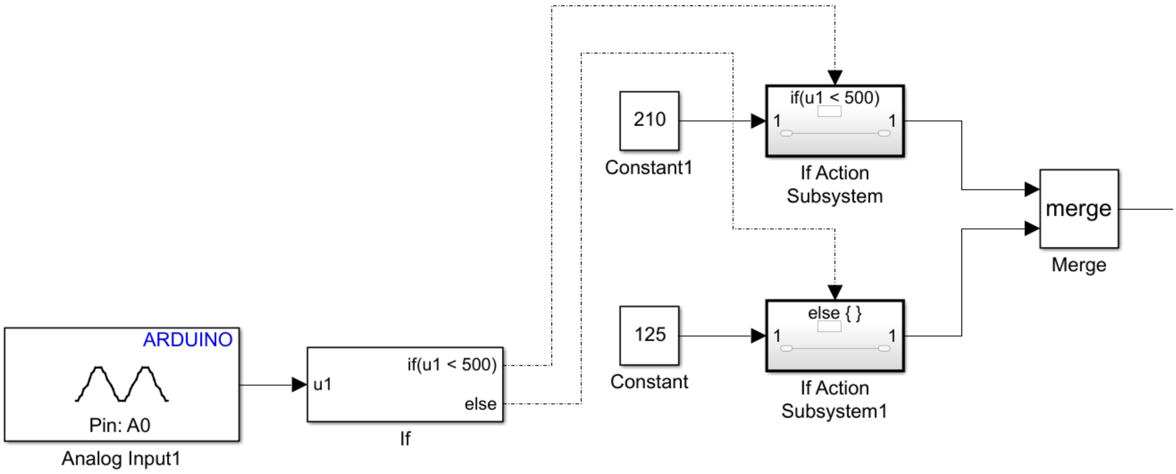
As a project, my design a system that can be switched on and off depending on rain.

Materials : 1 rain sensor(FC-37), 1 potentiometer, 1 DC motor, L295-H motor driver, Arduino uno, 12V 1a Adapter.

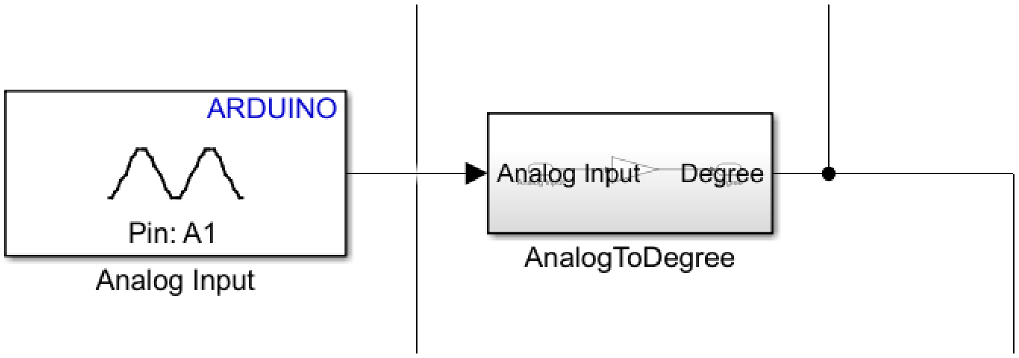
The aim of my project is to save water in greenhouses. With the rain sensor on the greenhouse, the roof opens when it rains.Thus, rainwater feeds the plants. Then the greenhouse is covered when there is no rain. In this way, I reduce the cost by saving in water use. I will be able to learn how many degrees the roof will open and how long it will remain open with the potentiometer.



