LAB 1: Modeling and Control of a Hot Air Plant –Simulation/Experimentation Lab

ENGG4420: Real-Time Systems Design

Instructor:
Dr.Radu Muresan

Group 17: Wed-8:30 Section

Bilal Ayyache: 0988616
Robert Mackenzie Beggs: 0819747

Lab Start Date: October 7, 2020
Lab End Date: November 7, 2020

Contents

List of Figures

List of Tables

1 Introduction

1.1 Problem Description

Real time systems require a continuous quantitative expression of time, throughout which the behaviour of the system can be continuously reported. In real time systems, inputs from and to the plant are freely provided and are not limited by computational resources. In large real time systems, such as a hot air blower, there are typically multiple inputs and outputs. This presents several design challenges, that are overcome via division of the process into a planning and development phase. Given a hot air blower, the aforementioned design process was followed to implement a simulation model of a hot air plant with a PID controller via LabView Control Design ToolKit Box VIs (LabView). Additionally, using the Ziegler and Nichols approach, the PID controller was tuned.

1.2 System Requirements

The difference between an application of a signal and the system's reaction to the signal is referred to as the transportation lag. The implementation aims to correct the presence of transport delay and transfer lag displayed in the PT326 apparatus. A complete and accurate model of the PT326 Process Trainer was provided using LabView, in which the model must include a front panel as specified and be capable of being used to accurately estimate system variables at a given point in time. Additionally, both closed and open loop PID tuning were performed to calculate given parameters. In implementing an accurate model, a given temperature is maintained at the output of the air by varying the amount of power given to the heater grid.

2 Background

2.1 Benefits of LABView

Real time systems are considered non-terminating because interaction with the environment is continuous. In order to model a real time system effectively, a complete plant and an environment model needs to be established; this increases the complexity of designing and testing real time systems. LabView is a graphical programming language that allows both plant systems and environments models to be quickly and efficiently designed and altered.

2.2 PID Control and Tuning Overview

Given a control loop it is possible to adjust various parameters to satisfy given requirements. The act of this adjustment is referred to as PID tuning. A major concept within PID tuning is stability. If a control parameter exits a given range, the process can become unstable; the output can diverge or it can continuously oscillate without reaching steady state. The transfer function is tuned in such a way that it satisfies the Nyquist stability criterion using the Ziegler-Nichols tuning method. The Ziegler-Nichols method dictates that the derivative and integral gains of the system are zeroed and the proportional gain is incrementally increased until the output of the loop is within steady-state as shown in the results section.

3 Implementation

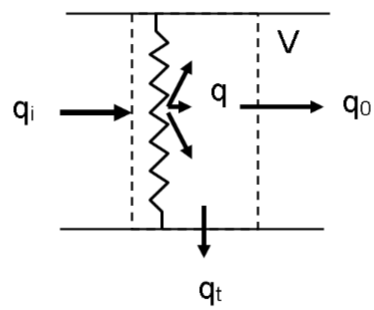

Figure 1: System

The implementation of this system was completed using LabView. The first steps of implementation was reading about the PT326 apparatus and modeling the system using controls modelling methods. The system in Figure 1 was modeled to create the following transfer function :

$$\frac{V_0(s)}{V_i(s)} = \frac{Ke^{-\tau_d s}}{\tau s + 1} \tag{1}$$

After system analysis, a block diagram was used to represent the system to be modeled on LabView.

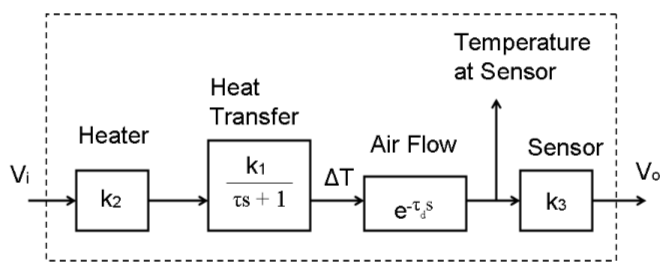

Figure 2: Block Diagram Of System

Using the block diagram in Figure 2, a system can be designed and simulated using LabView. The K value and tau can be calculated using a polynomial fit block in LabView and the data in the tables below:

3.1 Designing Plant

A control panel was designed in LabView to simulate the system created, and display results to the user. Figure 3 represents the control panel design.

