

PLC's and control

by Daniel Lennon

FILE	2792_DANIEL_LENNON_PLCS_AND_CONTROL_85309_1878775349.DOC X		
TIME SUBMITTED	02-MAY-2018 05:42PM (UTC+0100)	WORD COUNT	3029
SUBMISSION ID	88152099	CHARACTER COUNT	16053



PRIFYSGOL CYMRU

Y Drindod Dewi Sant

UNIVERSITY OF WALES

Trinity Saint David

SWANSEA - ABERTAWE

P.I.D control system.

PLC's and Control.

M3X0005.

By Daniel Lennon.

P144031.

Level 6.

Tutor Dr Gary Jones.

1. Abstract.

The purpose of this assignment was to produce a P.I.D. control system for a servo motor driven linear positioning system. The system carries a drilling head that is designed to drill holes in heavy metal sheets. In addition, it was a requirement to provide visual indication that the system is running and implement warnings and alarms for faults in operation. It was also required to generate an operator control screen which allows the system to be initiated or stopped and displays the required position, the control parameters and the current state.

A file was provided which contained the output data from an open loop response to a unity step change of the system input. It was to be used to generate, using MATLAB, the transfer function of the system and generate the parameters to produce a control system which operated within the following specification:

- Motor Drive – PWM.
- Max drive PWM output 85%.
- Warning if PWM output is greater than 80% of max.
- Alarm if PWM output is greater than 85% of max.
- Absolute Max response time (to steady state) 20secs.
- Absolute Max over-shoot to a step-change 5%.
- Warning if output exceeds 75% of max.
- Warning if position is greater than 80% of max.
- Warning if position is less than 20% of max.
- Alarm if position is greater than 90% of max.
- Alarm if position is less than 10% of max.

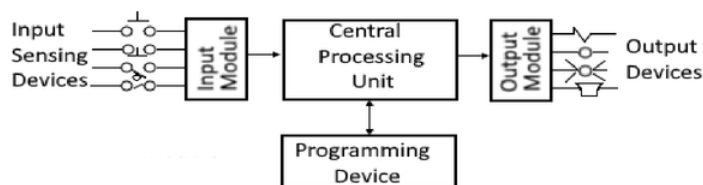
The system was designed using the Siemens TIA13 software developed to run on a S7-1200 (model 1214) PLC. This was done using the ladder logic method to generate a user interface which displays on the main screen the measured variable, the controller set point, the Values of Kp, Ki and Kd and an alarm indicator that shows when an alarm occurs. Also included in the design were two additional screens, one to display the details of the alarms or warnings and a third engineer screen to allow for adjustments to the warning and alarm parameters.

Contents

1. Abstract.....	2
2. Introduction.....	5
3. proportional–integral–derivative (PID) controller.....	6
4. Pulse Width Modulation (PWM).....	7
5. user interfaces for the PLC.....	8
6. the ladder logic program.....	11
7. generation of Kp, Ki and Kd values using MATLAB.....	17
8. Conclusion and potential future enhancements.....	23
9. References.....	24

2. Introduction.

A programmable Logic Controller (PLC) is an industrial digital computer. This type of machine is used in many different areas of industry and is designed to perform in harsh conditions such as extreme temperatures and high levels of moisture and dust. A PLC works in much the same way as a PC, having inputs that are processed by the control unit and outputs. The inputs of a PLC are generally sensors or switches, the control unit being the CPU which stores the control program and processes and decodes the input signals which are then sent to the outputs which are actuators. The signals processed by the control unit can be either digital or analogue but which ever they are they need to be compatible with the type of actuators being used meaning that there could be the need for conversion to or from digital to analogue [1]. The PLC was invented by Richard Morley in the late 1960's, designed to provide the same functions as relay logic systems. Relay systems at the time tended to fail quite regularly and create delays in the processes meaning that technicians would then have to troubleshoot an entire wall of relays to find and fix the problems. Since the invention of the PLC it has become an integral part of many industrial sectors and the operation of these devices have been designed for simplicity and accuracy. There are five main programming languages that are used by PLC's, the main one being that used in this assignment, ladder logic. The other four programming languages are function block diagram (FBD), Structured Text (ST) which is a high-level language that uses sentence commands, Instruction list (IL) which is a low-level language with functions and variables and Sequential Function Chart (SFC) language which is a method of programming using complex control systems[2]. Below is a block diagram of a PLC.



PLC Block Diagram

Figure 1. PLC block diagram.

3. proportional–integral–derivative (PID) controller.

PID controllers are used to bring things to a desired level, such as the speed of a motor, using the PID's 3 components Proportional, Integral & Derivative. With these three components a user can adjust how quickly and cleanly desired levels are achieved. A PID controller continuously calculates an error value as the difference between a desired setpoint (SP) and a measured process variable (PV), the actual position, and then applies a correction factor based upon the proportional, integral and derivative terms. These three terms also known as modes of control, are mathematical terms used to modulate and control the ability of the system to reach and maintain its setpoint and avoid overshoot and errors. A simplified block diagram of what the PID controller does is shown below [3].

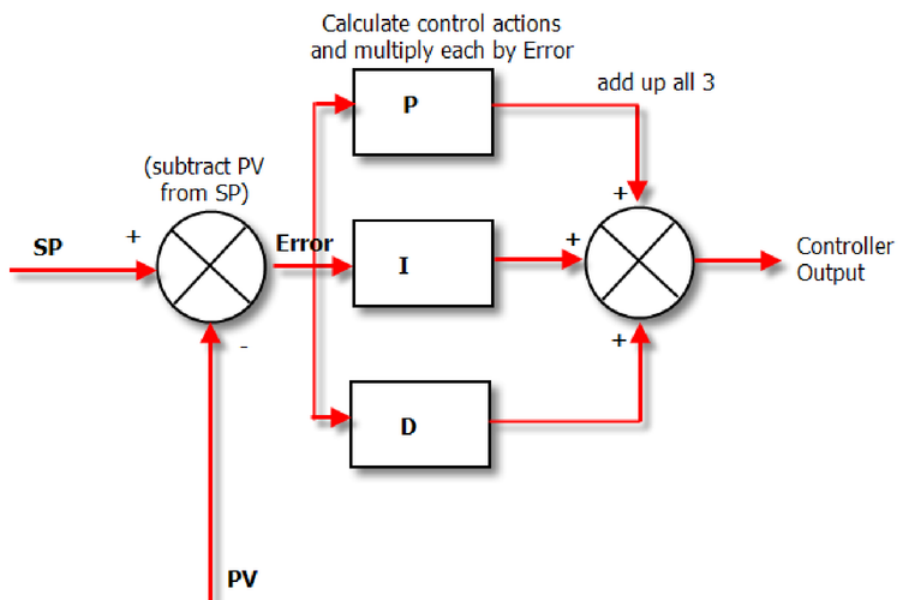


Figure 2. P.I.D function diagram

In this assignment these three terms are used in conjunction with the provided file to define the Kp, Ki and Kd values using MATLAB to apply to the PLC.

