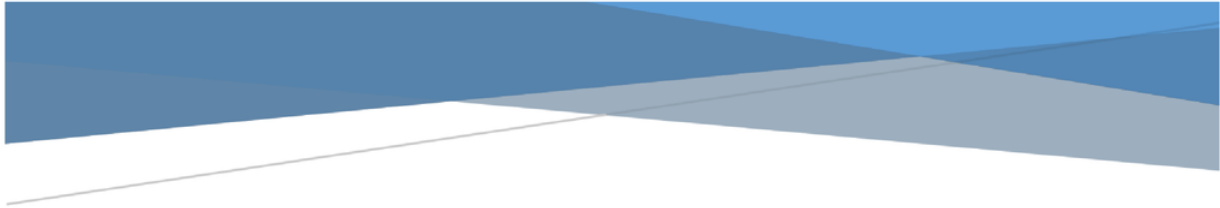


PID Control PLC Assignment

by Adam Peter Davies

FILE	2784_ADAM_PETER_DAVIES_PID_CONTROL_PLC_ASSIGNMENT_85309_555988881.DOCX		
TIME SUBMITTED	02-MAY-2018 10:20PM (UTC+0100)	WORD COUNT	8281
SUBMISSION ID	88172999	CHARACTER COUNT	56822



(M3X0005) PLC AND CONTROL

PID Control system

Abstract

This report outlines the design of a PID controller which is applied to a linear position system. The positional system is a drill which forms holes in metal sheeting. The system is designed using a PLC. The PLC used is a Seimens S7 1214. The drill is controlled via a PWM output in which the advantages and disadvantages of this signaling is explored. Screens were designed using for a HMI with the PLC ladder logic providing logic control. The screens included all display for warnings and errors and user input for PID parameters and set point data. Overall the designed system meets the design specification with future recommendations including improved graphics and user login.

Lecturer: Dr. Gary Jones

Adam Davies (P147908)
P147908

Table of Contents

1. Introduction	7
2. Background Information	7
2.1. Siemens S7-1200	7
2.2. PID Controller	8
2.2.1. Proportional Control	10
2.2.2. Integral Control	10
2.2.3. Derivative Control	10
2.2.4. Tuning.....	11
2.3. Pulse Width Modulation (PWM)	13
2.3.1. PWM Advantages	13
2.3.2. PWM Disadvantages	14
3. Design Specification	15
4. System Design.....	16
4.1. System Block Diagram.....	16
4.2. Operation Flowchart	17
4.3. Operator Flowchart.....	20
5. System Implementation	21
5.1. Screen Design	22
5.1.1. User Screen	23
5.1.2. Engineer Screen	26
5.1.3. Errors and Warnings Screen.....	29
5.2. Ladder Logic Design.....	32
5.2.1. Network 1.....	32
5.2.2. Network 2.....	33
5.3. PID Controller Design	35
6. User Documentation	64
6.1. Error Codes.....	64
6.2. Warning Codes	65
7. Improvements and Recommendations	66
7.1. Speed and Positional Data Graphs	66
7.2. User Login.....	67
7.3. Drill Position Display.....	67
7.4. Error and Warning Logging	67
7.5. Proximity Sensors.....	Error! Bookmark not defined.
8. Conclusion.....	68

9. References	69
10. Bibliography.....	71

Table of Figures

Figure 1: Siemens S7-1200 with 1214 CPU [3]	8
Figure 2: PID Controller Block Diagram	9
Figure 3: PID Control Output Graph [7]	9
Figure 4: Zeigler-Nichols Method Amplitude Principle [14].....	12
Figure 5: PWM Duty Cycle.....	13
Figure 6: System Block Diagram	16
Figure 7: Operation Flowchart Main	17
Figure 8: Error Check Operation	18
Figure 9: Warning Check Operation	19
Figure 10: Operator Flowchart.....	20
Figure 11: PLC Tags.....	21
Figure 12: Error Byte Operation.....	22
Figure 13: User Screen	23
Figure 14: Start Button.....	24
Figure 15: Stop Button	24
Figure 16: E-Stop	24
Figure 17: Error Indicator	24
Figure 18: Warning Indicator	25
Figure 19: Current Position Output.....	25
Figure 20: Set Position Output	25
Figure 21: Kp Input.....	25
Figure 22: Ki Input	26
Figure 23: Kd Input.....	26
Figure 24: Engineering Screen.....	26
Figure 25: Error Indicator	27
Figure 26: Warning Indicator	27
Figure 27: Analogue Input Display	27
Figure 28: Home Button.....	28
Figure 29: Engineering Warning and Error Input Layout	29
Figure 30: Errors and Warnings Screen.....	29
Figure 31: Error Indicator and Error Flag Reset	30
Figure 32: Warnings Layout	31
Figure 33: Alarms/Error Layout.....	31

Figure 34: Network 1.....	32
Figure 35: Network 2.....	33
Figure 36: Network 3.....	33
Figure 37: Network 4.....	34
Figure 38: Network 5.....	34
Figure 39: Network 6.....	35
Figure 40: Network 7.....	36
Figure 41: Network 8.....	37
Figure 42: Network 9.....	38
Figure 43: Network 10.....	39
Figure 44: Network 11.....	40
Figure 45: Network 12.....	40
Figure 46: Initial Value Allocation	41
Figure 47: Output File Import Data.....	42
Figure 48: Workspace Values.....	42
Figure 49: Step Response Import.....	43
Figure 50: Single Pole Manual.....	44
Figure 51: Automatic Single Pole	45
Figure 52: Single Pole Transfer Function.....	45
Figure 53: Two Real Poles Manual	46
Figure 54: Two Real Poles Automatic.....	47
Figure 55: Two Real Poles Transfer Function	47
Figure 56: Underdamped Pair Manual.....	48
Figure 57: Underdamped Pair Automatic	49
Figure 58: Underdamped Pair Transfer Function	49
Figure 59: Underdamped Pair and Real Poles Manual	50
Figure 60: Underdamped Pair and Real Pole Automatic	51
Figure 61: Underdamped Pair and Real Pole Transfer Function.....	51
Figure 62: Proportional Only Control.....	52
Figure 63: Proportional Control Performance	53
Figure 64: PI Control.....	54
Figure 65: PI Control Performance.....	55
Figure 66: PID Control	56
Figure 67: PID Control Performance	57
Figure 68: Start Function Testing	58
Figure 69: Stop Function Testing.....	58

Figure 70: Analogue Positional Test..... 59

Figure 71: E-Stop Activation..... 60

Figure 72: E-Stop Alarm Activation 60

Figure 73: Setting Error and Warning Thresholds..... 61

Figure 74: Set Point High Warning Activated 61

Figure 75: Position Greater than 900mm 62

Figure 76: Error Position High 62

Figure 77: Error Flag Set 63

Figure 78: Error Flag Cleared After back is Activated 63

Figure 79: Speed and Position Graphs 66

Figure 80: Drill Example Graphic..... 67

Table of Tables

Table 1: Zeigler-Nichols Method Table of Values 12

Table 2: Errors and Warnings Specification 15

Table 3: Warning and Error Threshold Input Tags 28

Table 4: Warning Tag Layout..... 30

Table 5: Error Fault Codes 64

Table 6: Warnings Fault Codes..... 65

1. Introduction

This report outlines the design of a PID controller to be used for a linear positioning system driven via a servo motor. The intended application of the controller is for a drilling head which forms holes in metal sheeting. The final system is designed using a 'Siemens S7-1200' Programmable Logic Controller (PLC) paired with a Human Machine Interface (HMI) screen for user interface. Sections in the report include the background information which overviews all pertinent literature, the system design, system implementation and user documentation. All findings are concluded at the end of the report along with a list of recommendations and improvements to the current designed system.

2. Background Information

A PLC is an industrial form of microprocessor based controller which uses a programmable memory to implement functions, control inputs and outputs and perform arithmetic operations [1]. PLC systems are usually used in automation applications within rugged environments such as factory lines controlling related machinery. Due to their rugged build and built in input/output (I/O) interface they are preferred to PC based systems in the previous mentioned application [2].

PLCs are mainly programmed using ladder logic. Ladder logic is a diagrammatic representation of a program which is based upon relay circuits. Unlike normal programming languages ladder logic is not sequential with each rung operating in parallel. The PLC unit used in this design is the 'Siemens S7-1200' which is discussed in the following section.

2.1. Siemens S7-1200

The Siemens SIMATIC S7-1200 controller is a compact and versatile PLC unit produced for a variety of industrial applications (figure 1). The unit features integrated Input/Output (I/O), a range of communication interfaces and provide a variety of technological functions [3].

Another feature of this controller is the ability to connect to the 'Totally Integrated Automation Portal' (TIA Portal). This is software produced by Siemens to integrate and connect all controllers and related devices such as HMI displays together, thus allowing control and programming of all related units together. The S7-1200 unit allows many performance classes of CPU to be integrated to the controller. The CPU used in this design is the 1214.



Figure 1: Siemens S7-1200 with 1214 CPU [3]

The 1214 CPU is a compact built high performance CPU. Its main feature is it provides 24 integrated I/O. However the CPU has expandable and optional features in which it can feature one signal board, up to eight signal modules, up to three communication modules and six counters with three counting at a 100 kHz of frequency [3]. Although the majority of these features will not be implemented within this design they are common features used in industrial applications.