Blower Opening	Gain, K	Tau (s)	Pure Time Delay
15	1.37	2	0.195
30	1.33	2	0.19
45	1.26	2	0.185
60	1.22	2	0.18
75	1.18	2	0.17
90	1.13	2	0.16

Temp	Output Voltage
20	0
30	1.13
35	1.669
40	3.611
45	5.326
50	6.154
55	7.611
60	7.783

Table 1: Dataset Table

3.2 Control Panel of System

A control panel was designed in LabView to simulate the system created, and display results to the user. Figure 3 represents the control panel design. Using the control panel, user is able to set the plant control either automatic or manual. If user selects manual control, a voltage input is required to simulate the system. If user selects auto control, the temperature required and the PID gains have to be inputted into the interface. The order was set to 2 and the sampling time was set to 200ms. A stop button can be used to stop the simulation.

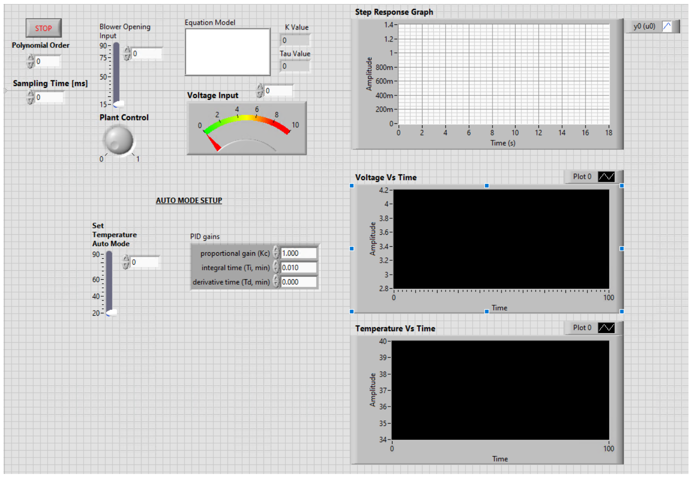

Figure 3: Main Control Panel

3.3 Block Diagram Of System

The block diagram in Figure 4, 5, 6 represents the control panel designed in LabView (Figure 3). In this section, three different block diagrams are presented. The block diagram in Figure 4 computes the transfer function of the system, the block diagram in Figure 5 represents the plant under manual control, and the block diagram in Figure 6 represents the plant under automatic control. In this section, these block diagrams are explained.

3.3.1 Block Diagram Of Transfer Function modelling

In the block diagram presented in Figure 4, a transfer function is created using the data presented in Table 1. To get the K value of the transfer function, a polynomial fit between the blower opening(x) and the gain(y) experimental data. The coefficients where multiplied by the blower opening inputted to the power of the nth term (n in data set). The same procedure was used to compute the tau delay of the transfer function. After K and Tau for the specific blower opening is calculated, a transfer function of the plant was created. The step response of the transfer function is displayed on the control panel. This allows the user to understand the behavior of the system as the data can be exported and analyzed.