4. Pulse Width Modulation (PWM).

Pulse Width Modulation (PWM) is a technique for obtaining analogue results using digital signals. Digital pulses are used to create a square wave, a signal that switches between on and off. This on-off pattern can simulate voltages in between fully on and off by varying the portion of time the signal spends on to the time the signal spends off. This duration of time that the signal is on is called the pulse width and the period of time changes that the signal is on or off is the modulation. This method of creating an analogue signal from a digital one has several advantages over plain voltage (analogue) control. Firstly, when varying an analogue voltage in accordance to Ohms law you will change the power used and alter the current. When using PWM to simulate the analogue signal, because it is either off or on the voltages and current are either at their full value or off, hence no varying changes which leads to less energy wastage. Another important advantage of a digital signal over an analogue one is that if the signal has been affected by noise it can be recovered more easily. In analogue signals, when the signal is amplified to return to its original height the noise will get amplified as well [4].

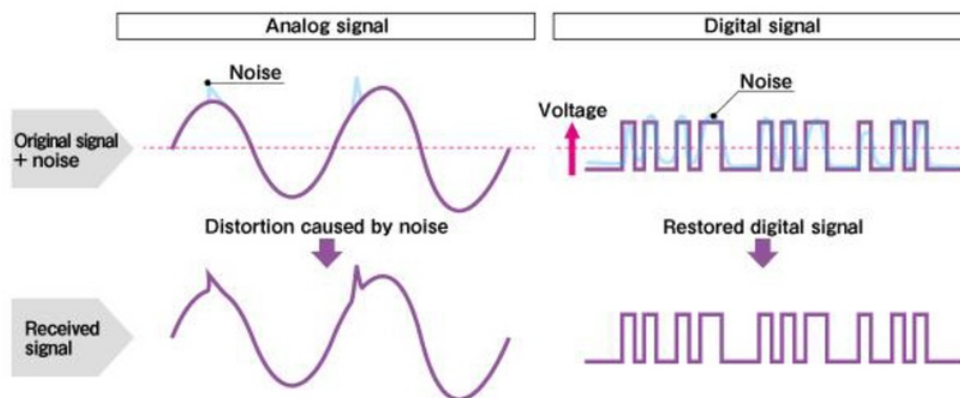


Figure 3. Comparison of noise on a digital and analogue signal.

In contrast, the disadvantage of using a digital signal to an analogue one is that an analogue signal unaffected by noise will be a more accurate representation of the true signal than a digital one.

5. user interfaces for the PLC.

As specified in the assignment brief a user interface which displays on the main screen the measured variable, the controller set point, the values of Kp, Ki, Kd and an alarm indicator showing that an alarm has occurred was required. In addition, two screens were included in the design, one to display the details of the alarms or warnings and a third engineer screen to allow for adjustments to the warning and alarm parameters. Following are screenshots and brief explanations of the interface screens that were designed.

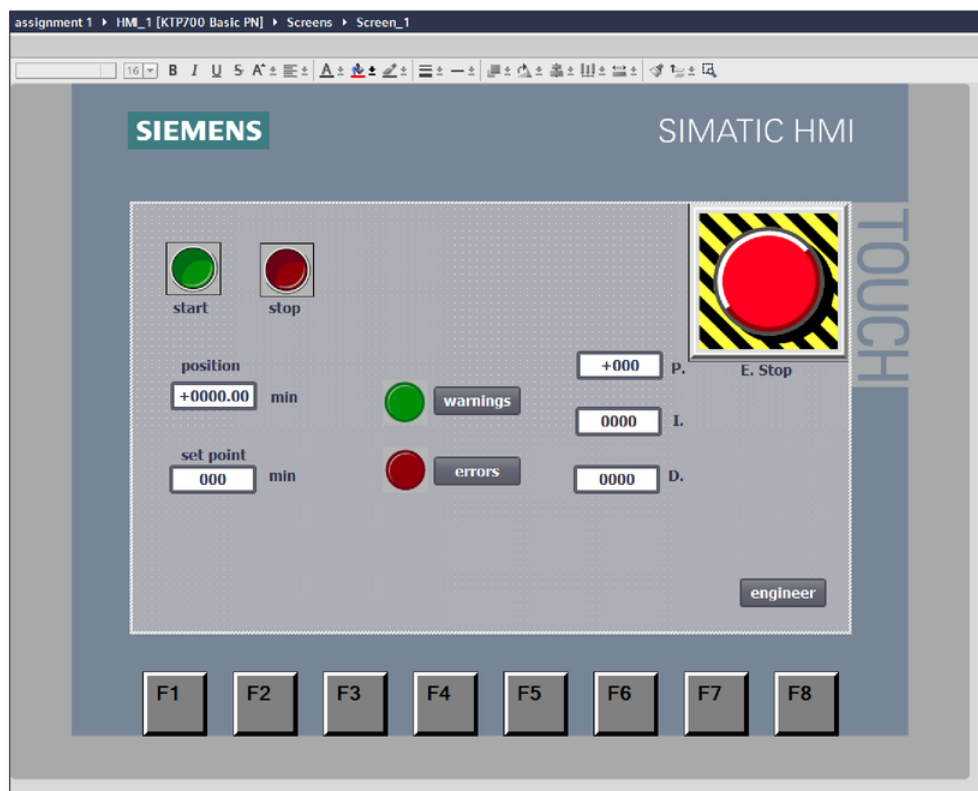


Figure 4. The main interface screen.

On this screen the position and setpoint can be set and the Kp, Ki and Kd values can be input. There are also start and stop buttons for operating the PLC, an emergency stop button in case of any unforeseen errors that may occur and error and warning buttons/lights to alert the user of

any issues that have arisen and allow the user to switch to the error screen for a more detailed understanding of what that error is. There is also a button in the right-hand corner that will take the user to the engineer screen to allow for adjustments in the parameters set for alarms and warnings.