2.2. PID Controller

A Proportional Integral Derivative (PID) Controller is a closed loop control system which is used widely in industrial applications. The three terms used are combined to provide a control signal [4]. By using error feedback the control signal can be kept stable for a desired output.

The PID controllers input is the set point. In the application of controlling the drill the set point would be the position set by the operator. The output from the controller sets the drill to the desired position. A variable also referred to a 'Process Variable' provides the feedback to the system and the value within this variable is compared with the set point. Any difference between the signals produces an error [5]. With this error signal the controller provides the appropriate signal to keep the output at the set point position (figure 2).

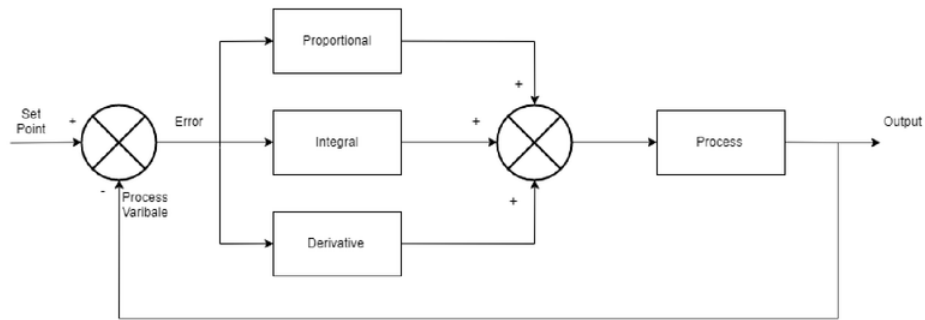


Figure 2: PID Controller Block Diagram

There are three main parameters considered with the output of a PID controller these are overshoot, rise time and settling time (figure 3). These are equally important and are dependent on the application. Overshoot is when the output exceeds the steady state value. Rise time is the time it takes for the response to reach 90% from 10% of the steady state response [6]. Settling time is the time taken for the error between the response and steady state response to drop to below 2% [6]. Different types of control parameters maximise or minimise these parameters.

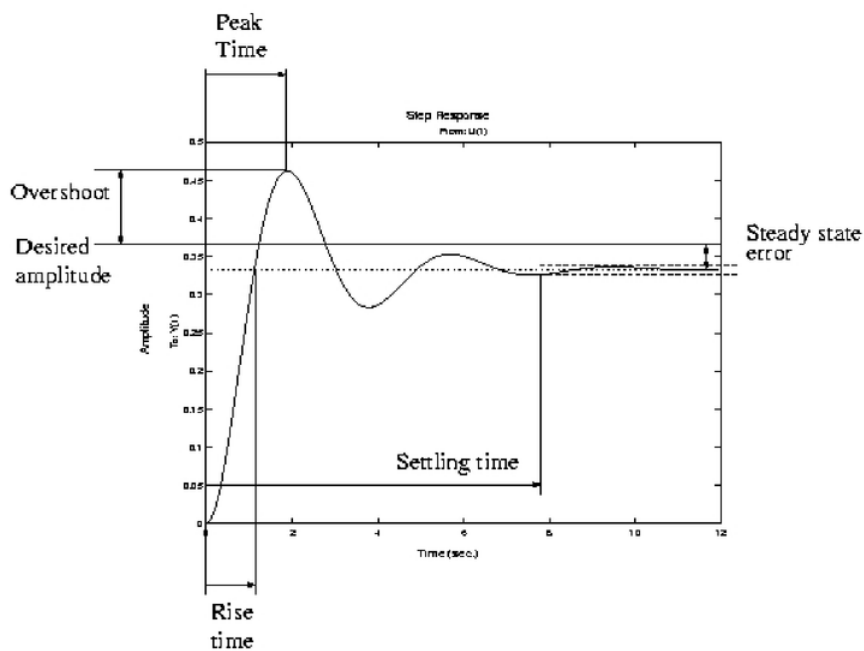


Figure 3: PID Control Output Graph [7]

2.2.1. Proportional Control

The proportional control in a PID controller is effectively the gain of the controller. It controls the output in proportion to the error. The gain of the controller is adjustable and is known as the 'Kp' parameter. Controllers which have a higher gain means that they will have an increased amount of proportional control for a given error signal.

Proportional control only has limitations. As a controller which has too much gain can lead to unstable control. This being due to the control loop can begin to oscillate causing an unstable signal [8]. Another limitation with proportional only controller is offset. Offset error cannot be prevented with solely proportional control [9].

2.2.2. Integral Control

The integral term in a PID controller controls the output signal so there is minimum error feedback [10]. It performs this by incrementing or decrementing the output signal so the error is reduced to eventually to zero [8]. The speed of the integral controller is given by the time setting given.

Limiting factors in integral control is provided by changing this integral time setting. By setting the integral time too high the controller may perform at a slow speed, whereas if it is set too short the controller loop may oscillate and make the signal unstable [8] [11].

In many applications integral and proportional control is used together without derivative control. This controller is known as a PI Controller. It uses the proportional gain with the integral control to minimise any offset errors.

2.2.3. Derivative Control

Derivative control in a PID controller can improve the control loop performance by providing higher response. It produces an output based on the errors rate in change. If the error signals rate of change is high then the resulting derivative control is increased. Derivative control is set by the derivative time setting [8]. By setting this high there will be more derivative control provided by the controller.

As with integral timing if the derivative time is set too high it can make the output signal unstable due to oscillation. Additionally noise in derivative control provides a significant challenge as the variability in the process feedback signal can result in agitated responses on the output [12]. Furthermore using derivative control in many applications can increase the complexity as the challenge of tuning can be considerably increased compared with PI control.

2.2.4. Tuning

There are many types of techniques and methods for tuning PID controllers. A selected few are discussed below.

2.2.4.1. Trial and Error

This method is a simple method where all values of Kp, Ki and Kd are set to zero. Initially Kp is adjusted until the system shows oscillating behaviour [11]. Once this is apparent Ki can be adjusted to eliminate this oscillation and reduce any offset. Kd is then set to equip the system with the required overshoot.

2.2.4.2. Process Reaction Curve Technique

This technique firstly provides a step input to the system. The system is performed in an open loop situation so the process variables response can be isolated. Once the step input or disturbance is provided to the system the process variable is recorded. After this adjustments and calculations are made to provide for slope, dead time, rise time of the curve set [13]. From this substitution can be made to acquire Kp, Ki and Kd.

2.2.4.3. Zeigler-Nichols Method

Zeigler and Nichols developed two methods of PID control tuning these being Zeigler-Nichols closed loop method and open loop method. These two methods effectively use the same procedures. It uses the principle of using their made definition which works of the principle of the ratio of amplitudes which occur in the same direction from a step change, is approximated to $\frac{1}{4}$ [14].

$$\frac{A2}{A1} = \frac{1}{4}$$

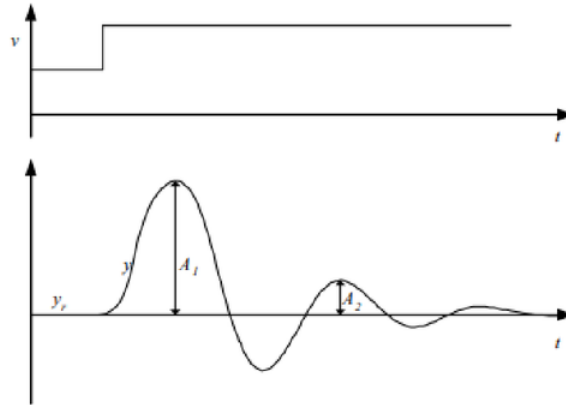


Figure 4: Zeigler-Nichols Method Amplitude Principle [14]

The procedure for a closed loop system is firstly to set K_p to a set value whilst setting K_i and K_d to zero. K_p is then increased until oscillation occurs at a constant amplitude [11] [14]. This is also known as the ultimate gain and the period of oscillation is known as the ultimate period. By then noting the value of K_p , K_d and K_i the Zeigler-Nicholas table can be used to provide values [11] (table 1).

Table 1: Zeigler-Nichols Method Table of Values

	K_c	T_i	T_D
P	$K_u/2$		
PI	$K_u/2.2$	$P_u/1.2$	
PID	$K_u/1.7$	$P_u/2$	$P_u/8$

2.2.4.4. Graph Matching

This method uses a pre made output from a previous plant. This is used as a template where graph alignment of graphs using software is made. K_p , K_i and K_d values can be adjusted until a 'good enough' match is made by designers choice. All step response and overshoot can be refined for the desired application.

2.3. Pulse Width Modulation (PWM)

Pulse Width Modulation (PWM) is a type of digital signal which is used in a variety of control applications. The main role of PWM is to impose a mean voltage level over a specified switching frequency [15]. By changing the on/off time of the signal the mean voltage can be increased or decreased. Varying the on/off time is known as the duty cycle of the waveform this is usually expressed as a percentage. So at a 50% duty cycle the PWM signal is effectively on for half the time and off for half the time (figure 5). There are many advantages of using PWM over traditional analogue signalling these are discussed in more detail in the following sections.