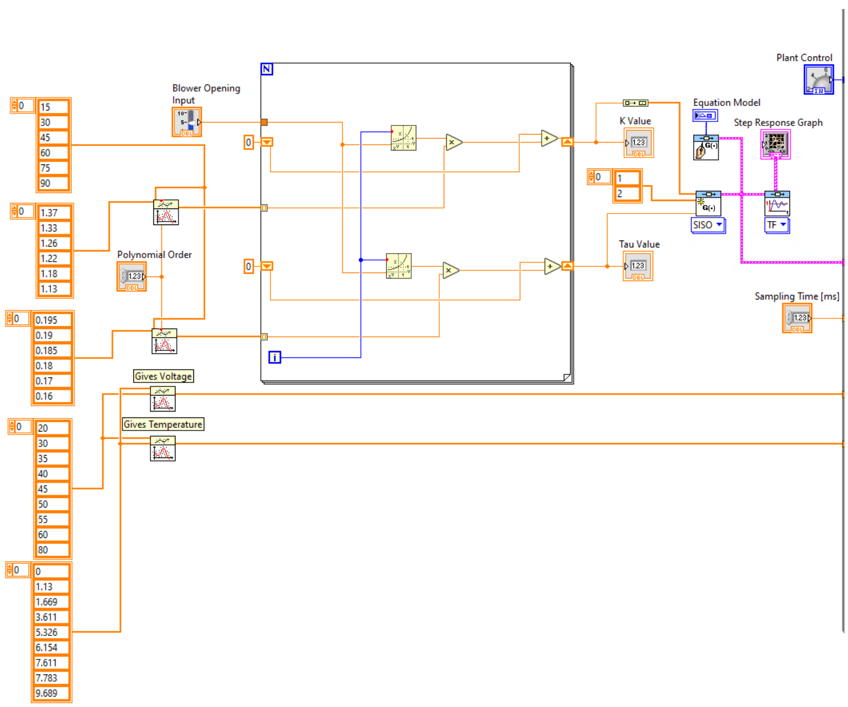

Figure 4: Modeling Transfer Function Block Diagram

3.3.2 Block Diagram of Manual control

The block diagram presented in Figure 5 allows the user to manually control the voltage input into the plant using a variable gauge. The transfer function is first converted to a discrete transfer function in which a voltage value is inputted. A Voltage vs Time graph is plotted using the output of the transfer function.

To get the Temperature vs Time graph, a polynomial fit was used, following same procedure described in Section 3.3.1. This allows user to manually control the temperature output by applying a voltage to the plant.