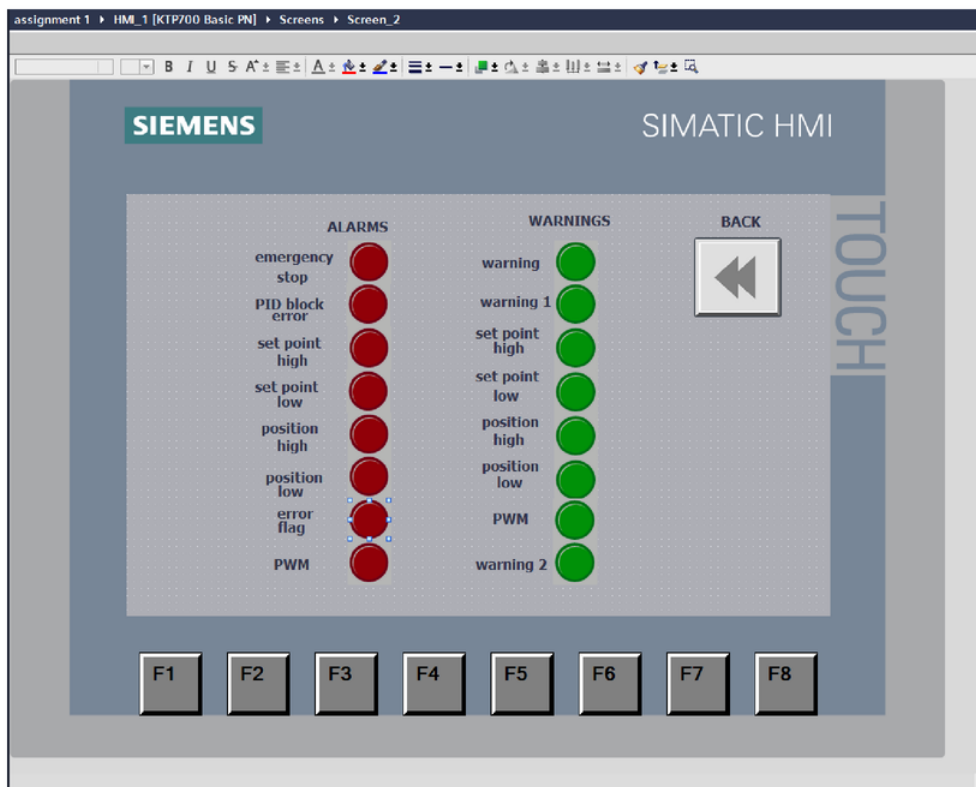


Figure 5. The warning and alarm screen.

This screen shows, when an error has occurred, what that error is, be it a warning or alarm in more specific detail than that shown on the main interface. there is also a back button to return the user to the main screen after the issue has been identified.

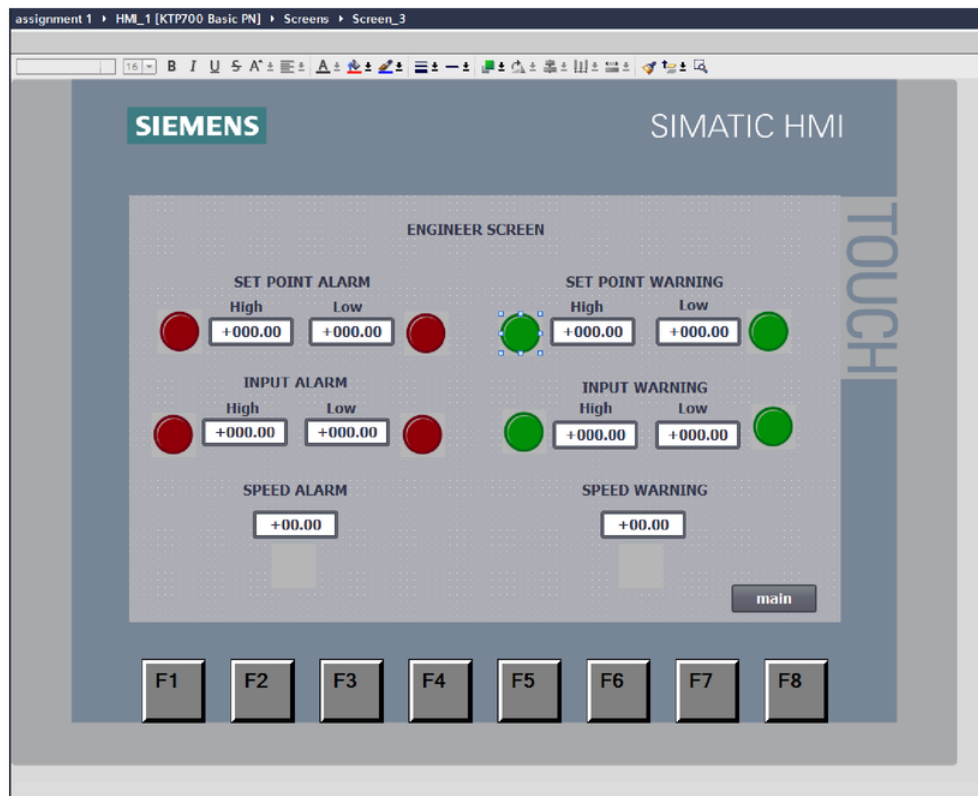


Figure 6. The engineer screen.

This last screen, the engineer screen allows for adjustment of the alarm and warning points. It allows for adjustments in the high and low warnings and alarms for the setpoint, the input and has a speed alarm and warning setpoint. There is also a button to return the user to the main screen.

All of the buttons and lights on these screens were associated, using the plc memory tags to the respective networks developed in the ladder logic program that was designed for the system. The association of these buttons/lights to the ladder logic program are explained in the next section of the report.

6. the ladder logic program.

In this section there is a screenshot of all the PLC tags created for the ladder logic program shown and the memory allocation of these tags is given. Also in this section the networks developed for the ladder logic program will be shown and descriptions of the functions and their association to the user screens will be presented.