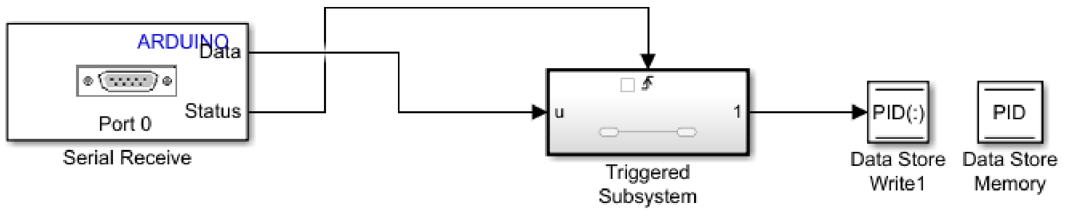
EXPLANATION OF SIMULINK MODEL



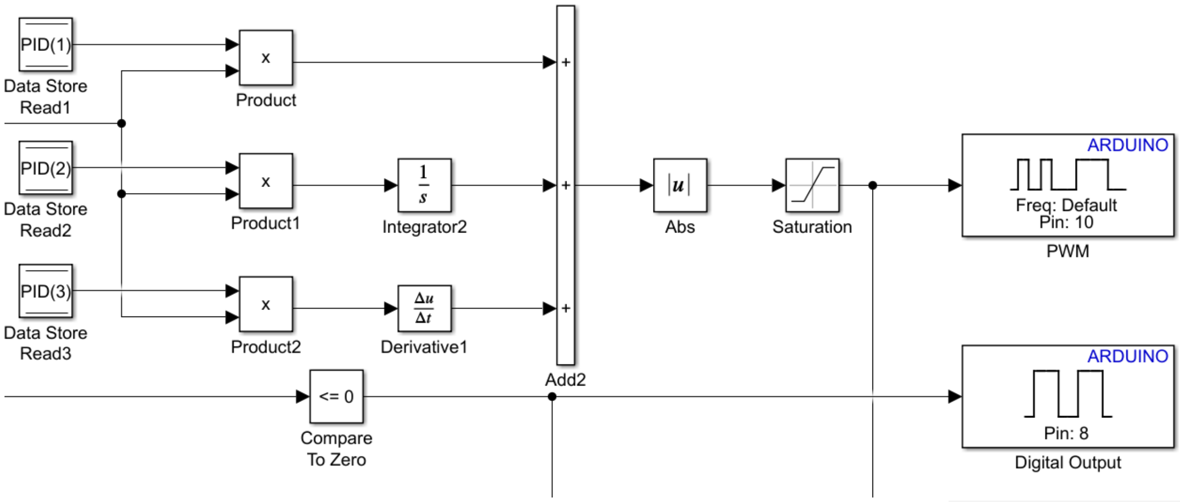
I am getting the analog data I measured from the sensor from the A0 pin. The values I measured from the sensor are between 0 and 1023. I converted these analog data to digital signals with the if else command, 0 if greater than 500, and 1 if less than 500. I combined this data with the merge block. I also added constant blocks for the rotation angle of the motor. With the potentiometer, I can find out the rotation angle of the DC motor and how long the roof is open and closed. When the potentiometer shows 210 °, the roof is open and the roof is closed when it points 125°.



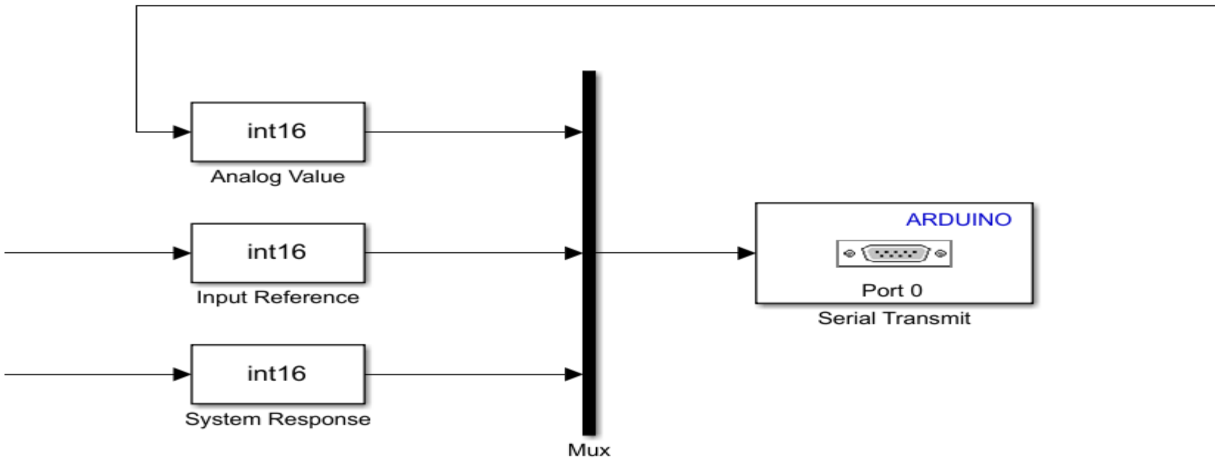
I'm getting the analog data on the potentiometer from pin A1. I converted these analog to degrees. I extracted the data from the sensor from the data from the potentiometer with the sum block.



I made data length 3 in this part because I have 3 values (P, I, d) .When I give an input value, it gives status 1, then it enters the triggered subsystem, and then my values come out. Values from Data Store Write are being read and written in Data Store Memory.