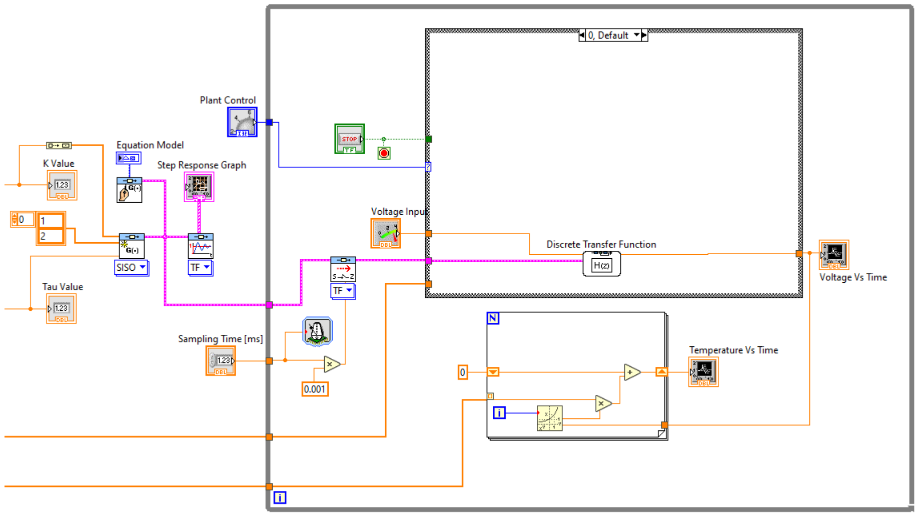

Figure 5: Manual Control Block Diagram

3.3.3 Block Diagram of Auto Control

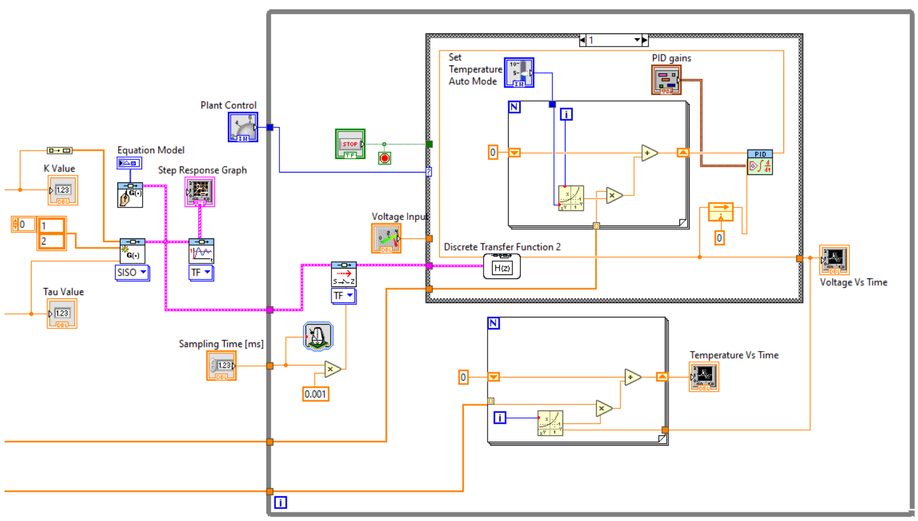

Figure 6: Auto-Control Block Diagram

Figure 6 represents the implementation of a PID controller. The PID uses a feedback system to determine how much voltage should be inputted to achieve the desired temperature provided by the user. To ensure that PID achieves the threshold set, the right PID gains should be calculated. The tuning of the PID is discussed in Section 4. Using a polynomial fit function and Table 1, a corresponding voltage is inputted into the transfer function and a feedback system is inputted back into the PID. This process will ensure the right temperature is achieved by the system.

4 PID Tuning

The PID tuning method of Ziegler and Nichols was used. This method developed the Quarter-Decay Ratio tuning techniques based on theory and empirical observation. The closed loop method is a more accurate method, but harder to implement compared to the open loop method due to the fact that a sustained oscillation must be achieved using a p controller. The sections below highlights the two different methods, and the procedure behind implementing the methods.

4.1 Open Loop Method

This method requires more analysis but does not need to reach sustained oscillation compared to the closed loop method. This analysis must be conducted in manual mode in which the step response graph is used to find the PID gains. To review the steps taken to complete this method review the Appendix.

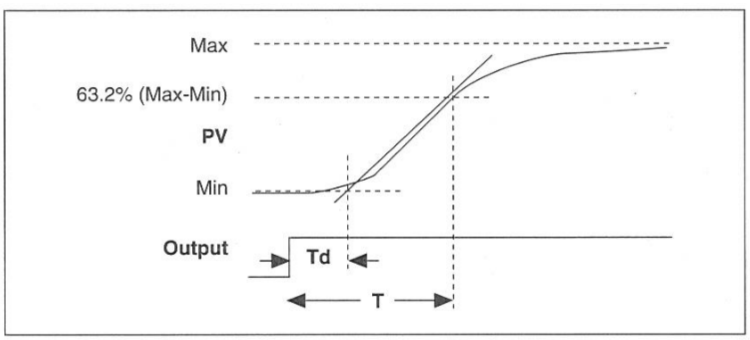

Figure 7: Step Response Analysis

To compute the PID gains the pv value, τ_d, and τ should be calculated first.

Variable	Blower at 15	Blower at 30	Values Calculated	PID Gains Calculated
τ_d	0.003241	0.00318	**Change In PV: 0.865956**	**P = 1.340832**
τ	0.036667	0.03650	**Change In Temp: 0.012**	**I = 0.006482**
pv	25.90342	25.03747	**K Value: 0.189617**	**D = 0.001621**

Table 2: Open Loop Method Calculations

As the blower opening input was set to 15 degree, the temperature out valued at 28.966 under 1 volt. The blower opening input was later changed to 30 degree resulting to an decrease in temperature. The change in temperature was calculated to be 0.012. The pv value at 15 degree was calculated to be 25.90342, and at 30 degree the pv was calculated to be 25.03747. The change in pv was 0.865956. The K value is the ratio between the change in temperature / change in PV.

Controller	kp	T_I [min]	T_D[min]
P	$100\frac{KT_d}{T}$	-	-
PI	$100\frac{KT_d}{T}$	3.33*Td	-
PID	$80\frac{KT_d}{T}$	2.00*Td	0.50*Td

Figure 8: Open Loop Quarter Decay Ratio Values

Using the ratios in Figure 8, the PID gains were calculated. Results of the following PID gains can be reviewed in Section 5 of this report. The temperature threshold was set to 30 degree in this experiment.