	Name	Tag table	Data type	Address	Retain	Visibl...	Acces...	Comment
1	Green Button	Default tag table	Bool	%I0.6	☐	☒	☒	
2	Red Button	Default tag table	Bool	%I0.5	☐	☒	☒	
3	operate	Default tag table	Bool	%Q0.3	☐	☒	☒	
4	ERRORS	Default tag table	Byte	%MB100	☐	☒	☒	
5	ES error	Default tag table	Bool	%M100.0	☐	☒	☒	
6	PID error	Default tag table	Bool	%M100.1	☐	☒	☒	
7	SPH error	Default tag table	Bool	%M100.2	☐	☒	☒	
8	SPL error	Default tag table	Bool	%M100.3	☐	☒	☒	
9	PH error	Default tag table	Bool	%M100.4	☐	☒	☒	
10	PL error	Default tag table	Bool	%M100.5	☐	☒	☒	
11	PWM error	Default tag table	Bool	%M100.6	☐	☒	☒	
12	error flag	Default tag table	Bool	%M100.7	☐	☒	☒	
13	WARNINGS	Default tag table	Byte	%MB101	☐	☒	☒	
14	warning	Default tag table	Bool	%M101.0	☐	☒	☒	
15	warning1	Default tag table	Bool	%M101.1	☐	☒	☒	
16	SPH warning	Default tag table	Bool	%M101.2	☐	☒	☒	
17	SPL warning	Default tag table	Bool	%M101.3	☐	☒	☒	
18	PH warning	Default tag table	Bool	%M101.4	☐	☒	☒	
19	PL warning	Default tag table	Bool	%M101.5	☐	☒	☒	
20	PWM warning	Default tag table	Bool	%M101.6	☐	☒	☒	
21	warning 2	Default tag table	Bool	%M101.7	☐	☒	☒	
22	SPH alarm	Default tag table	Real	%MD0	☒	☒	☒	
23	SPL alarm	Default tag table	Real	%MD4	☒	☒	☒	
24	IPH alarm	Default tag table	Real	%MD8	☒	☒	☒	
25	IPL alarm	Default tag table	Real	%MD12	☒	☒	☒	
26	SPDH alarm	Default tag table	Real	%MD16	☒	☒	☒	
27	SPH warn	Default tag table	Real	%MD24	☒	☒	☒	
28	SPL warn	Default tag table	Real	%MD28	☒	☒	☒	
29	IPH warn	Default tag table	Real	%MD32	☒	☒	☒	
30	IPL warn	Default tag table	Real	%MD36	☒	☒	☒	
31	SPDH warn	Default tag table	Real	%MD40	☒	☒	☒	
32	millimeters	Default tag table	Real	%MD102	☐	☒	☒	
33	analog in	Default tag table	Int	%IW64	☐	☒	☒	
34	warning flag	Default tag table	Bool	%M103.1	☐	☒	☒	
35	warn light	Default tag table	Bool	%M106.0	☐	☒	☒	
36	error light	Default tag table	Bool	%M106.1	☐	☒	☒	
37	<Add new>				☐	☒	☒	

Figure 7. The PLC tags and memory locations.

The above figure shows all of the tags used in producing the user screens and ladder logic program. Tag number 4, the ERROR tag is one byte containing the 8 error bits associated with the error light on the main operator screen and the subsequent errors on the warning and alarm screen as shown in figure 4. Tag number 13, the WARNING tag is again a byte containing the next 8 bits

associated with the warning light on the operator screen and the warnings on the warning and alarm screen. Tags 22 through to 31 are associated with the setpoint, input and speed warnings and alarms. These tags, as can be seen are retained, this means that they do not reset and retain the values that have been input for them when the system resets after an alarm or warning has been activated and then dealt with.

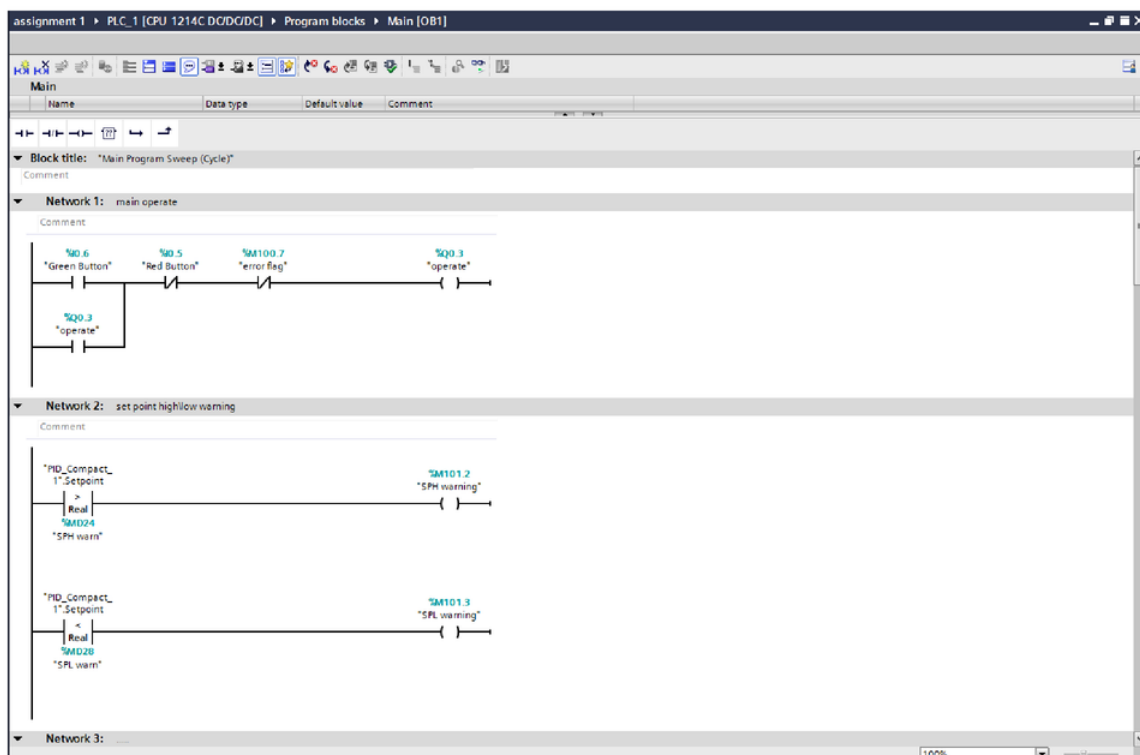


Figure 9. Networks 1 and 2 of the ladder logic program

The rung in network 1 is the main operate function, this basically dictates that when there is no error flag set and the red button has not been pushed that, when the green button is pushed the system will operate and if either red button is pushed or the error flag is set the system will stop. The emergency stop button on the operator screen if pushed as well as any alarm becoming active will set the error flag. Network 2 contains 2 rungs, the first contains a conditional instruction that depicts the setpoint high warning, if this bit is set it operates the setpoint high warning which will operate the high warning light on the screen in figure 5. The second rung contains a conditional

instruction that depicts the setpoint low warning, if this bit is set it operates the setpoint low warning which will turn on the low warning light on the screen in figure 5.

