PROJECT REPORT

The project includes webserver which has the two applications, out of which one application is capable of simulating the monitor and control action of PID. It will help a person understand the working of PID. The other application is capable of monitoring and controlling real-time temperature value of any heating device using PID.

The project statement says PID control and monitoring using Webserver. Therefore, in this project we have a webserver, which has applications, which are easily downloadable by anyone around the globe and can used for control and monitoring purposes. The webserver has all the necessary information regarding how to download and use applications along with their manuals. Through this, webserver serves as a chapter for learning PID by anyone anywhere from the applications provided.

While working on the project, we came across difficulties in transporting the data from hardware to webserver and vice versa due to the closed loop process of PID. If webserver is used only to switch On/Off the lights, fans etc. can be done easily as we only need to send a HIGH/LOW signal. However, in case of PID we require a continuous loop to be formed and a strong internet connection to transfer values as fast as possible to reduce the delay in process.

For further improvements in the project, we are focusing on PID applications, which can run on the webserver as a person clicks; this will simulate the closed loop action of PID and removes the need of downloading the application.

Further, in the document, we have explained how the webserver interface created by us works and steps on how to operate the applications.

Made by: Mentor:

Vinod Gujral, B-tech ECE, 3rd year. Jetendra Joshi, Asst. Professor.

Sujoy Roy, B-tech CSE, 4th year.

NIIT University

WEBSERVER INTERFACE

Webserver is developed using HTML and CSS.

Just go to website (DynamicAnalytics.github.io) and follow the steps mentioned below:

Fig 1: Website home page

Fig 2: Click on desired application according to requirement

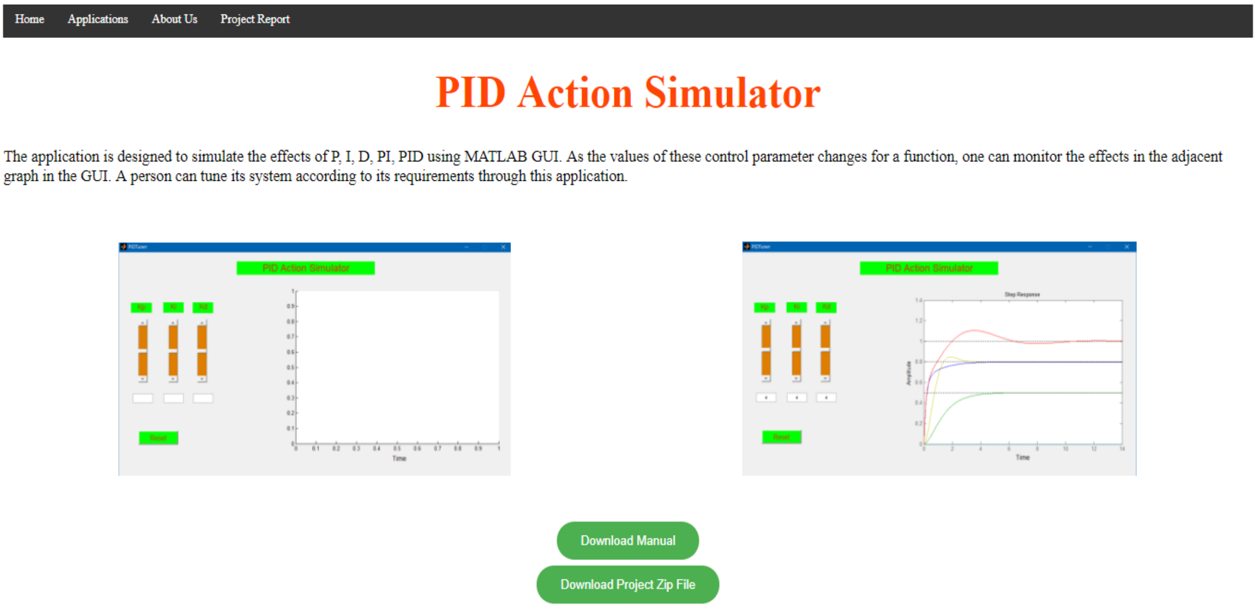

Fig 3: Application 1 page

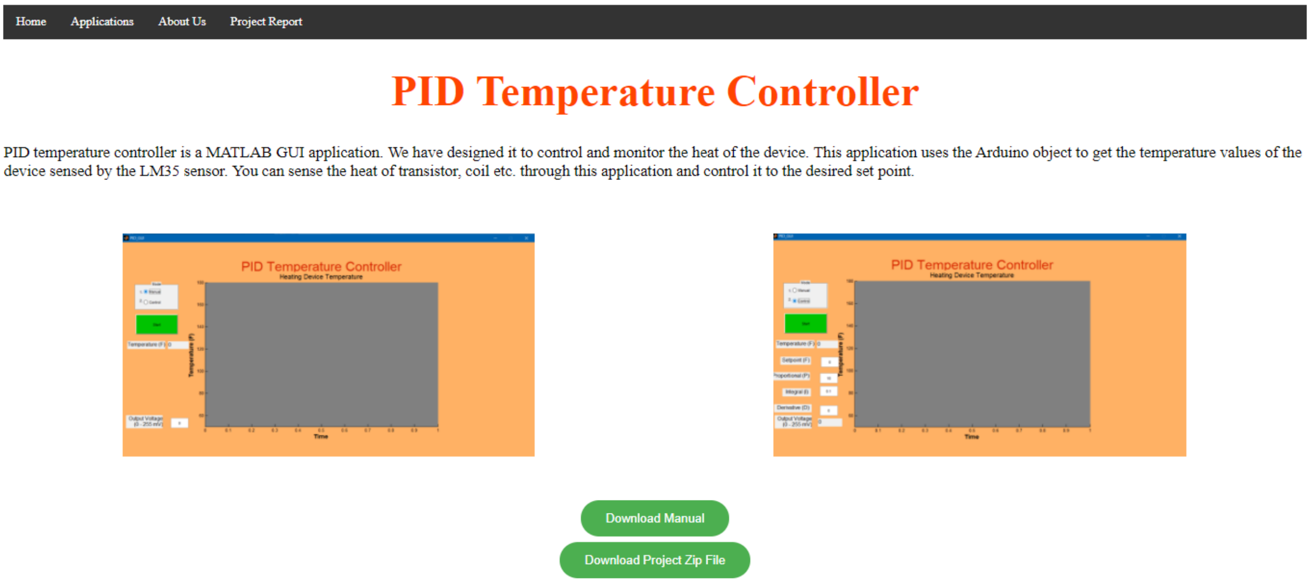

Fig 4: Application 2 page

Anyone can easily download the application source files and their manuals from their respective pages. To download the full project report click on Project Report button to download it.

APPLICATION 1: PID ACTION SIMULATOR

The application is designed to simulate the effects of P, I, D, PI, PID using MATLAB GUI. As the values of these control parameter changes for a function, one can monitor the effects in the adjacent graph in the GUI. A person can tune its system according to its requirements through this application.

To make this application and to execute it we only require MATLAB.

Steps to run the application:

1. Download the application folder from the website.
2. Add the downloaded folder to your MATLAB directory.
3. Run PIDTuner.m file. This will run the MATLAB GUI application.
4. Change the Kp, Ki, Kd and see the effect of these controls in the adjacent graph.
5. Before closing the application, drop down the values of Kp, Ki, Kd to zero and click reset.

Screenshots of the application:

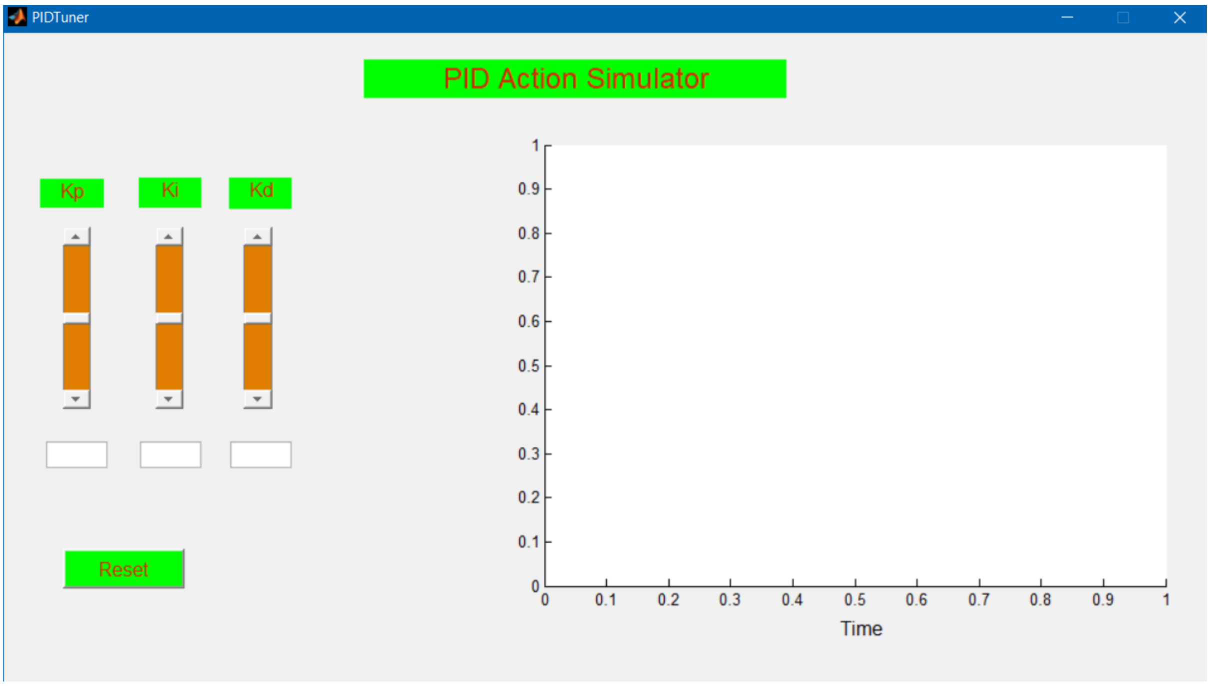

Fig 1: Application interface

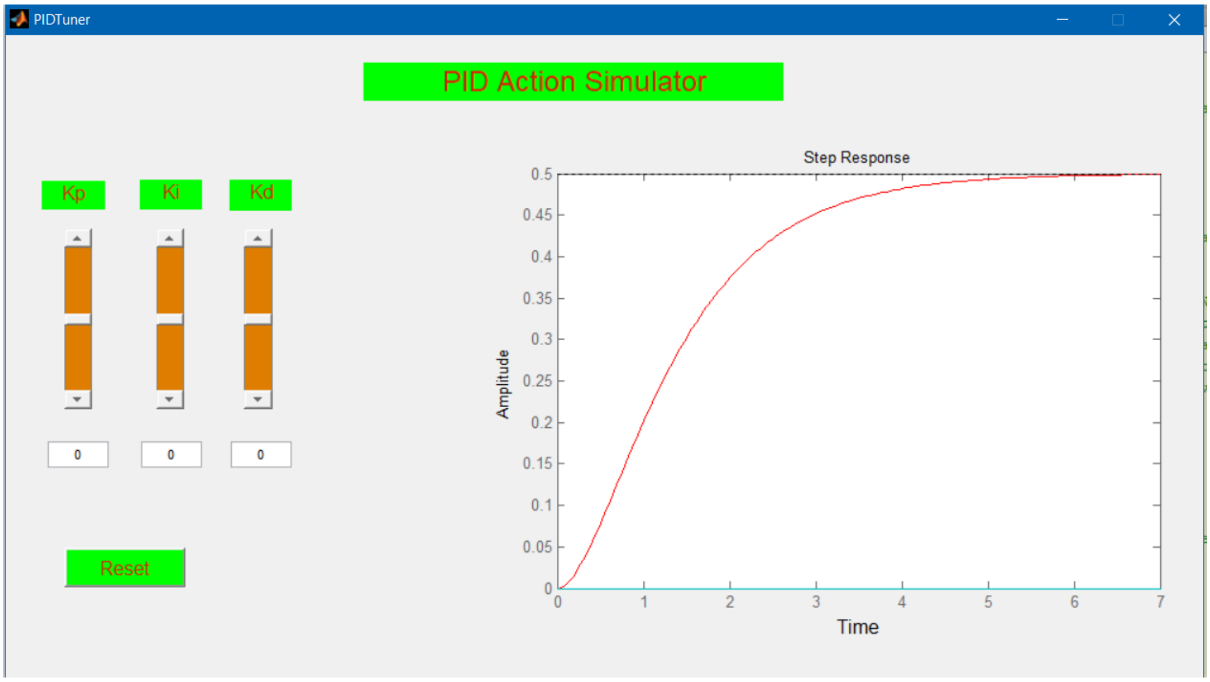

Fig 2: Step response without control

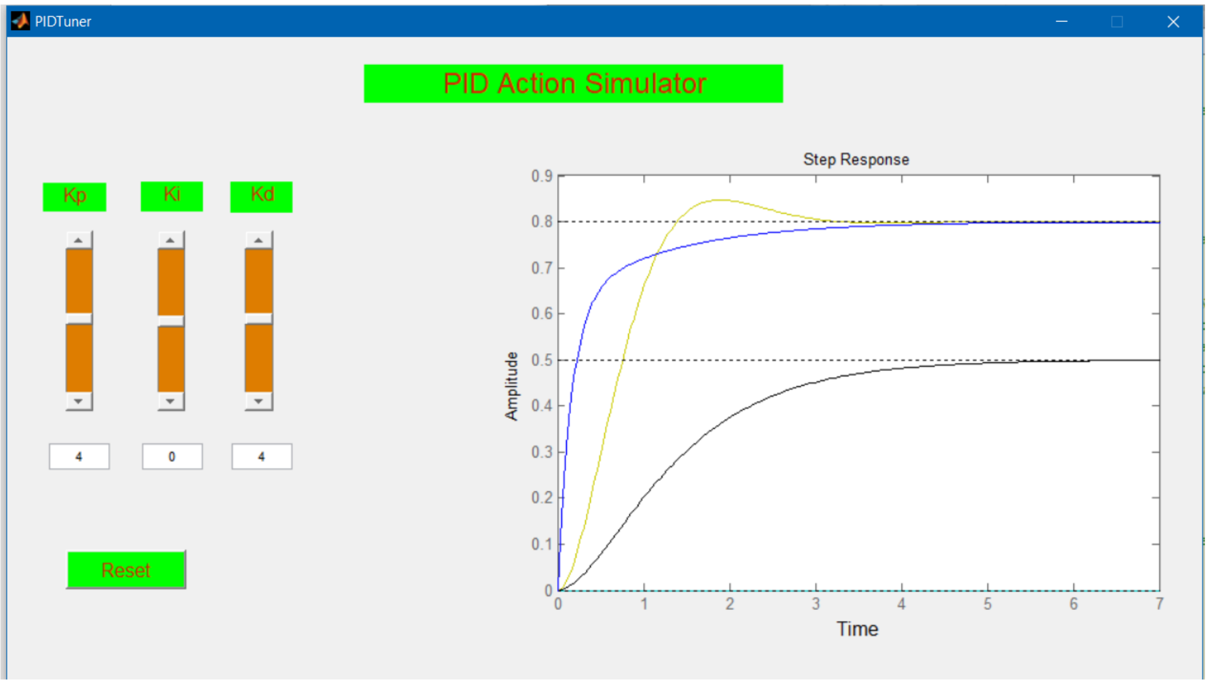

Fig 3: PD control

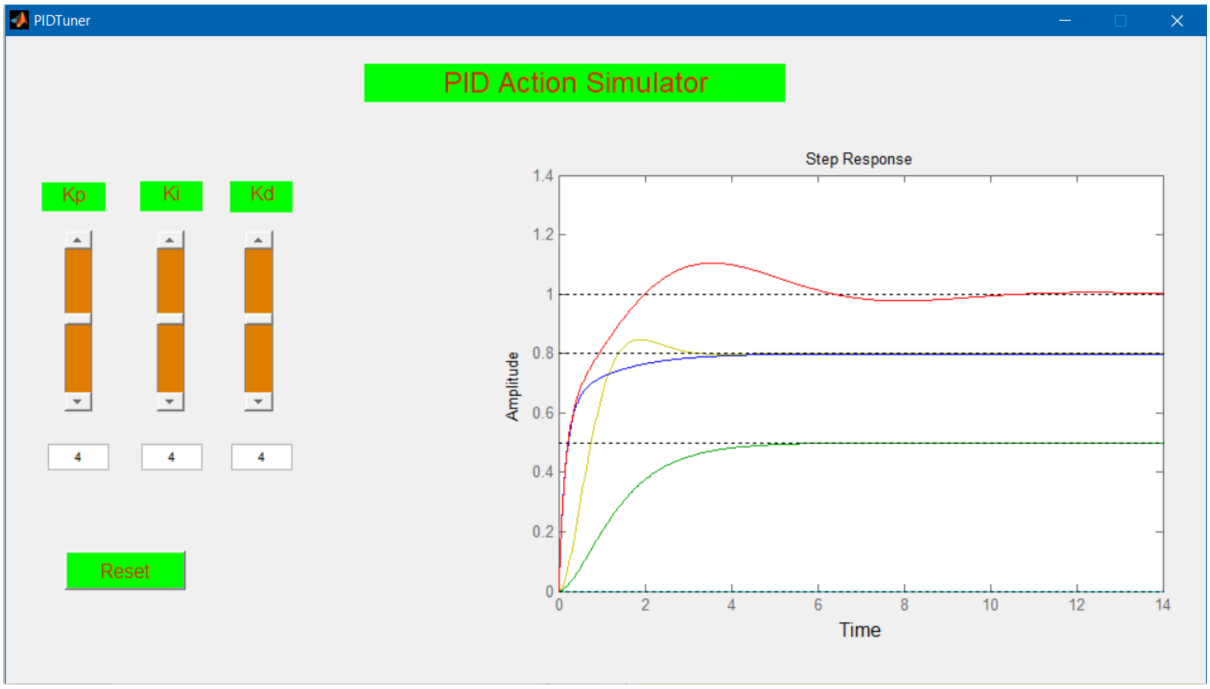

Fig 4: PID control

APPLICATION 2: PID TEMPERATURE CONTROLLER

PID temperature controller is a MATLAB GUI application. We have designed it to control and monitor the heat of the device. This application uses the Arduino object to get the temperature values of the device sensed by the LM35 sensor. You can sense the heat of transistor, coil etc. through this application and control it to the desired set point.

For this application, you will require these hardware components listed below-

Arduino Uno

LM35 temperature sensor

Power supply

Transistor, heater (to get variation in temperature values for monitoring and controlling).

Connecting wires

For the software, you will require Arduino IDE and MATLAB version should be above R2013b.

Connections:

Connect the Vcc and GND of the LM35 to the Arduino Uno 3.3V and GND.

Connect the A0 pin of the Arduino Uno to the LM35 data pin.

Connect the transistor or heater with a power supply and connect their data pin through resistor to any PWM pin on Arduino Uno board (prefer pin 3).

After this, connect your Arduino Uno board to your computer.

Download the application folder from the website.

Add it to your MATLAB directory.

Open the PID_GUI.m file and run it. It will show up the application interface.

The application has two modes Manual for open loop temperature monitoring and Control to control the temperature towards set point.

Choose the mode according to your requirement, click Start and observe the effects on graph and values in text box.

Screenshots of the application:

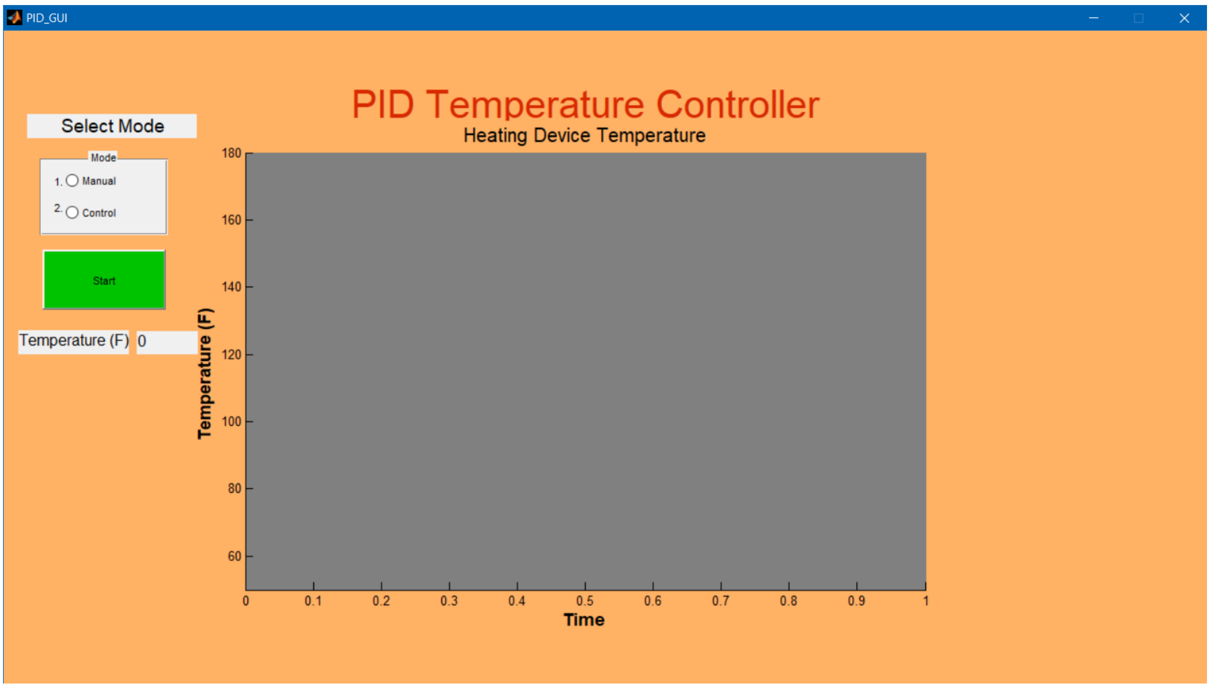

Fig 1: Application interface

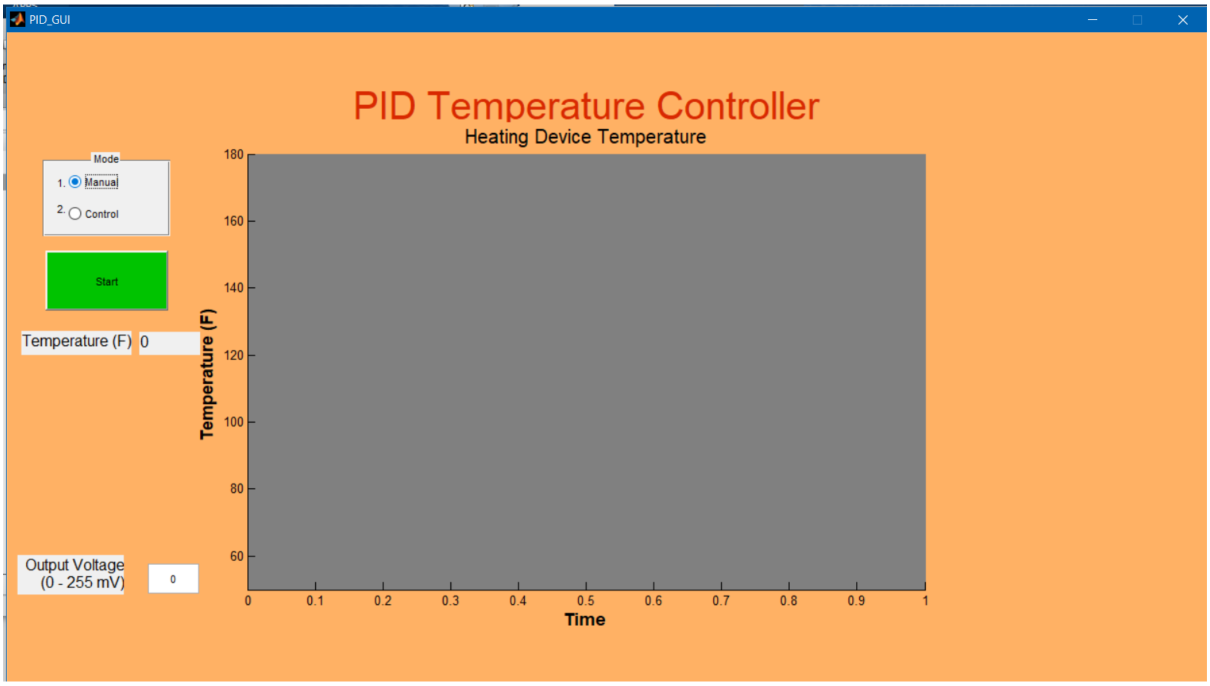

Fig 2: Manual mode