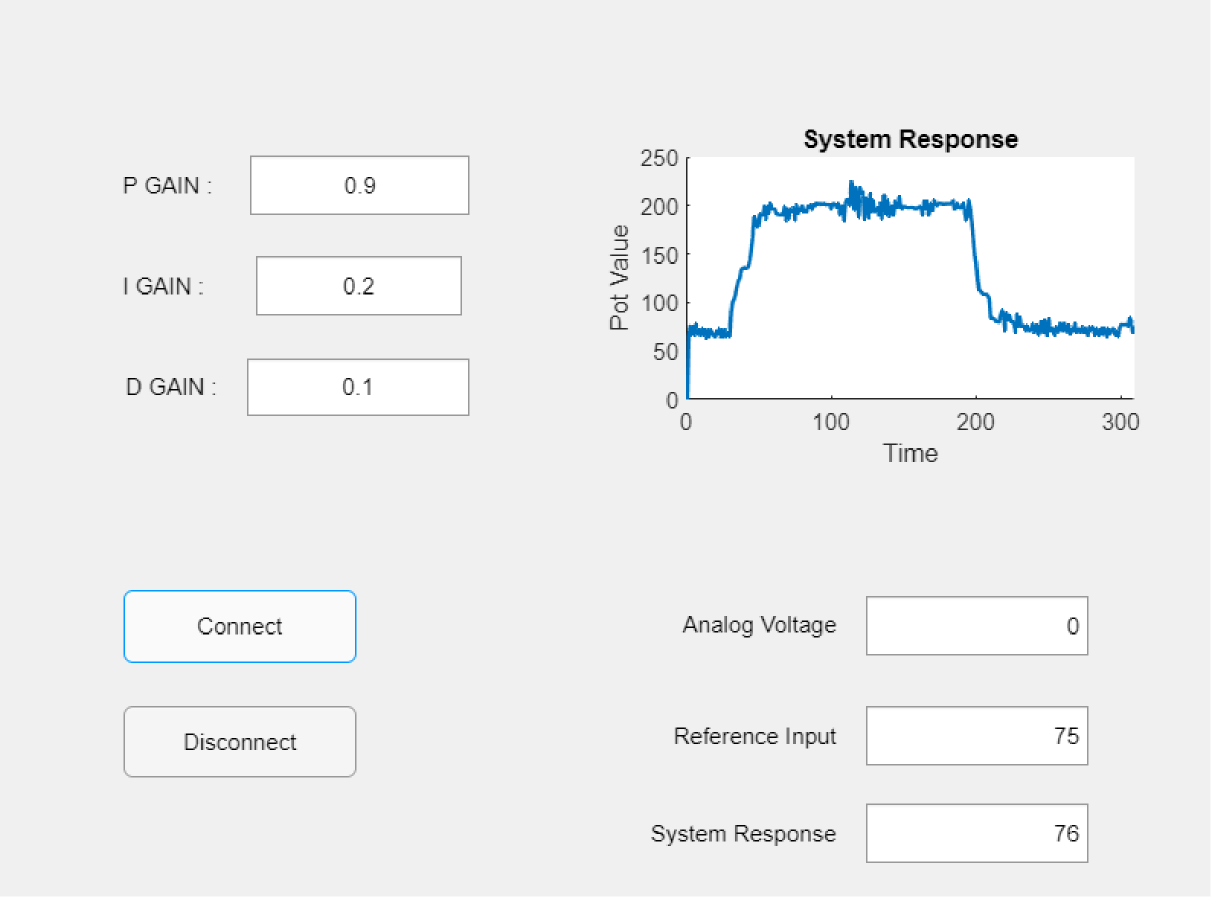


I'm reading the data in the datastore here by Data Store Read. There are 3 Data Store Read because I'm inputting 3 values. Then the values here are multiplied by the gain and the reference. Then i used absolute block. I used saturation block and limited the operation of the DC motor(0-65). I used logic so I can control the direction of the DC motor.



In this part, I print the data I get from my system in the GUI where necessary.