4.2 Closed Loop Method

To tune a PID using the closed loop method, the process should be in steady-state oscillation which makes this method sometimes unsuitable for real life processes. This should be achieved using a P controller by incrementing the Kp value with small increments. The plant should be in automatic. After process is in steady-state oscillation, the period of oscillation (P_u) and the amplitude of oscillation (K_u) is calculated (Figure 13 was used to calculate the values). The procedure to execute this method can be reviewed in the Appendix. The PID gains are calculated using the following ratios:

Controller	kp	T_I [min]	T_D[min]
P	0.5 Ku	-	-
PI	0.45Ku	Pu/1.2	-
PID	0.6Ku	0.5Pu	0.125Pu

Figure 9: Closed Loop Quarter Decay Ratio Values

Using a temperature threshold of 30 and a P value of 7.565, the P_u was calculated to be 0.012 and the K_u was calculated to be 0.012. Using the ratios in Figure 9 the PID gains were calculated (P = 1.321, I = 0.006, D = 0.0015).

5 Results

5.1 Manual Control Results

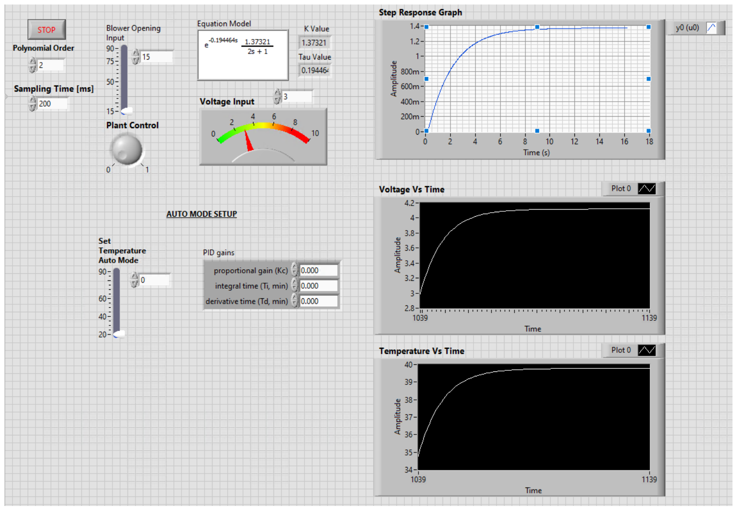

Figure 10: Manual Control Results

5.2 Open-Loop Control Results

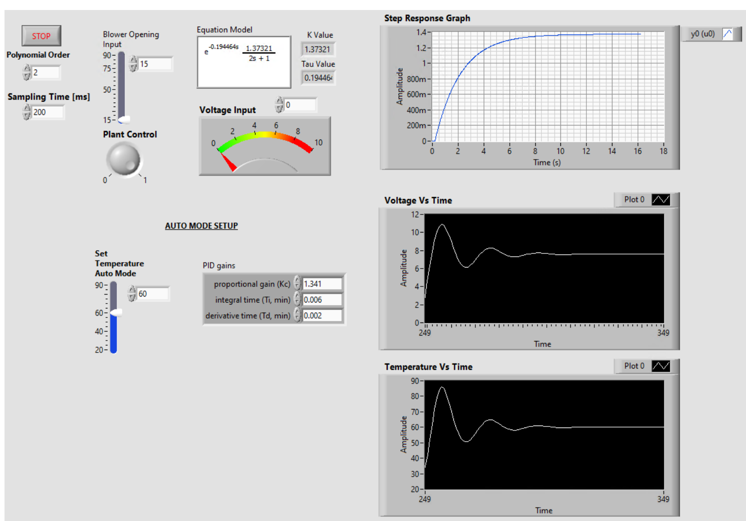

Figure 11: Open-Loop Control Results

5.3 Closed-loop Control Results

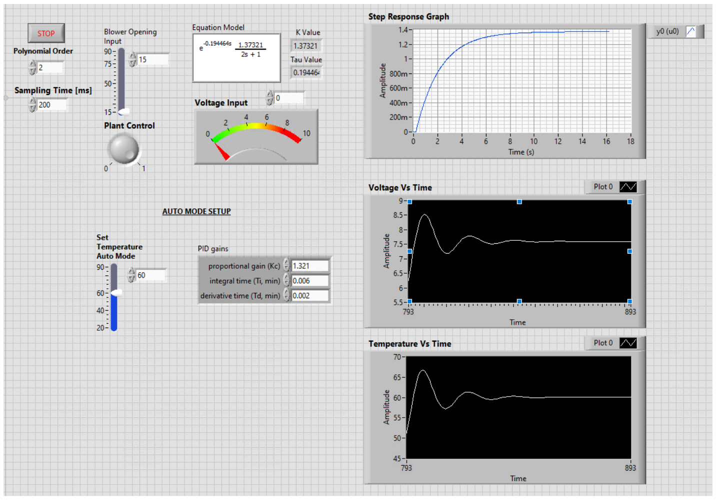

Figure 12: Closed-Loop Control Results

5.4 Results Analysis

Figure 10 presents the manual controller. The blower opening input was set to 15 degree. The voltage input was set to 3 volts. The amplitude of the temperature Vs time is approximately 40 degree which is close to the data presented in Table 1. In Figure 10, the settings used was displayed to prove that manual control functionality works.

Figure 11 presents the auto controller using open-loop method. The blower opening input was set to 15 degree. This controller uses a PID gains of 1.341, 0.006, and 0.002. The threshold temperature was set to 60 degree, the PID was able to achieve that threshold with an overshoot of 20%, an undershoot of 12%, and an estimated settling time of 60ms. Figure 12 uses the closed-loop method. This controller uses a PID gains of 1.321, 0.006, and 0.002 to achieve a temperature threshold of 60 degrees. This controller achieves a percent overshoot of 13%, a percent undershoot of 4%, and an estimated settling time of 70 ms.