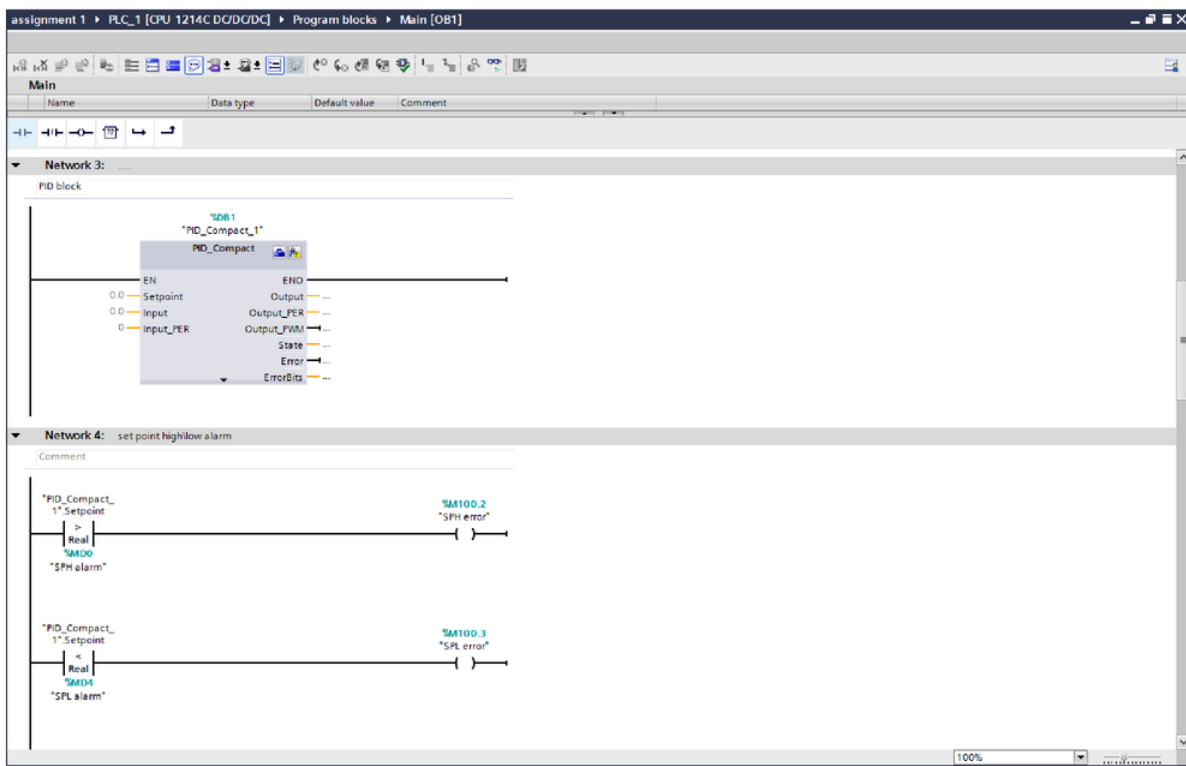


Figure 10. Networks 3 and 4 of the ladder logic program.

The rung in network 3 is the P.I.D block, this block is where the data developed in MATLAB that was mentioned earlier and will be discussed further, later in this report would be entered.

Network 4 contains 2 rungs, the first contains a conditional instruction that depicts the setpoint high alarm, if this bit is set it operates the setpoint high error which will operate the alarm light on the screen in figure 5. The second contains a conditional instruction that depicts the setpoint low alarm, if this bit is set it operates the setpoint low error which will operate the warning light on the screen in figure 5. If either of these bits are set it will stop the system from operating.

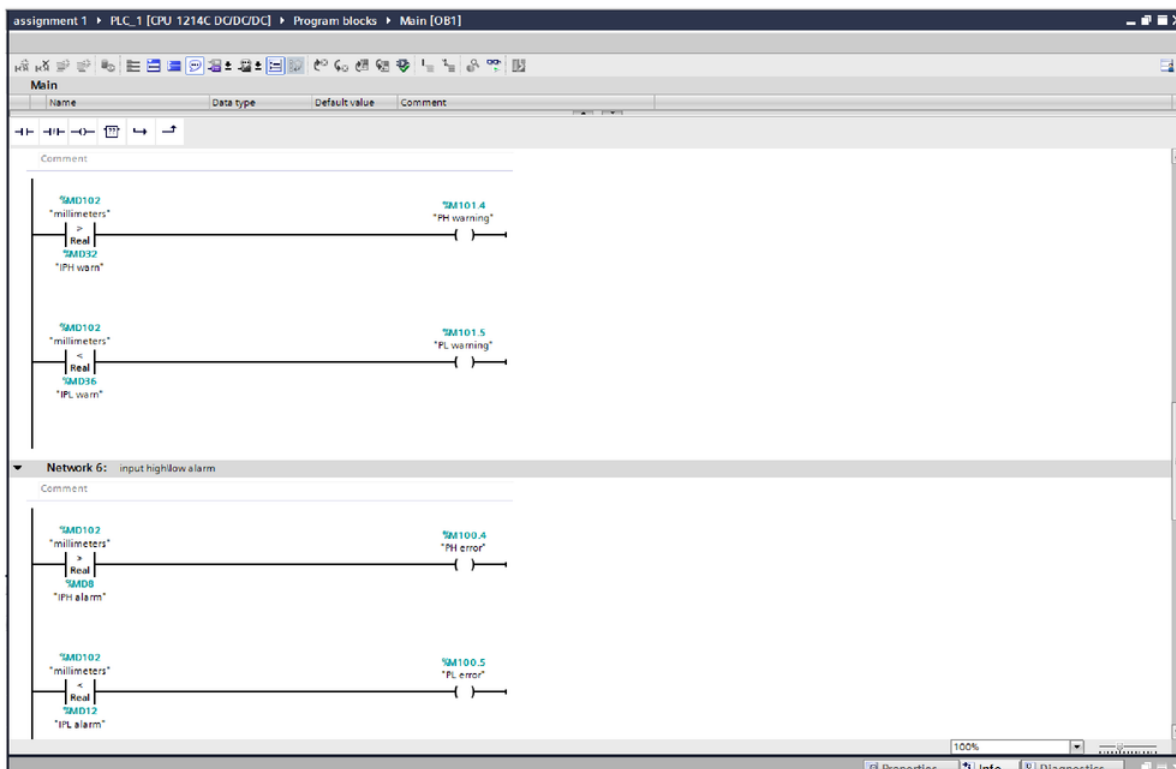


Figure 11. Networks 5 and 6 of the ladder logic program.

The first rung in network 5 has a conditional instruction that when set operates the position high warning. This instruction is in relation to the input that can be set on the operator screen and is measured in millimeters. This will turn on the associated warning light on the warning and alarm screen. The second rung in network 5 has a conditional instruction that when set operates the position low warning. This instruction is also in relation to the input that can be set on the operator screen and will turn on the associated warning light on the warning screen. The 2 rungs in network 6 operate the position high and low alarms and will turn on the associated alarm lights and stop the system when set.