GUI EXPLANATION



P GAIN : 0.9

I GAIN : 0.2

D GAIN : 0.1

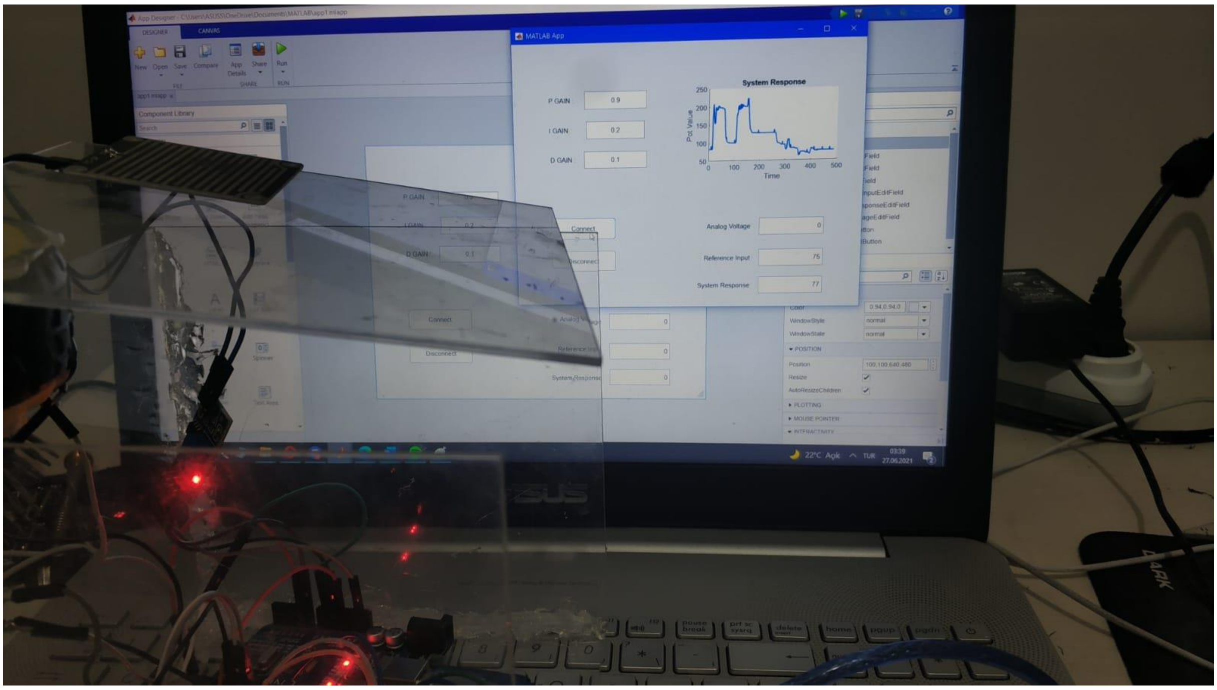
Analog Voltage 0

Reference Input 75

System Response 76

I used App designer instead of GUI because it is simpler to use than GUI. I start the system with the connect button and stop the system with disconnect.I can control my system by entering PIDV and reference input values. By entering the pid values, I make the system work more smoothly. You can see the values necessary for my system to work well. At these values, the error varies between 1-2 in the input and output of my system.