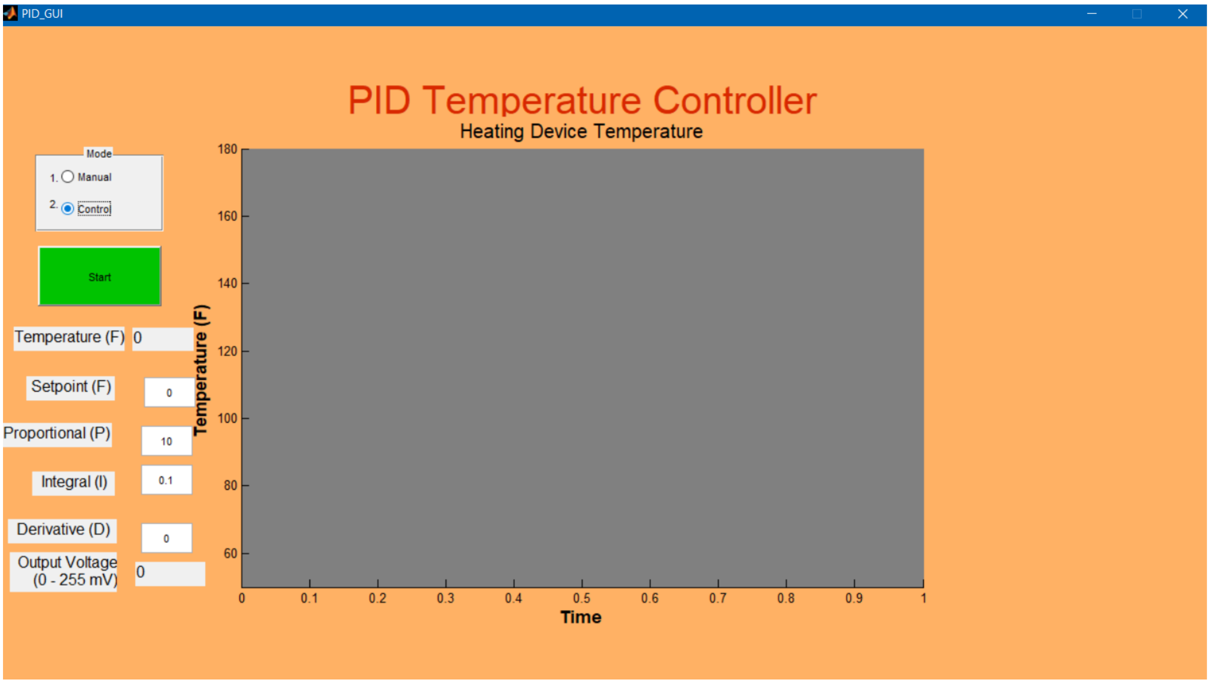

Fig 3: Control mode

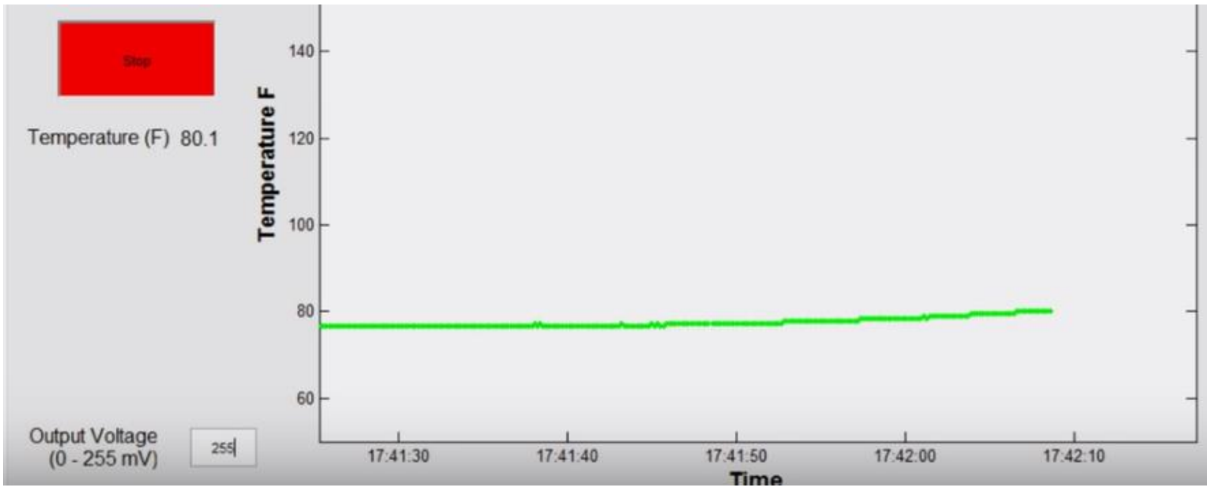

Fig 4: Manual mode graph