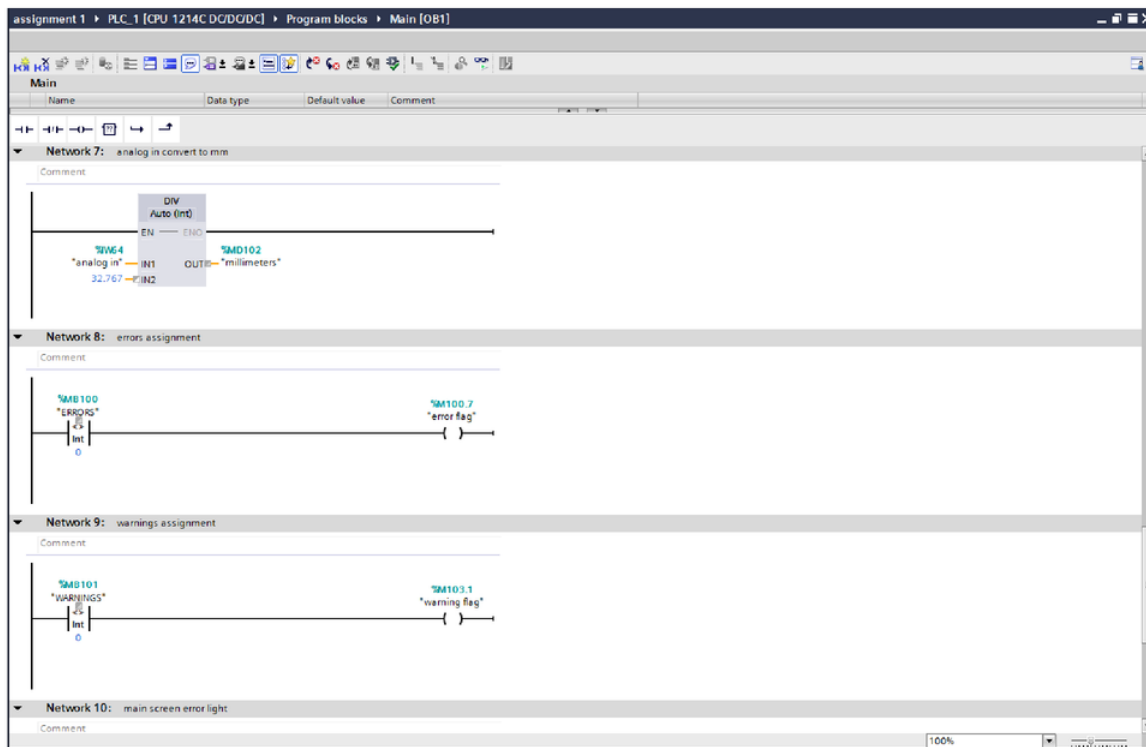


Figure 12. Networks 7-9 of the ladder logic program

Network 7 converts the analogue input that is set on the operator screen through the use of a division function block to millimeters. Network 8 has a conditional instruction that when set will operate the error flag function, stopping the system, this will be set if any of the errors contained within the ERROR byte are set or the emergency stop is activated. Network 9 has a conditional instruction that will operate the warning flag if any of the warnings within the WARNINGS byte are set and turn on the associated warning lights on the operator and warnings screens.



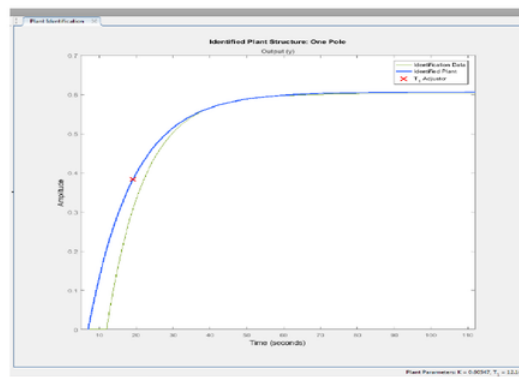
Figure 13. Networks 10 and 11 of the ladder logic program.

Network 10 has a conditional instruction that when set will operate the warning light on the operate screen if any of the warnings inside the WARNING byte are set. Network 11 has a conditional instruction that will operate the error light on the operator screen when any of the errors inside the ERROR byte occur.

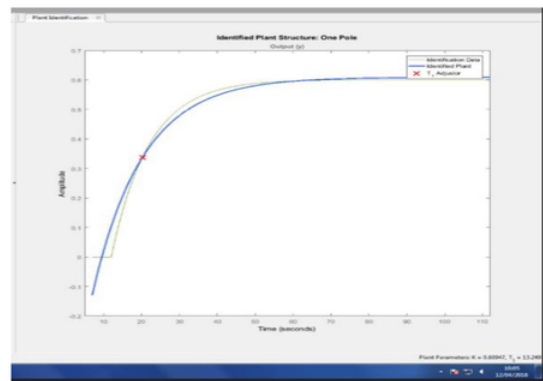
These 12 networks complete the ladder logic program that was designed for this assignment. All of the parameters that could be tested for this design were tested and all of the associated warning and alarm lights worked as expected as well as the emergency stop, and when the setpoints were altered and the system was tested it still worked as expected.

7. generation of K_p , K_i and K_d values using MATLAB.

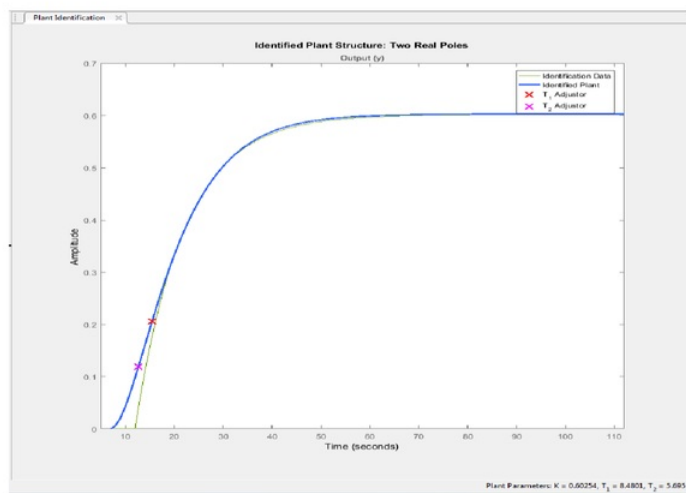
In order to generate the values needed for this assignment, an output file of the system to a step change for a plant was provided. This was used to generate the transfer function and subsequently the K_p , K_i and K_d values using MATLAB. The transfer function was developed by using the file provided with the pidtool function and trying to map the response curve as closely as possible. This was done using one real pole, two real poles, underdamped pair and underdamped pair and real pole. Following are the graphs created when using the PID tuner.



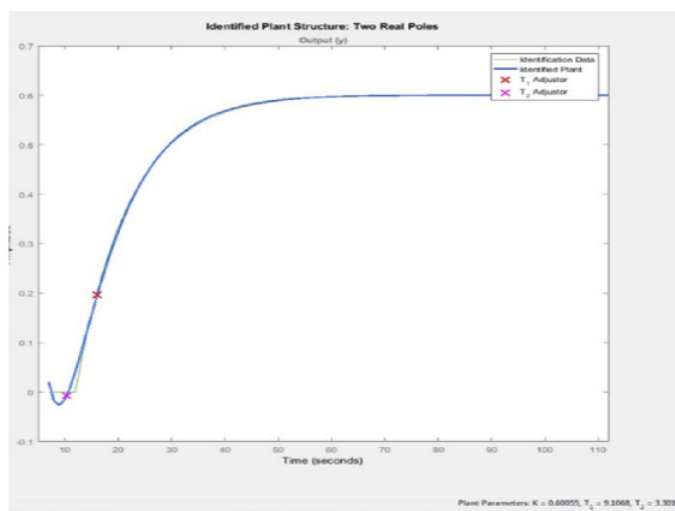
This graph shows the response curve using one real pole.