When the roof of the system is closed



P GAIN 0.9

I GAIN 0.2

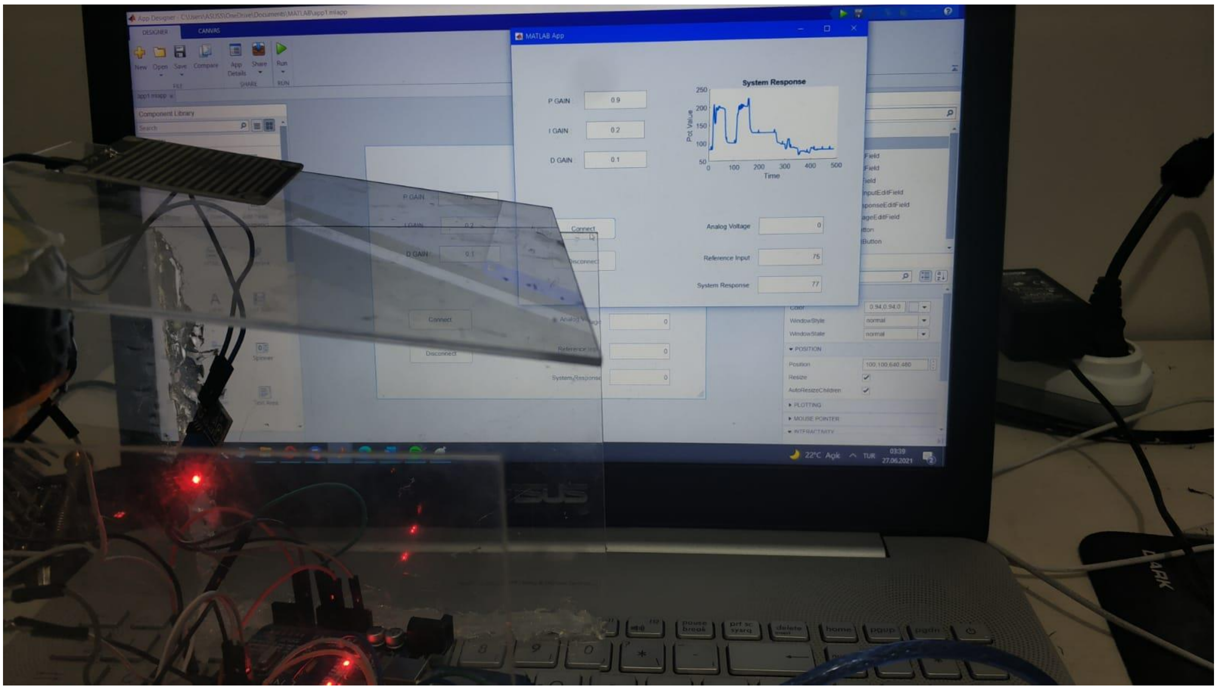
D GAIN 0.1

Analog Voltage 0

Reference Input 75

System Response 77

When the roof of the system is opened



P GAIN 0.9

I GAIN 0.2

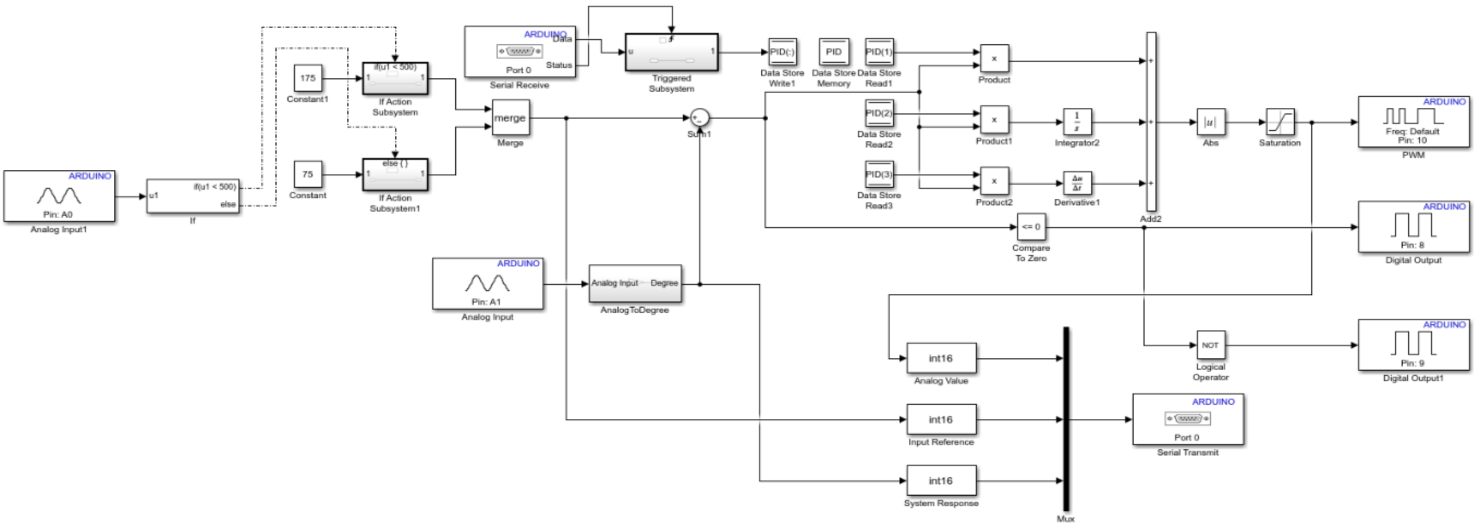
D GAIN 0.1

Analog Voltage 0

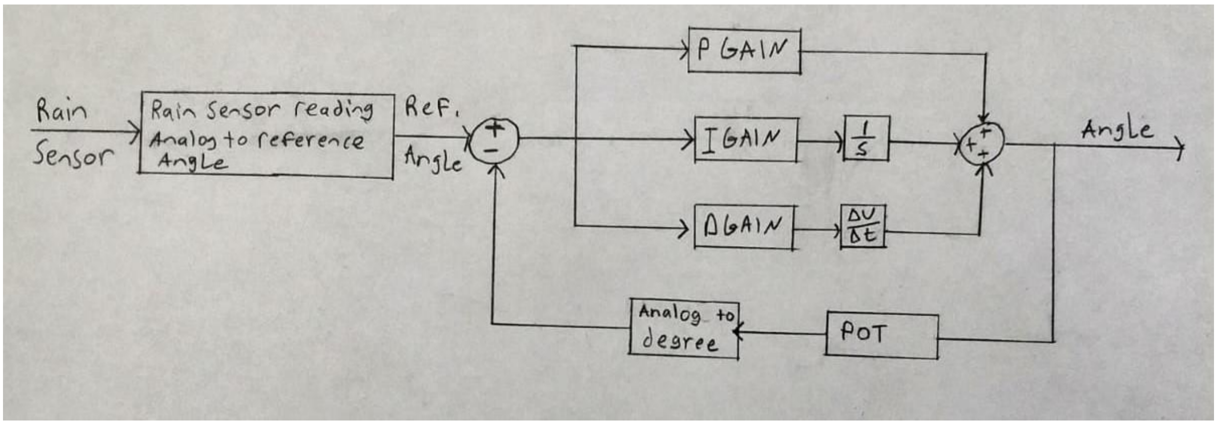
Reference Input 75