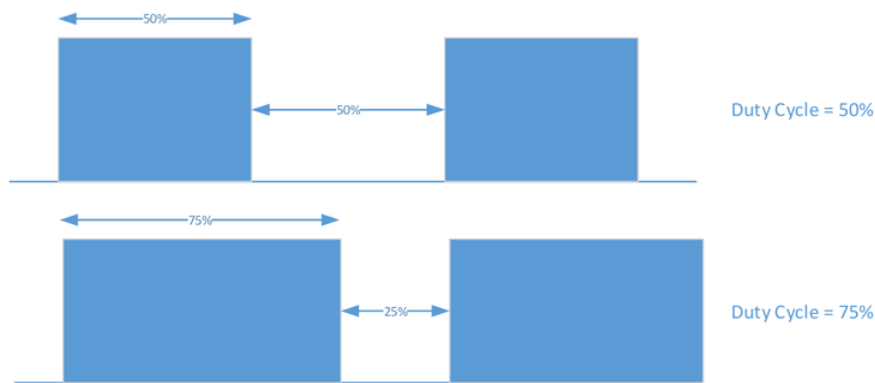


Figure 5: PWM Duty Cycle

2.3.1. PWM Advantages

PWM has many advantages over traditional analogue signalling in control applications. In motor controlling applications PWM can provide higher motor and power efficiency [16]. This increase in efficiency is due to the device being controlled by effectively being off for the specified duty cycle therefore no current draw. Another advantage of PWM is fast control response time [17]. This means that if a shut off is required the pulse can be set to zero within a small amount of cycles, thus allowing for accurate control over the controlled device.

Signal driven motors generally require slow start up control. Using PWM can achieve this with greater precision over analogue control. The duty cycle can be set to a very high resolution (at just 1%) and can be incremented gradually due to its fast response times. This allows for optimal start up responses to be achieved [18].

When using analogue signalling to control motor drivers if a reduction in speed is required the motor is taken out of its operating optimum operating point [19]. Using PWM allows the motor to be controlled only within optimal operating points due to it being either off or on. This can therefore reduce energy wastage and can lead to an increase in a devices life.

Below is a summary of the advantages associated with PWM.

- Dynamic fast response time.
- High precision and control.
- Low maintenance.
- Increased efficiency.
- Reduced energy wastage.

2.3.2. PWM Disadvantages

Although PWM has many advantages over analogue signal control it does have disadvantages. Many of these disadvantages are apparent over PWM signals running at higher frequencies. One of the main disadvantages in motor control is at higher frequencies motors can heat up which therefore causes insulation breakdown [16] [20]. Another disadvantage of PWM control is that at higher frequencies there can be resultant switching losses [19]. This means that although PWM has increased efficiency it can become inefficient at higher frequencies. Using PWM at higher frequencies can also make the signalling septal to high frequency noise which can lead to distortions in the signal.

Below is a summary of the disadvantages of PWM.

- Can lead to motor damage.
- Can become inefficient at higher frequencies due to switching losses.
- Can pick up high frequency noise.

3. Design Specification

Outlined below is the set design specification for the system.

The system **must**:

1. Provide visual indication when the system is operating.
2. Have an emergency stop button.
3. Stop when an alarm has been breached.
4. Provide visual indication of an error.
5. Provide visual indication of a warning.
6. Provide an alarm and warning screen which should present all alarms and warnings with corresponding indicators.
7. Provide an engineering screen to enter related parameters to the system.
8. Provide an operator screen so the system can be activated and any relevant parameters can be displayed.
9. Be driven from a PWM signal with a maximum output of 85%.
10. Be controlled from a PID controller with a maximum overshoot of 5% and a maximum response time of 20 seconds.

Presented below is the specification for each alarm and warning condition (table 2).

Table 2: Errors and Warnings Specification

Parameter	Alarm/Warning	Condition
PWM Output	Alarm	Greater than 85% of maximum
	Warning	Greater than 80% of maximum
PID Controller Output	Warning	Exceeds 75% of maximum
Position	Warning	Less than 20% of maximum
	Warning	Greater than 80% of maximum
	Alarm	Less than 10% of maximum
	Alarm	Greater than 90% of maximum
Set Point	Warning	Set Point Low
	Warning	Set Point High
	Error	Set Point Low
	Error	Set Point High

4. System Design

In the following section is a block system diagram and software flowcharts.

4.1. System Block Diagram

Presented below is a system block diagram (figure 6).

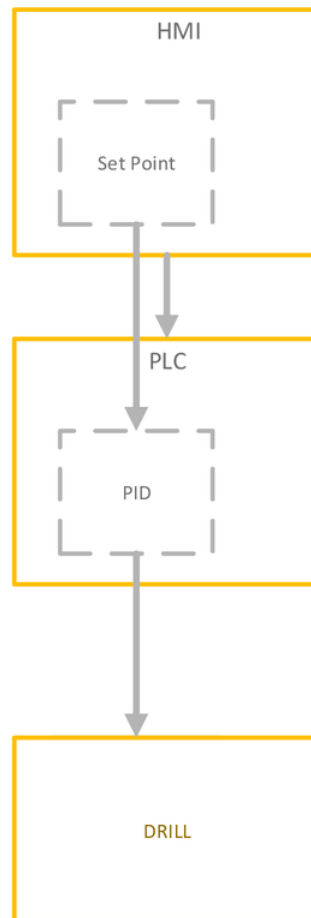


Figure 6: System Block Diagram

4.2. Operation Flowchart

Shown below is a flowchart which shows the overview of the main operation of the system (figure 7).

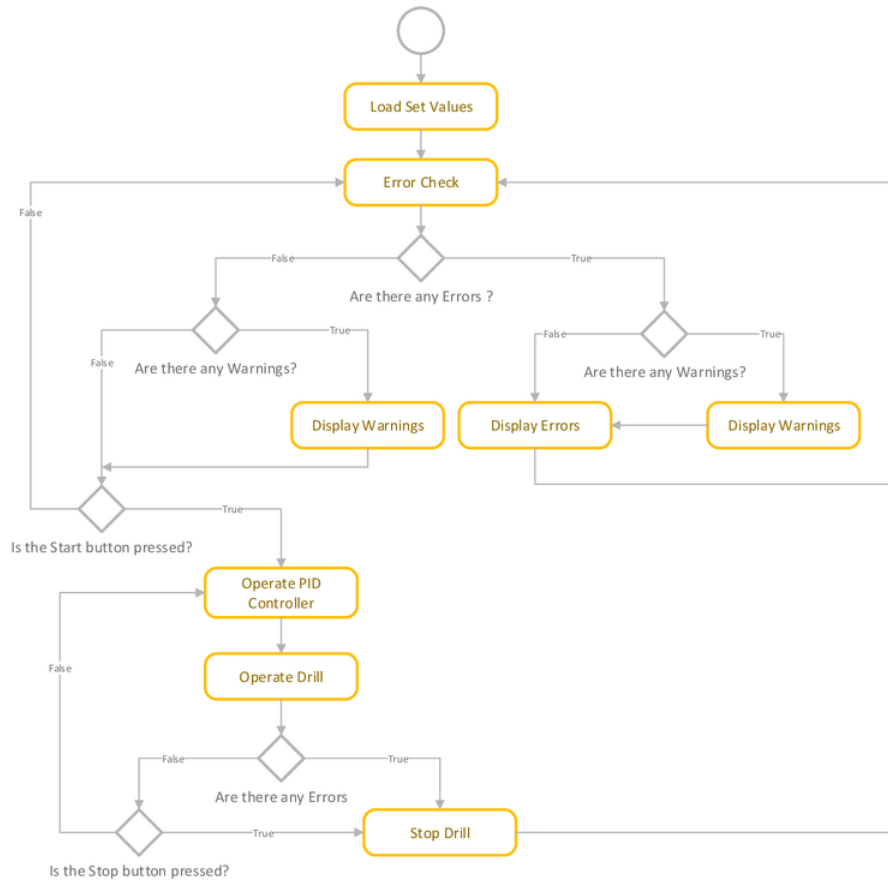


Figure 7: Operation Flowchart Main

Shown below is the error check routine within the main operation flowchart (figure 8).

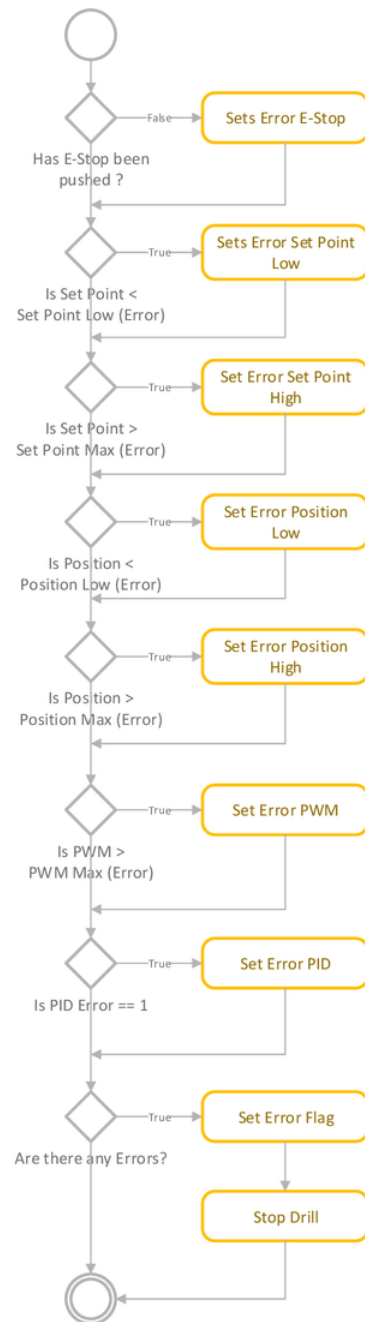


Figure 8: Error Check Operation

Presented below is the warning check sub routine performed within the main operation (figure 9).