6 Conclusion

Dynamic equations used to model heatflow can be successfully used to model systems with the end goal to regulate temperature. A control system capable of accurately reproducing a PT326 process trainer and regulating the temperature of air at the output vent was designed. Furthermore, both closed and open loop tuning were utilized to tune the system. In order to do so LabView was used; LabView provides a simple way to realistically simulate environment and plant interaction, obtain data, and implement control systems. The more difficult problem of developing a real time embedded controller of a hot air plan using a real time operating system (RTOS) is left for future experimentation.

Appendices

Steps for the open-loop method:

- Put the controller in manual mode (open loop), set the output (of the controller) to a nominal operating value, and allow the PV to settle completely. Record the PV and output (of the controller) values.
- Make a step change in the output (of the controller). Record the new output value
- Wait for the PV to settle.
- Multiply the measured values by the factors shown in Table 4 and enter the new tuning parameters into your controller. The table provides the proper values for a quarter-decay ratio. If you want less overshoot, reduce the gain, Kc.

Steps for the closed-loop method:

- Set both the derivative time and the integral time on your PID controller to 0.
- With the controller in automatic mode, carefully increase the proportional gain (K_c) in small increments. Make a small change in SP (set point) to disturb the loop after each increment. As you increase K_c, the value of PV should begin to oscillate. Keep making changes until the oscillation is sustained, neither growing nor decaying over time.
- Record the amplitude of the oscillations K_u (obtained from the smallest K_c that produces sustained oscillations).
- Record the period of oscillation (P_u) in minutes.
- Multiply the measured values by the factors shown in Table 3 and enter the new tuning parameters into controller.

Oscillation Graph for closed-loop method:

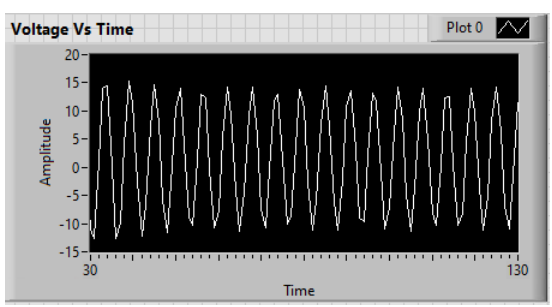

Figure 13: Closed Loop Control Oscillation