System Response 77

BLOCK DIAGRAM

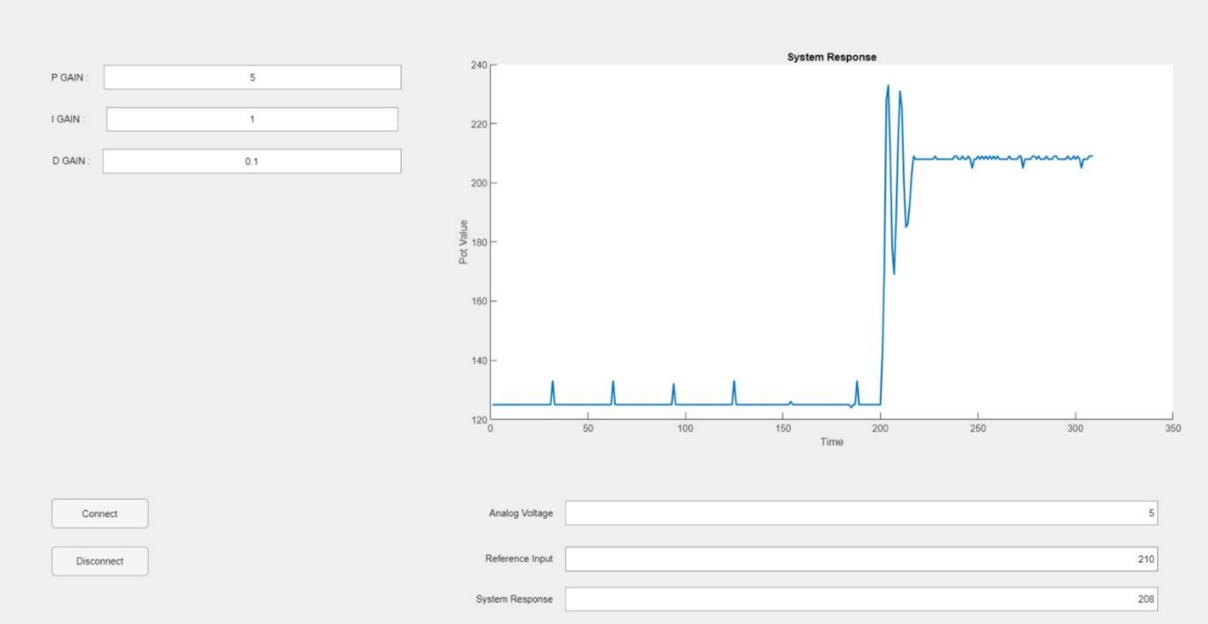


SYSTEM DEMONSTRATION



0-1023 analog value coming from rain sensor, reference input is determined with if-else command. Then compared with the reference angle and degree analog values from the feedback. It sends a command to the dc motor according to the error. Trying to reduce the error with pid values.