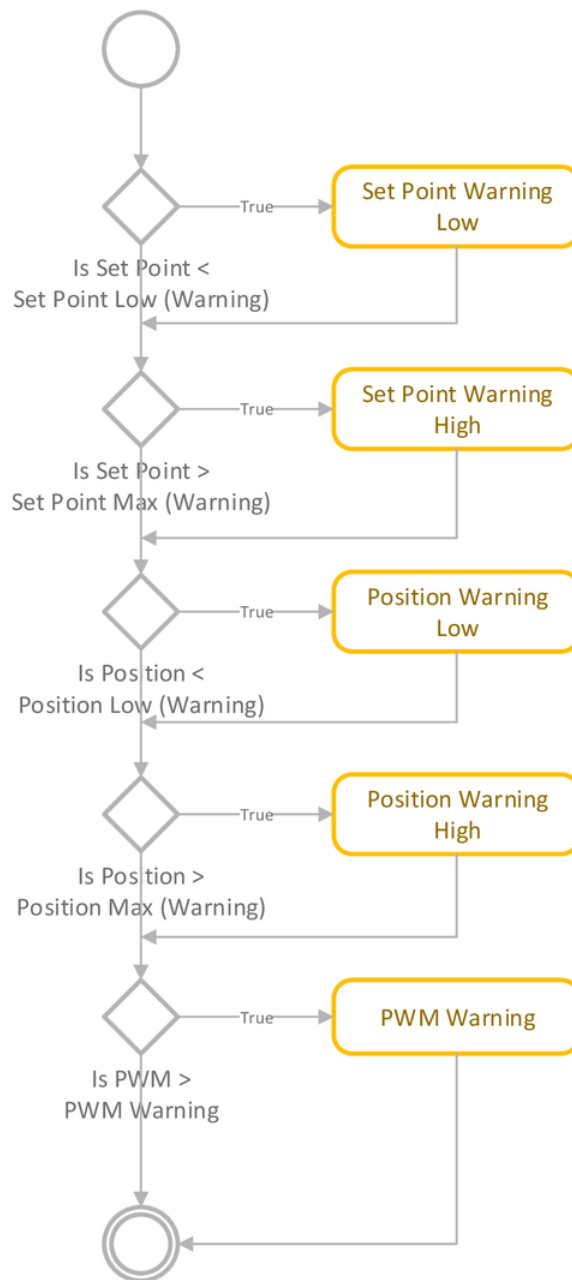


Figure 9: Warning Check Operation

4.3. Operator Flowchart

Shown below is the user screen operator flowchart (figure 10).

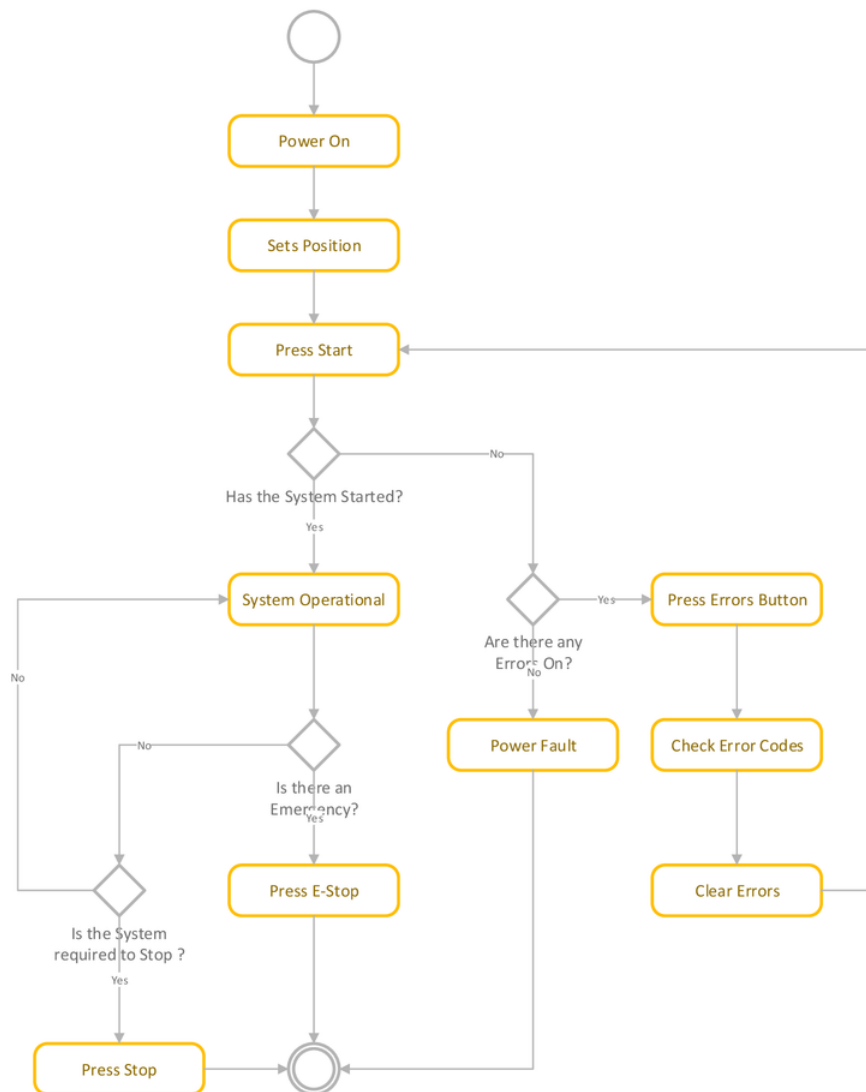


Figure 10: Operator Flowchart

5. System Implementation

Below shows the process of the main system design. The first stages taken was to layout the tags then design the screens and ladder logic. After this was completed the PID controller values were obtained from Matlab by graph matching with a desired output.

5.1. Tags

The screenshot below presents all tags defined in the design (figure 11).

	Name	Tag table	Data type	Address	Retain	Visibl...	Acces...
1	Start	Default tag table	Bool	%I0.6	☐	☒	☒
2	Stop	Default tag table	Bool	%I0.5	☐	☒	☒
3	Operate	Default tag table	Bool	%Q0.3	☐	☒	☒
4	Emergency_Stop	Default tag table	Bool	%M120.0	☐	☒	☒
5	Error_Setpoint_Low	Default tag table	Bool	%M120.1	☐	☒	☒
6	Error_Setpoint_High	Default tag table	Bool	%M120.2	☐	☒	☒
7	Error_Position_Low	Default tag table	Bool	%M120.3	☐	☒	☒
8	Error_Position_High	Default tag table	Bool	%M120.4	☐	☒	☒
9	Error_PWM	Default tag table	Bool	%M120.5	☐	☒	☒
10	Warning_1	Default tag table	Bool	%M121.0	☐	☒	☒
11	Warning_Setpoint_Low	Default tag table	Bool	%M121.1	☐	☒	☒
12	Warning_Setpoint_High	Default tag table	Bool	%M121.2	☐	☒	☒
13	Warning_Position_Low	Default tag table	Bool	%M121.3	☐	☒	☒
14	Warning_Position_High	Default tag table	Bool	%M121.4	☐	☒	☒
15	Warning_PWM	Default tag table	Bool	%M121.5	☐	☒	☒
16	Warning_7	Default tag table	Bool	%M121.6	☐	☒	☒
17	Warning_8	Default tag table	Bool	%M121.7	☐	☒	☒
18	Error_Light	Default tag table	Bool	%M122.2	☐	☒	☒
19	Analogue_Input	Default tag table	Int	%IW64	☐	☒	☒
20	Position_Warning_Min	Default tag table	Real	%MD0	☒	☒	☒
21	Position_Warning_Max	Default tag table	Real	%MD4	☒	☒	☒
22	Setpoint_Warning_Min	Default tag table	Real	%MD8	☒	☒	☒
23	Setpoint_Warning_Max	Default tag table	Real	%MD12	☒	☒	☒
24	Speed_Warning	Default tag table	Real	%MD16	☒	☒	☒
25	Position_Alarm_Min	Default tag table	Real	%MD20	☒	☒	☒
26	Position_Alarm_Max	Default tag table	Real	%MD24	☒	☒	☒
27	Setpoint_Alarm_Min	Default tag table	Real	%MD28	☒	☒	☒
28	Setpoint_Alarm_Max	Default tag table	Real	%MD32	☒	☒	☒
29	Speed_Alarm	Default tag table	Real	%MD36	☒	☒	☒
30	Position_Millimeter	Default tag table	Real	%MD140	☐	☒	☒
31	Output_PER	Default tag table	Int	%MW150	☐	☒	☒
32	Error_PID_Block	Default tag table	Bool	%M120.6	☐	☒	☒
33	Error	Default tag table	Byte	%MB120	☐	☒	☒
34	Error_Set	Default tag table	Bool	%M120.7	☐	☒	☒
35	Red_Light	Default tag table	Bool	%Q0.5	☐	☒	☒
36	Warning	Default tag table	Byte	%MB121	☐	☒	☒
37	Warning_Set	Default tag table	Bool	%M122.0	☐	☒	☒
38	Warning_Light	Default tag table	Bool	%M122.1	☐	☒	☒
39	Reset_Button	Default tag table	Bool	%M140.0	☐	☒	☒
40	PWM_Output	Default tag table	Bool	%M141.0	☐	☒	☒

Figure 11: PLC Tags

There are a total of 40 tags defined in the program. One of the main design considerations taken was retaining tags. Retaining a tag allows the value to be retained in memory so it can be reused. It was decided that all warning and error input thresholds were set as retain. This meant that if any errors were still present, in the next system use they would display. It was decided 100 memory locations were allocated for retention. Although only up to 36 were used it allows allocation for future tags to be added and retained. All retained tags are set as real data types which requires 4 bytes.