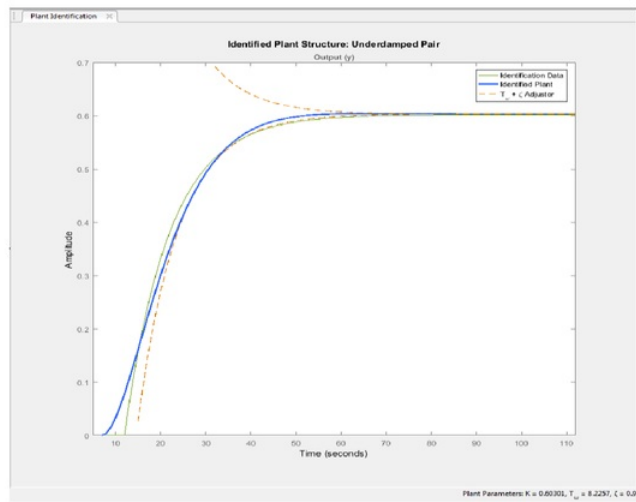
This graph shows the response curve using one real pole and the auto correct function.



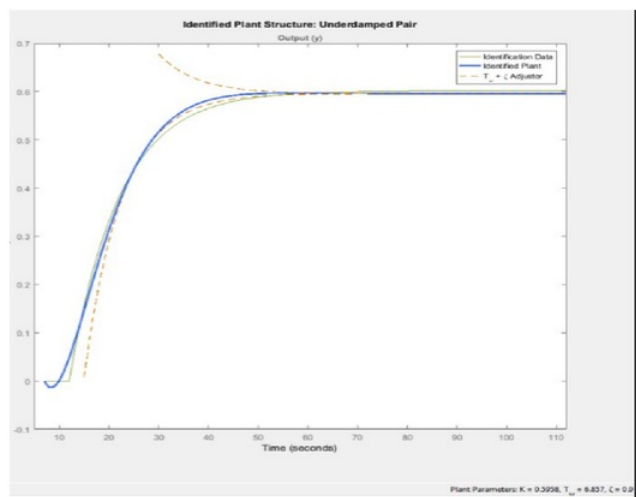
This graph shows the response curve using two real poles.



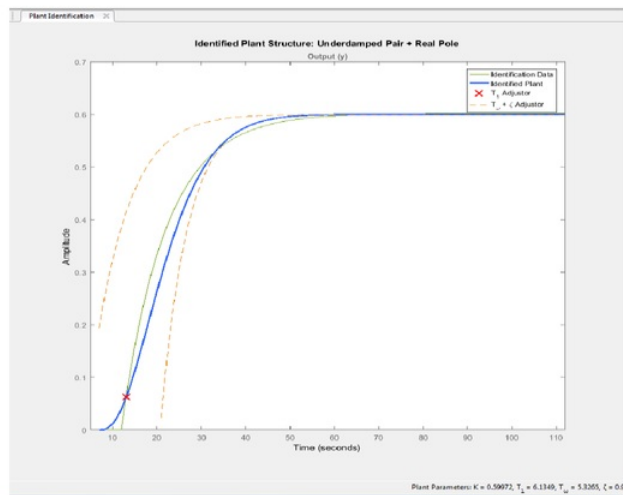
This graph shows the response curve using two real poles and the auto correct function.



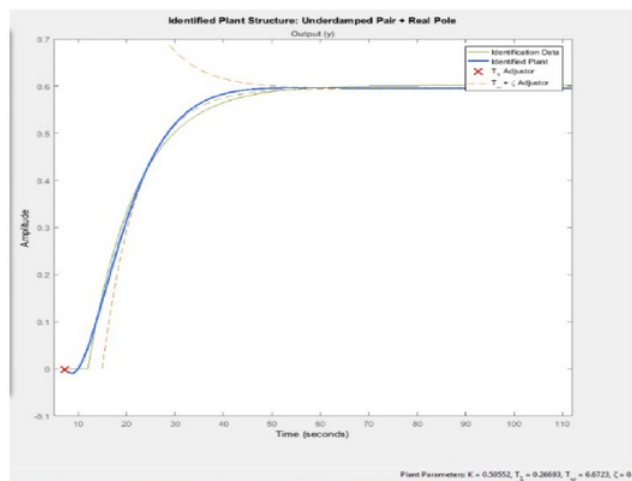
This graph shows the response curve using an underdamped pair.



This graph shows the response curve using an underdamped pair with the auto correct function.



This graph shows the response curve using underdamped pair + real pole.

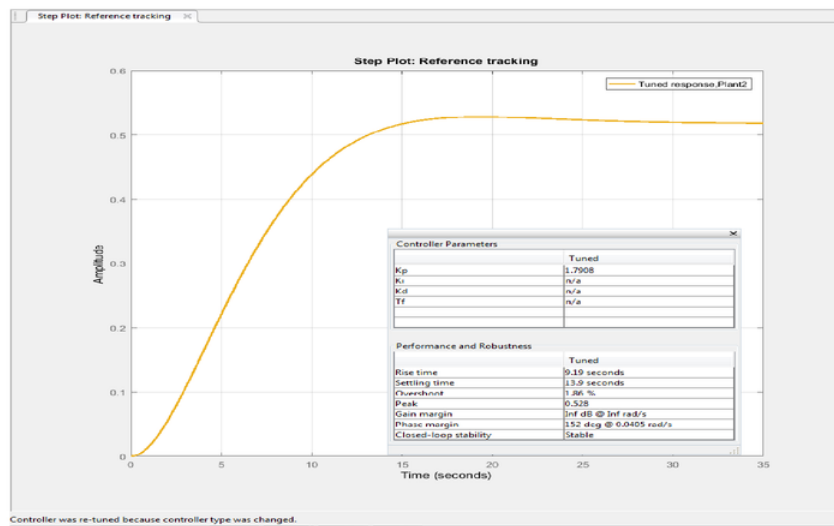


This graph show the response curve using underdamped pair + real pole and using the auto correct function.