UNDER DAMPED SYSTEM



P GAIN : 5

I GAIN : 1

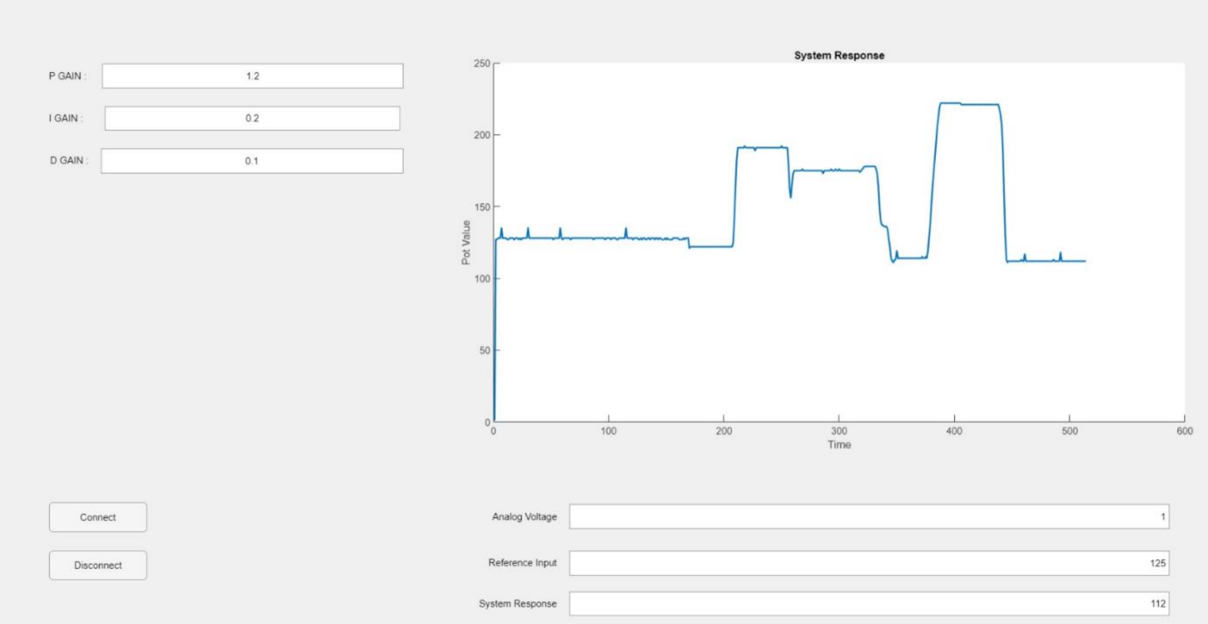
D GAIN : 0.1

Analog Voltage 5

Reference Input 210

System Response 208

OVER DAMPED SYSTEM



P GAIN : 1.2

I GAIN : 0.2

D GAIN : 0.1

Analog Voltage 1

Reference Input 125

System Response 112

CONCLUSION AND RESULT

In this project, I saw how we would encounter the theoretical knowledge I learned in the lessons in real life. I made a computer simulation of the system I had planned with block diagrams. The system extracts the reference value by analyzing the data from the sensor. With this value, the value measured from the potentiometer is Sum. then it enters the PID blocks and gives direction to the motor. We must choose PID values for the system to work well. I observed that when I increased the PID values of the system, the system constantly went back and forth. As a result of my observations, the best reference value for my system is P= 1.2 I=0.2 D=0.1 . Also, by changing the PID values, I obtained the over-damping and under-damping graphs in the system response. You can see these values in the video and in the report.

I could not draw the critical damping graph as an error in my system. I think the reason for this is mechanical problems in my system.Also, one of the other errors in my system is that when I enter the reference input value, the system responds very quickly to this answer and goes back to the old reference value.

In general, my system was working fine. This project was very useful for me to improve my MATLAB skills.