Another design consideration for the tags was to use a byte for the errors and warnings. The error byte included all eight errors. The first bit was allocated to the emergency stop with the final bit allocated to the error flag. By using this format it allowed the program to compare the contents of the byte. If one of the errors were set within the byte it would mean the value of the byte would be greater than zero. So if the error byte was greater than zero it would set the error flag defined as 'Error_Set'. Error flag can then be used to turn off the system (figure 12).

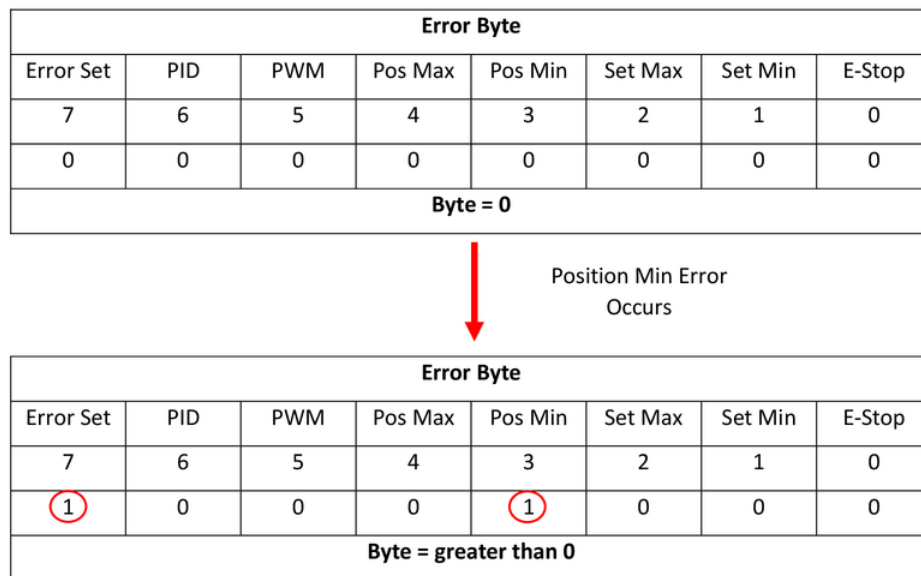


Figure 12: Error Byte Operation

The warning byte works off this same principle. However as the system does not require to stop when a warning is in place, the warning flag will just activate an indicator on the display. It does not actively shut down system operation.

Position is simulated from an analogue input. In the program this will need to be converted to 'mm'. A conversion tag name 'Position_Millimeter' is used to store the converted value.

5.2. Screen Design

For the system to meet the design specification there were three screens produced. The screens are a main screen (user screen), engineer screen and a alarms and warnings screen.

5.2.1. User Screen

Presented below is the user screen layout (figure 13).

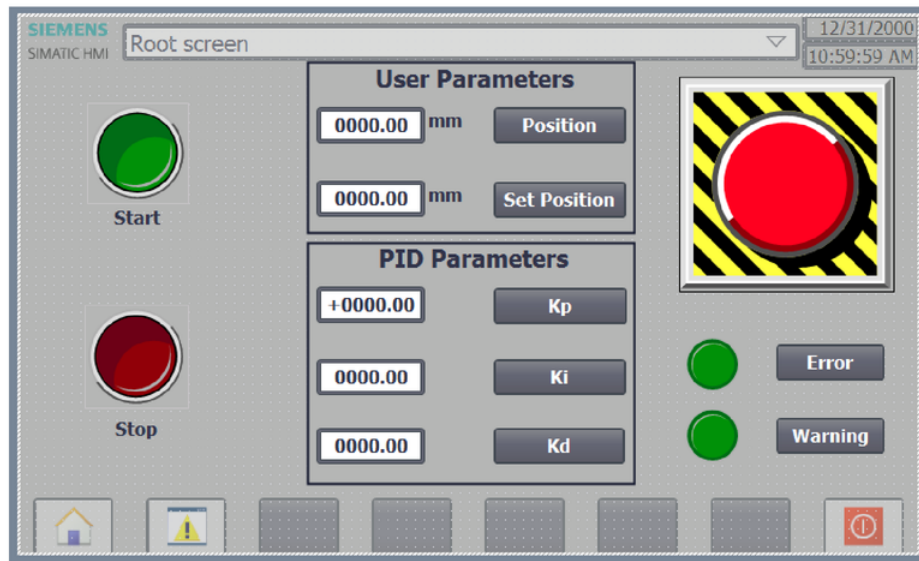


Figure 13: User Screen

The screen was laid out so the emergency stop button was the largest switch so it stood out and was easy to access for the user. The user was required to enter a desired position for drilling. This was set up as an input button which allowed the user to set this value. It was also decided to show the current drill position and the PID controller values. Both operate and stop buttons were reduced in size compared to the emergency stop and were situated in the opposite area of the screen. The error and warnings are set up with two lights and buttons. These take the user to the warnings and errors screen. F2 of the screen is set out to access the engineering screen upon activation.

5.2.1.1. Associated Tags

Presented below is a description of all associated tags used and function of each component.

Start Button (figure 14)	
Tag Name	'Operate'
Tag Address	%I0.6
Data Type	Bool (initial value == 1)
Retain	No
Function	- Start button to operate the drill. - Lights when drill is operating.

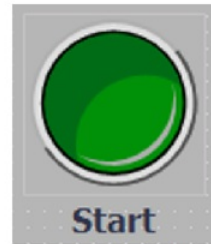


Figure 14: Start Button

Stop Button (figure 15)	
Tag Name	'Operate'
Tag Address	%I0.5
Data Type	Bool (initial value == 0)
Retain	No
Function	- Stops drill operating upon pressing. - Lights when drill is not operational.



Figure 15: Stop Button

Emergency Stop (figure 16)	
Tag Name	'Emergency_Stop'
Tag Address	%M120.0
Data Type	Bool
Retain	No
Function	- Provides an emergency stop of the system. - Bit zero of error byte.



Figure 16: E-Stop

Error Indicator & Error Screen Button (figure 17)	
Tag Name	'Error_Light'
Tag Address	%M122.2
Data Type	Bool
Retain	No
Function	- Error light switches on if error flag is set - Error button takes user to error screen



Figure 17: Error Indicator

Warning Indicator & Warning Screen Button (figure 18)	
Tag Name	'Warning_Light'
Tag Address	%M122.1
Data Type	Bool
Retain	No
Function	- Warning light switches on if warning flag is set. - Button takes user to warning screen.

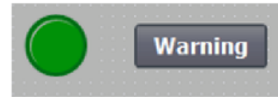


Figure 18: Warning Indicator

Current Position Output (figure 19)	
Tag Name	'Position_Millimeter'
Tag Address	%MD140
Data Type	Real
Retain	No
Function	- Current position output in mm. - 4 bytes required due to real type.



Figure 19: Current Position Output

Set Position Input (figure 20)	
Tag Name	'PID_Compact_1_Setpoint'
Tag Address	%DB1
Data Type	Real
Retain	No
Function	- Input set position of the drill. - Tag from PID Controller Block

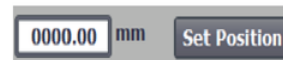


Figure 20: Set Position Output

Kp Input (figure 21)	
Tag Name	'PID_Compact_1_CtrlParamsBackUp_Gain'
Tag Address	%DB1
Data Type	Real
Retain	No
Function	- Input for proportional (kp) for PID. - Built in tag to PID controller.



Figure 21: Kp Input

Ki Input (figure 22)	
Tag Name	'PID_Compact_1_CtrlParamsBackUp_Ti'
Tag Address	%DB1
Data Type	Real
Retain	No
Function	- Input for integral (ki) for PID. - Built in tag to PID controller.



Figure 22: Ki Input

Kd Input (figure 23)	
Tag Name	'PID_Compact_1_CtrlParamsBackUp_Td'
Tag Address	%DB1
Data Type	Real
Retain	No
Function	- Input for derivative (kd) for PID. - Built in tag to PID controller.



Figure 23: Kd Input

5.2.2. Engineer Screen

Presented below is the engineer screen (figure 24).