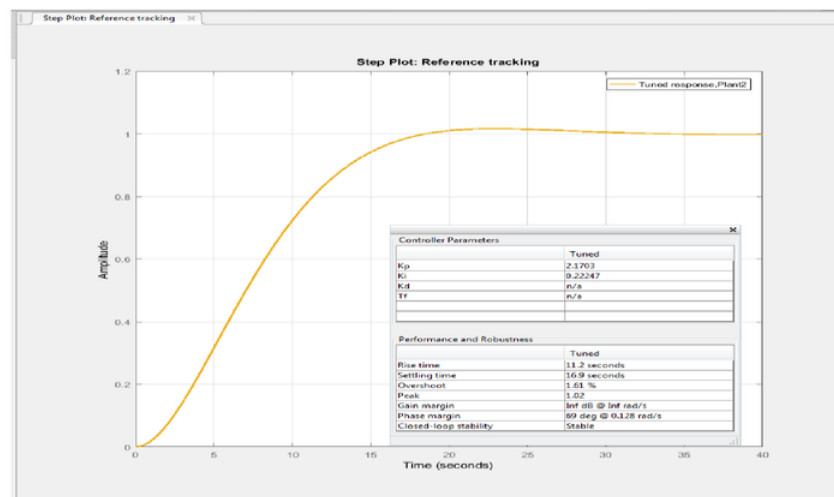
After observing the response curves, it was decided that the one that closest matched that provided in the file was by using two real poles. This produced the following transfer function.

$$\frac{K}{(T_1 s + 1)(T_2 s + 1)}$$

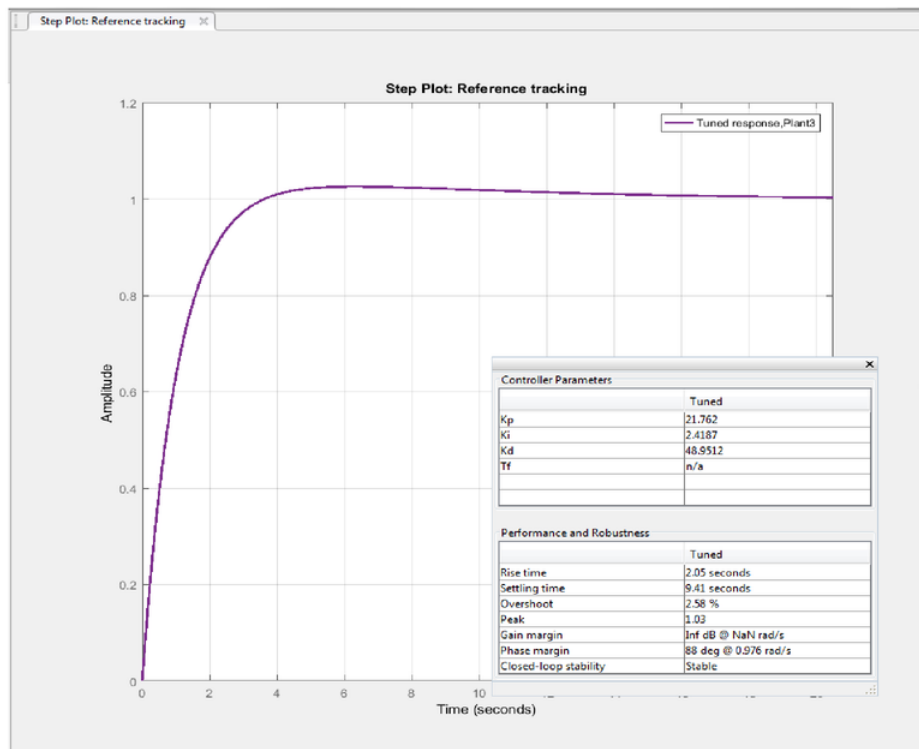
This was then used to develop first the Kp, then Kp and Ki, and then Kp, Ki and Kd values and to ensure that the rise and settling times and the overshoot values were within the parameters set for the assignment.



This is the graph that was produced using just the proportional function.



This is the graph that was produced using the proportional and integral functions.



This graph was produced using the proportional, integral and derivative functions. After some manual adjustment of the gain, the rise time, settling time and overshoot all sat comfortably within the parameters required for this assignment. This provided the required values of Kp, Ki and Kd that can be seen in the above graph that were needed for the specifications of the assignment, creating the output of the system, a PWM signal for controlling a motor.

8. Conclusion and potential future enhancements.

During this assignment an introduction to PLC's and the corresponding ladder logic programming was presented. Brief discussions on PID controllers and PWM have been made with their association to PLC's. In order to improve the overall design of this system a deeper understanding of ladder logic and the mathematical theory behind PID controllers would be needed. The system designed, when tested worked within the parameters prescribed for the assignment but could have been improved with a better understanding of the subject. The designs of the user screens could have been made more clear and easier to use and more ways for the user to control the different aspects of the system could have been introduced. It is also believed that the ladder logic program that was created could have been more precise and compact if a better understanding of the language was obtained. Overall the system worked and it is believed that it satisfied the requirements but could definitely be improved with further investigation and understanding.

9. References.

- [1] "What is PLC ? Programmable Logic Controller - Unitronics." [Online]. Available:
<https://unitronicsplc.com/what-is-plc-programmable-logic-controller/>. [Accessed: 01-May-2018].

- [2] "Engineering Essentials: What Is a Programmable Logic Controller?" [Online]. Available:
<http://www.machinedesign.com/engineering-essentials/engineering-essentials-what-programmable-logic-controller>. [Accessed: 01-May-2018].

- [3] "PID for Dummies - Control Solutions." [Online]. Available:
https://www.csimn.com/CSI_pages/PIDforDummies.html. [Accessed: 01-May-2018].

- [4] "Communication Systems: Advantages of Digital Transmission." [Online]. Available:
<http://ecedunia.blogspot.co.uk/2016/03/advantages-of-digital-transmission.html>. [Accessed: 01-May-2018].

PLC's and control

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

Instructor

60/100

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

GRADING FORM: SMU GENERIC_GJ

DANIEL LENNON**59.65****RESEARCH-INFORMED LITERATURE**

The literature review is clear but limited with reasonable background information but would have been improved with a more detail, along with a PWM advantage/disadvantage table. 45%

6.75**KNOWLEDGE AND UNDERSTANDING**

The ladder logic program is clear and documented. 68%

17**ANALYSIS**

The mechanism for generating the transfer function from the plant input output response curve is good. Your method is a ok but vaguely documented with good graphics. 58%

14.5**PRACTICAL APPLICATION AND DEP.**

Your screens are functional and uncluttered and



well documented. All screens are nicely laid out and make it easy to identify faults/warnings and allow for user input and navigation. Limited attempt made at future improvements.62%

12.4

SKILLS FOR PROFESSIONAL PRAC.



The report is reasonably presented with the author using sufficient images and illustrations to allow the reader to digest information readily.60%

9