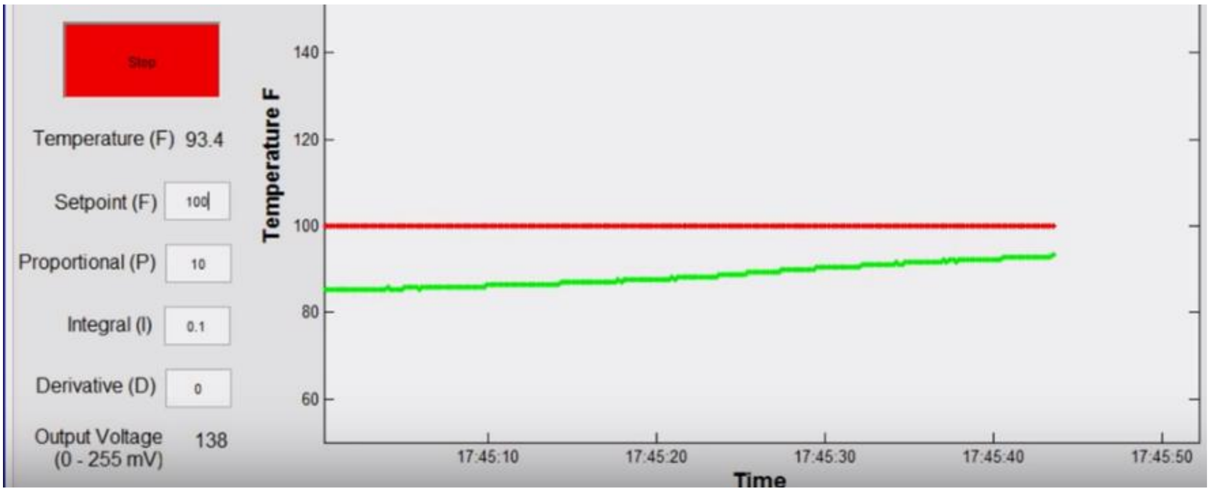

Fig 5: Control mode graph

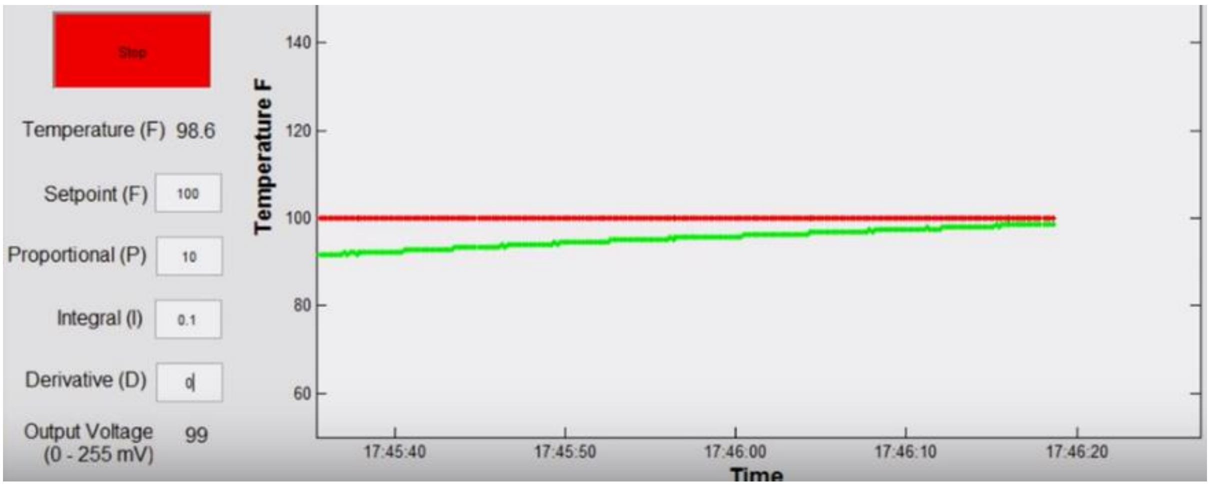

Fig 6: Control mode graph (close to Set point)