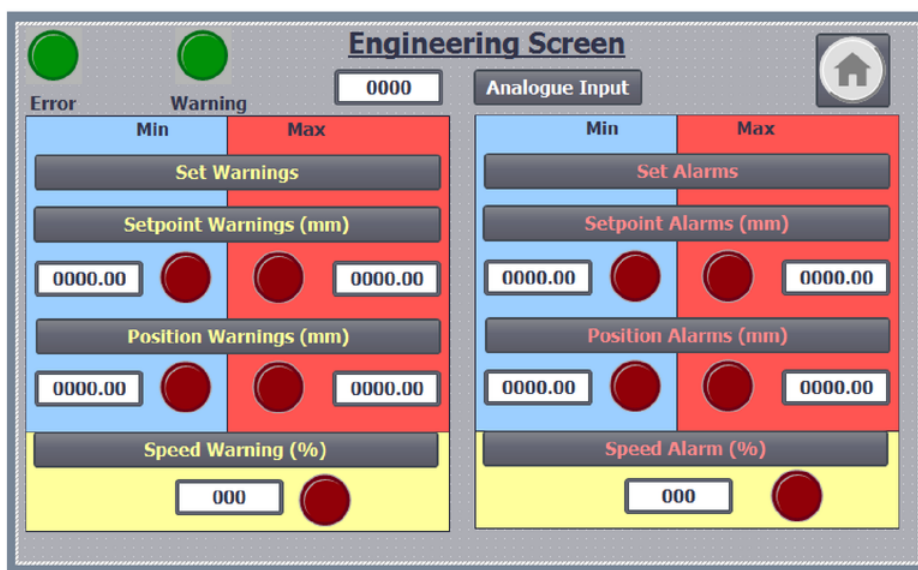


Figure 24: Engineering Screen

The engineering screen provides all inputs for each warning and error threshold. The screen is set out with the warnings on one side and the errors on the other. This was done to present warnings and errors individually. Analogue input is set up to test the value of the potentiometer connected to the PLC. Error and warning lights feature at the side of each represented threshold to display what value is producing an error. Two error and warning lights feature at the top to indicate an overall error or warning. The home button is located at the top right corner and is used to switch to the user screen.

5.2.2.1. Associated Tags

Presented below is a description of all associated tags used and function of each button.

Error Indicator (figure 25)	
Tag Name	'Error_Light'
Tag Address	%M122.2
Data Type	Bool
Retain	No
Function	- Error indicator when error flag is set.

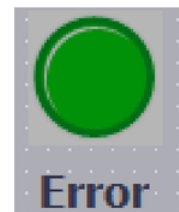


Figure 25: Error Indicator

Warning Indicator (figure 26)	
Tag Name	'Warning_Light'
Tag Address	%M122.1
Data Type	Bool
Retain	No
Function	- Warning indicator when warning flag is set.



Figure 26: Warning Indicator

Analogue Input Display (figure 27)	
Tag Name	'Analogue_Input'
Tag Address	%IW64
Data Type	Int
Retain	No
Function	- Displaying the analogue input which is simulating the position of the drill.



Figure 27: Analogue Input Display

Home Button (figure 28)	
Tag Name	N/A
Tag Address	N/A
Data Type	N/A
Retain	N/A
Function	- Once pressed it returns to the user screen.



Figure 28: Home Button

The table below presents all warning and error threshold inputs (table 3). See figure for image on display (figure 29).

Table 3: Warning and Error Threshold Input Tags

	Warning /Error	Tag Name	Tag Address	Data Type	Retain
Set Point Min Input	Warning	'Setpoint_Warning_Min'	%MD8	Real	Yes
	Error	'Setpoint_Alarm_Min'	%MD28	Real	Yes
Set Point Min Indicator	Warning	'Warning_Setpoint_Low'	%M121.1	Bool	No
	Error	'Error_Setpoint_Low'	%M120.1	Bool	No
Set Point Max Input	Warning	'Setpoint_Warning_Max'	%MD12	Real	Yes
	Error	'Setpoint_Alarm_Max'	%MD32	Real	Yes
Set Point Max Indicator	Warning	'Warning_Setpoint_High'	%M121.2	Bool	No
	Error	'Error_Setpoint_High'	%M120.2	Bool	No
Position Min Input	Warning	'Position_Warning_Min'	%MD0	Real	Yes
	Error	Position_Alarm_Min	%MD20	Real	Yes
Position Min Indicator	Warning	'Warning_Position_Low'	%M121.3	Bool	No
	Error	'Error_Position_Low'	%M120.3	Bool	No
Position Max Input	Warning	'Position_Warning_Max'	%MD4	Real	Yes
	Error	'Position_Alarm_Max'	%MD24	Real	Yes
Position Max Indicator	Warning	'Warning_Position_High'	%M121.4	Bool	No
	Error	'Error_Position_High'	%M120.4	Bool	No
PWM Input	Warning	'Speed_Warning'	%MD16	Real	Yes
	Error	'Speed_Alarm'	%MD36	Real	Yes
PWM Indicator	Warning	'Warning_PWM'	%121.5	Bool	No
	Error	'Error_PWM'	%120.5	Bool	No



Figure 29: Engineering Warning and Error Input Layout

All these work using the same function. The inputs provide the threshold for the alarm/warnings. The indicators which are situated next to each input provide corresponding warning/error indication. The inputs are all retained memory locations and are real 4 Byte values.

5.2.3. Errors and Warnings Screen

Presented below is the errors and warnings screen (figure 30).

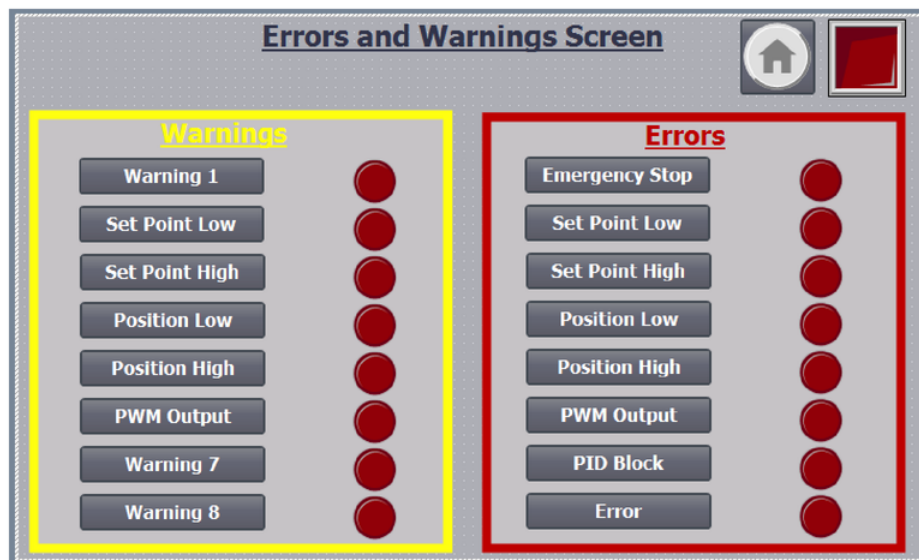


Figure 30: Errors and Warnings Screen

The errors and warnings screen is set out in two columns with one half of the screen displaying the warnings and the other half displaying the alarms. Each indicator light is corresponding to the label next to each. The light at the top of the page is a button/indicator which will light if error flag is set. To clear error flag the user must view the error screen to check for errors.

Once the button is pressed providing no other errors are ongoing the screen switches to home and turns off the error flag bit.

5.2.3.1. Associated Tags

Below are the tags which are associated with each indicator/input.

Error Indicator (figure 31)	
Tag Name	'Error_Light'
Tag Address	%M122.2
Data Type	Bool
Retain	No
Function	- Error indicator when error flag is set. - When pressed it goes to user screen and resets error flag.



Figure 31: Error Indicator and Error Flag Reset

The table below shows all indicator tags for the associated warnings and errors (table 4 and 5).

Table 4: Warning Tag Layout

Warnings (figure 32)				
Name	Tag Name	Tag Address	Data Type	Retain
Warning 1	'Warning_1'	%121.0	Bool	No
Set Point Low	'Warning_Setpoint_Low'	%121.1	Bool	No
Set Point High	'Warning_Setpoint_High'	%121.2	Bool	No
Position Low	'Warning_Position_Low'	%121.3	Bool	No
Position High	'Warning_Position_High'	%121.4	Bool	No
PWM Output	'Warning_PWM'	%121.5	Bool	No
Warning 7	'Warning_7'	%121.6	Bool	No
Warning 8	'Warning_8'	%121.7	Bool	No

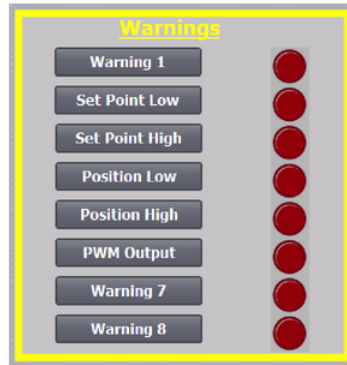


Figure 32: Warnings Layout

Errors (figure 33)				
Name	Tag Name	Tag Address	Data Type	Retain
Emergency Stop	'Emergency_Stop'	%120.0	Bool	No
Set Point Low	'Error_Setpoint_Low'	%120.1	Bool	No
Set Point High	'Error_Setpoint_High'	%120.2	Bool	No
Position Low	'Error_Position_Low'	%120.3	Bool	No
Position High	'Error_Position_High'	%120.4	Bool	No
PWM Output	'Error_PWM'	%120.5	Bool	No
PID Block	'Red_Light'	%120.6	Bool	No
Error	'Error_Set'	%120.7	Bool	No

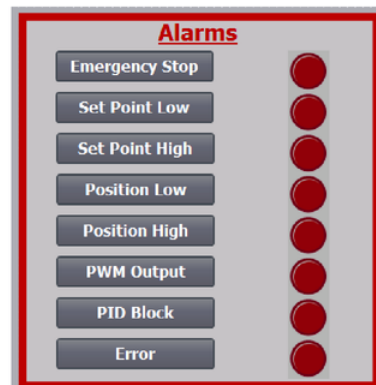


Figure 33: Alarms/Error Layout

5.3. Ladder Logic Design

Presented in this section is an explanation on the operation of all ladder logic networks designed.