APP DESIGNER (GUI) CODE

```
classdef app1 < matlab.apps.AppBase
% Properties that correspond to app components
properties (Access = public)
UIFigure matlab.ui.Figure
PGAINEditField matlab.ui.control.NumericEditField
PGAINEditFieldLabel matlab.ui.control.Label
DGAINEditField matlab.ui.control.NumericEditField
DGAINLabel matlab.ui.control.Label
IGAINEditField matlab.ui.control.NumericEditField
IGAINLabel matlab.ui.control.Label
ReferenceInputEditField matlab.ui.control.NumericEditField
ReferenceInputEditFieldLabel matlab.ui.control.Label
SystemResponseEditField matlab.ui.control.NumericEditField
SystemResponseEditFieldLabel matlab.ui.control.Label
AnalogVoltageEditField matlab.ui.control.NumericEditField
AnalogVoltageEditFieldLabel matlab.ui.control.Label
ConnectButton matlab.ui.control.Button
DisconnectButton matlab.ui.control.Button
UIAxes matlab.ui.control.UIAxes
end


properties (Access = private)
Response % variable y for the system's response
s % Serial
ctr % Counter to store the elements in arrays for plot
end
methods (Access = private)
function ReadInputDataOverSerial(app, hObject, eventdata, handles)
%Read
Values=fread(app.s,[1,app.s.BytesAvailable],'char')
if(~isempty(Values))
Analog_Voltage = Values(1);
Reference_Input = Values(3);
System_Response = Values(5);
%Display in GUI
app.SystemResponseEditField.Value = System_Response;
app.AnalogVoltageEditField.Value = Analog_Voltage;
app.ReferenceInputEditField.Value = Reference_Input;
%Store
app.Response{app.ctr} = System_Response;
app.ctr = app.ctr + 1;
end
%Plot for system response
x = 1:size(app.Response,2);
y = zeros(1,size(app.Response,2));
for i=1:size(y,2)
y(i) = app.Response{i};
end
```

```matlab
if (mod(app.ctr,5)==0)
plot(app.UIAxes,x,y,'LineWidth',1.5);
end
end
end
% Callbacks that handle component events
methods (Access = private)
% Code that executes after component creation
function startupFcn(app)
app.Response = cell(1,1); %Initialize the variable
app.ctr = 1; %Set the counter to 1
end

% Value changed function: PGAINEditField
function PGAINEditFieldValueChanged(app, event)
value = app.PGAINEditField.Value;
fwrite(app.s,value) % input to be able to enter a value.
End

% Value changed function: IGAINEditField
function IGAINEditFieldValueChanged(app, event)
value = app.IGAINEditField.Value;
fwrite(app.s,value) % input to be able to enter a value.
End



% Value changed function: DGAINEditField
function DGAINEditFieldValueChanged(app, event)
value = app.DGAINEditField.Value;
fwrite(app.s,value) % input to be able to enter a value.
End
% Button pushed function: ConnectButton
function ConnectButtonPushed(app, event)
app.s=serial('COM3','BaudRate',9600);
app.s.BytesAvailableFcn={@app.ReadInputDataOverSerial};
app.s.BytesAvailableFcnCount = 8;
app.s.BytesAvailableFcnMode='byte';
fopen(app.s); % Open The Gui
end


% Button pushed function: DisconnectButton
function DisconnectButtonPushed(app, event)
fclose(app.s); % stop the Gui
end


% Value changed function: ReferenceInputEditField
function ReferenceInputEditFieldValueChanged(app, event)
value = app.ReferenceInputEditField.Value;
fwrite(app.s,value) % input to be able to enter a value.
end
end
```

```
% Component initialization
methods (Access = private)


% Create UIFigure and components
function createComponents(app)


% Create UIFigure and hide until all components are created
app.UIFigure = uifigure('Visible', 'off');
app.UIFigure.Position = [100 100 640 480];
app.UIFigure.Name = 'MATLAB App';


% Create UIAxes
app.UIAxes = uiaxes(app.UIFigure);
title(app.UIAxes, 'System Response')
xlabel(app.UIAxes, 'Time')
ylabel(app.UIAxes, 'Pot Value')
zlabel(app.UIAxes, 'Z')
app.UIAxes.Position = [319 233 284 183];




% Create DisconnectButton
app.DisconnectButton = uibutton(app.UIFigure, 'push');
app.DisconnectButton.ButtonPushedFcn = createCallbackFcn(app,
@DisconnectButtonPushed, true);
app.DisconnectButton.Position = [67 71 123 37];
app.DisconnectButton.Text = 'Disconnect';


% Create ConnectButton
app.ConnectButton = uibutton(app.UIFigure, 'push');
app.ConnectButton.ButtonPushedFcn = createCallbackFcn(app,
@ConnectButtonPushed, true);
app.ConnectButton.Position = [67 131 123 38];
app.ConnectButton.Text = 'Connect';


% Create AnalogVoltageEditFieldLabel
app.AnalogVoltageEditFieldLabel = uilabel(app.UIFigure);
app.AnalogVoltageEditFieldLabel.HorizontalAlignment = 'right';
app.AnalogVoltageEditFieldLabel.Position = [356 140 86 22];
app.AnalogVoltageEditFieldLabel.Text = 'Analog Voltage';


% Create AnalogVoltageEditField
app.AnalogVoltageEditField = uieditfield(app.UIFigure, 'numeric');
app.AnalogVoltageEditField.Position = [457 135 117 31];
```

```
% Create SystemResponseEditFieldLabel
app.SystemResponseEditFieldLabel = uilabel(app.UIFigure);
app.SystemResponseEditFieldLabel.HorizontalAlignment = 'right';
app.SystemResponseEditFieldLabel.Position = [339 31 103 22];
app.SystemResponseEditFieldLabel.Text = 'System Response';


% Create SystemResponseEditField
app.SystemResponseEditField = uieditfield(app.UIFigure, 'numeric');
app.SystemResponseEditField.Position = [457 26 117 31];


% Create ReferenceInputEditFieldLabel
app.ReferenceInputEditFieldLabel = uilabel(app.UIFigure);
app.ReferenceInputEditFieldLabel.HorizontalAlignment = 'right';
app.ReferenceInputEditFieldLabel.Position = [351 82 91 22];
app.ReferenceInputEditFieldLabel.Text = 'Reference Input';


% Create ReferenceInputEditField
app.ReferenceInputEditField = uieditfield(app.UIFigure, 'numeric');
app.ReferenceInputEditField.ValueChangedFcn = createCallbackFcn(app,
@ReferenceInputEditFieldValueChanged, true);
app.ReferenceInputEditField.Position = [457 77 117 31];


% Create IGAINLabel
app.IGAINLabel = uilabel(app.UIFigure);
app.IGAINLabel.Position = [67 318 61 22];
            app.IGAINLabel.Text = 'I GAIN :';


            % Create IGAINEditField
            app.IGAINEditField = uieditfield(app.UIFigure, 'numeric');
            app.IGAINEditField.ValueChangedFcn = createCallbackFcn(app,
@IGAINEditFieldValueChanged, true);
            app.IGAINEditField.HorizontalAlignment = 'center';
            app.IGAINEditField.Position = [137 313 108 31];
            app.IGAINEditField.Value = 0.2;


            % Create DGAINLabel
            app.DGAINLabel = uilabel(app.UIFigure);
            app.DGAINLabel.HorizontalAlignment = 'right';
            app.DGAINLabel.Position = [63 265 54 22];
            app.DGAINLabel.Text = 'D GAIN :';


            % Create DGAINEditField
            app.DGAINEditField = uieditfield(app.UIFigure, 'numeric');
            app.DGAINEditField.ValueChangedFcn = createCallbackFcn(app,
@DGAINEditFieldValueChanged, true);
            app.DGAINEditField.HorizontalAlignment = 'center';
            app.DGAINEditField.Position = [132 260 117 31];
            app.DGAINEditField.Value = 0.1;

            % Create PGAINEditFieldLabel
            app.PGAINEditFieldLabel = uilabel(app.UIFigure);
            app.PGAINEditFieldLabel.Position = [67 371 52 22];
            app.PGAINEditFieldLabel.Text = 'P GAIN :';

            % Create PGAINEditField
            app.PGAINEditField = uieditfield(app.UIFigure, 'numeric');
            app.PGAINEditField.ValueChangedFcn = createCallbackFcn(app, 
@PGAINEditFieldValueChanged, true);
            app.PGAINEditField.HorizontalAlignment = 'center';
            app.PGAINEditField.Position = [134 366 115 31];
```