5.3.1. Network 1

The systems first network is used to start the drill (figure 34).

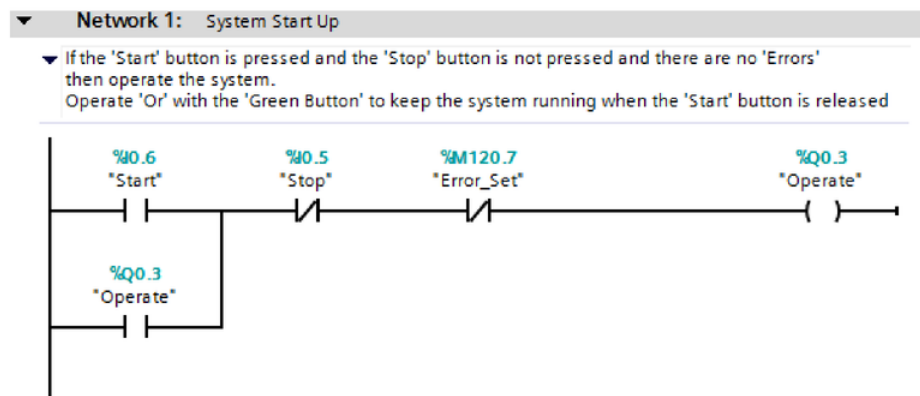


Figure 34: Network 1

The network operates by using a normally open switch which is assigned the 'Start' tag. This is 'AND' with two normally closed switches tagged to 'Stop' which is an input and 'Error_Set' which is a non-retained memory location and the most significant bit in the error byte. The output is 'Operate' which is set as an output address.

The logic works by if start is pressed, stop is not pressed and there are no errors then operate the system. Operate is used as an 'OR' with 'Start' to keep the drill operating when start is released.

5.3.2. Network 2

Network two presents the logic to perform an error check (figure 35).

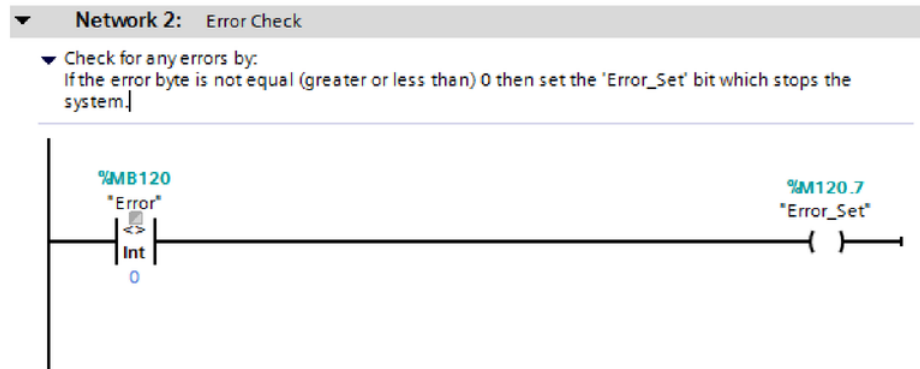


Figure 35: Network 2

The network operates by using a comparison block which compares the value of the error byte to 0. If the value is greater than zero then it outputs and sets the 'Error_Set' bit of the byte.

5.3.3. Network 3

The third network is presented below (figure 36).

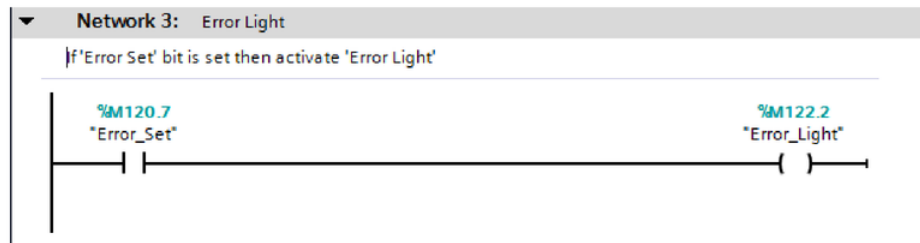


Figure 36: Network 3

The network operates by outputting and turning the error light on if the 'Error_Set' bit has been set.

5.3.4. Network 4

The fourth network created is shown below (figure 37).

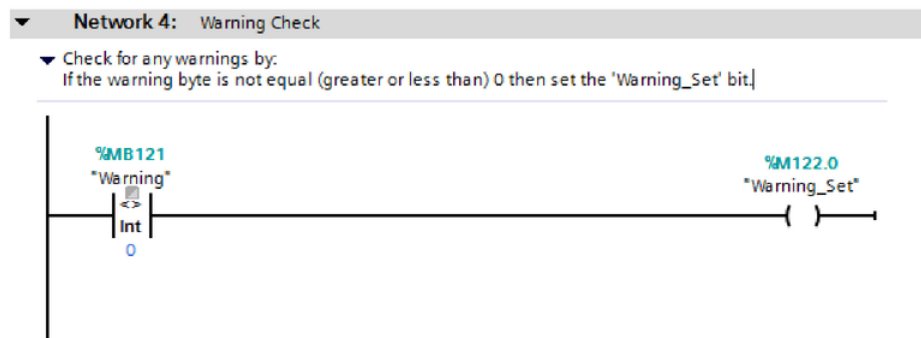


Figure 37: Network 4

The network operates by using a comparison block to compare the value of the warning byte against 0. If the value of the byte is greater than 0 then it sets the 'Warning_Set' bit. Unlike the error bit which is of the same byte the warning bit is kept separate. This is due to if a warning is active the system will still operate.

5.3.5. Network 5

The fifth network is presented below (figure 38).

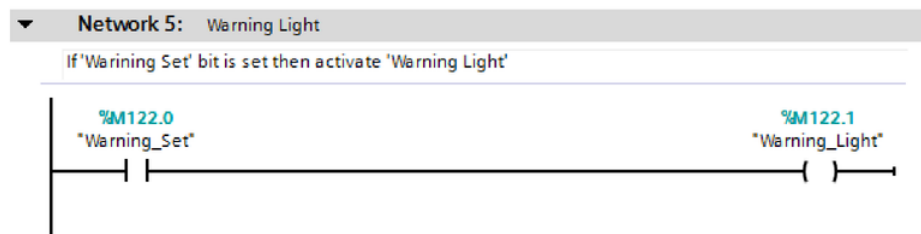


Figure 38: Network 5

This network operates by monitoring the 'Warning_Set' bit. If it is set then it outputs to the warning light which is part of the warning byte.

5.3.6. Network 6

The sixth network is shown below (figure 39).

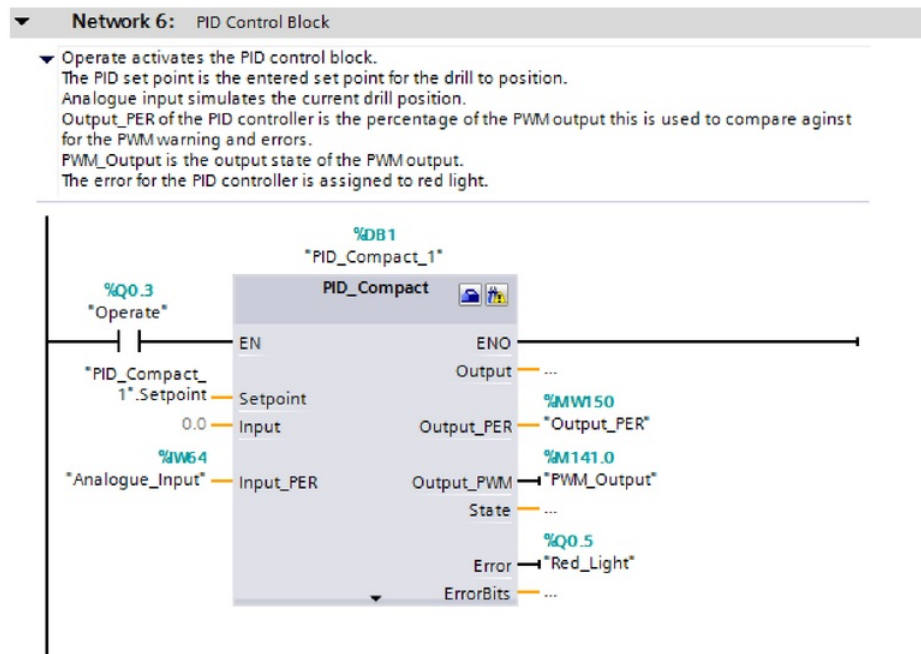


Figure 39: Network 6

Network six is the PID controller block. If the system is operating it enables this block. The set point is an input which is tagged to the built in tags within the block. This tag is also tagged to the user set point input. The input PER is provided by the analogue input to simulate the current position of the drill. The outputs from the system are provided by 'Output_PER' tag which provides the main system output value. The PWM output is an output which is either on or off. The red light is the output from the PID controller error block.

5.3.7. Network 7

Presented below is the seventh network (figure 40).

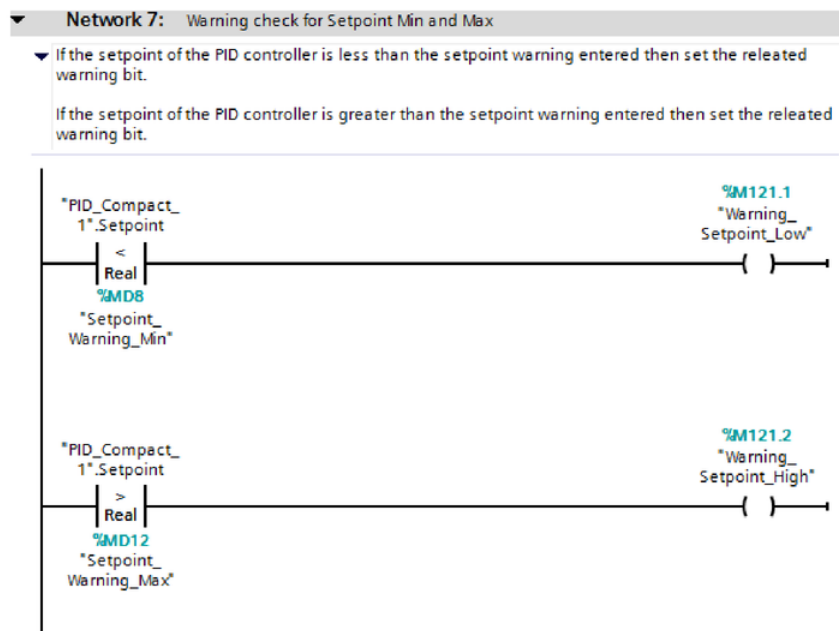


Figure 40: Network 7

The network operates by using a comparison block in which the set point entered is compared with the set point warning threshold. If the set point is too low it outputs and sets the 'Warning_Setpoint_Low, whereas if high it sets the 'Warning_Setpoint_High'.

5.3.8. Network 8

Below presents network eight of the system (figure 41).

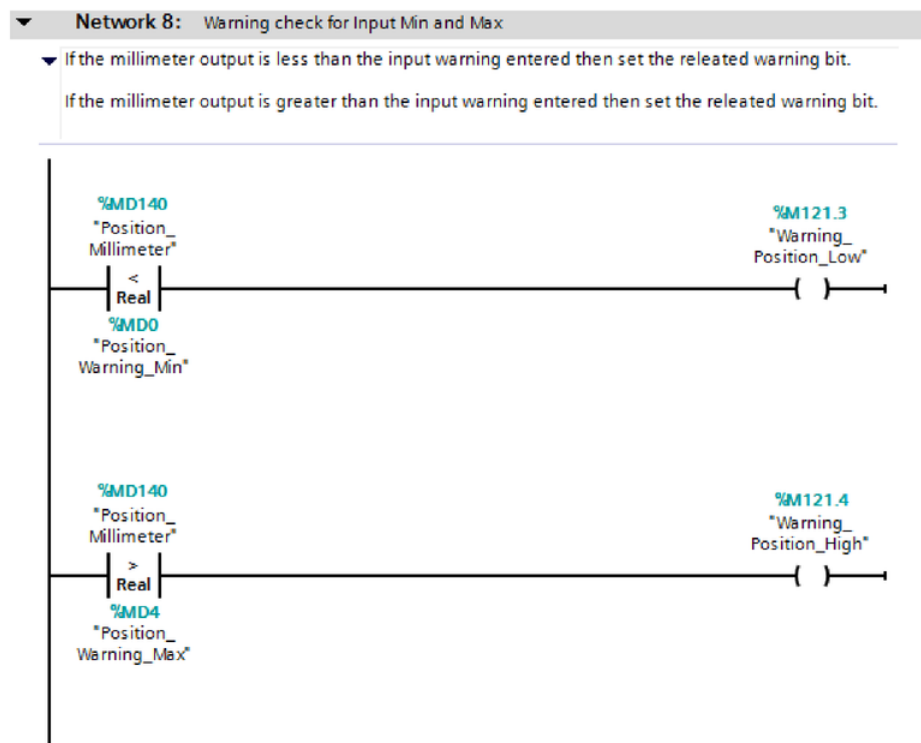


Figure 41: Network 8

The network operates by using comparison blocks to compare the converted position in millimetres to the position warning thresholds set in engineering. If the position is too low then it sets the 'Warning_Position_Low' bit. If it is too high it sets the 'Warning_Position_High' bit.

5.3.9. Network 9

Shown below is network nine of the system (figure 42).

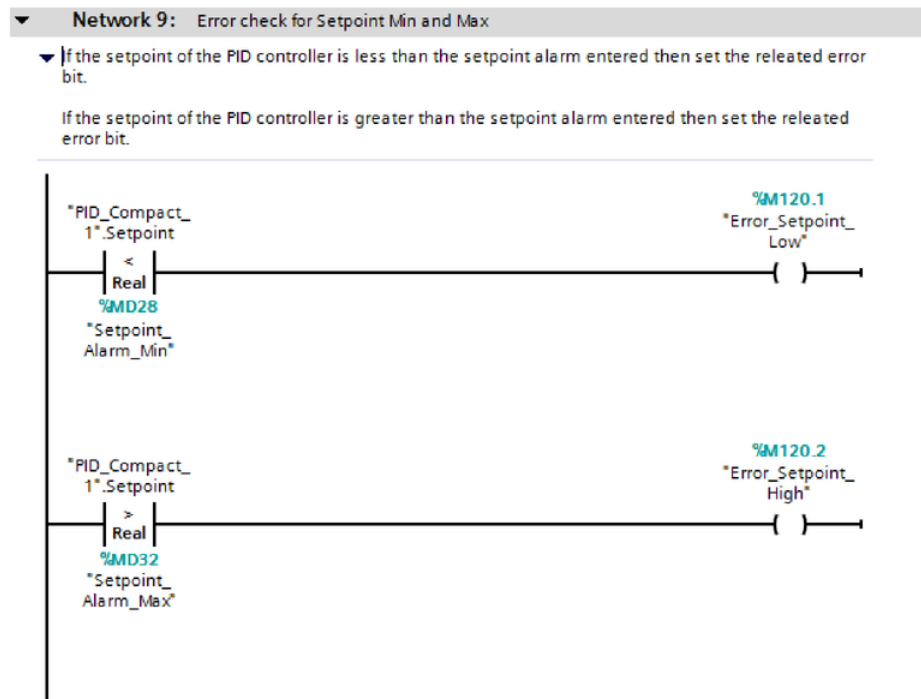


Figure 42: Network 9

The network operates using comparison blocks to compare the set point entered with the alarm set point thresholds. It then outputs a corresponding error bit.

5.3.10. Network 10

Below is the tenth network in the system (figure 43).

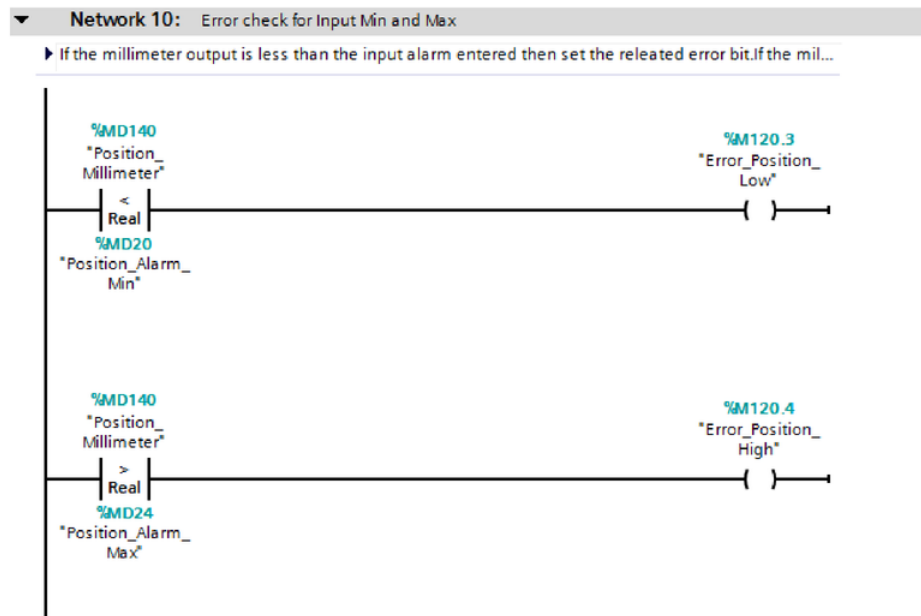


Figure 43: Network 10

The network functions by using comparison blocks. It compares the values of the position in millimetres with the alarm position threshold set. It then outputs a corresponding error bit.

5.3.11. Network 11

Presented below is the eleventh network (figure 44).

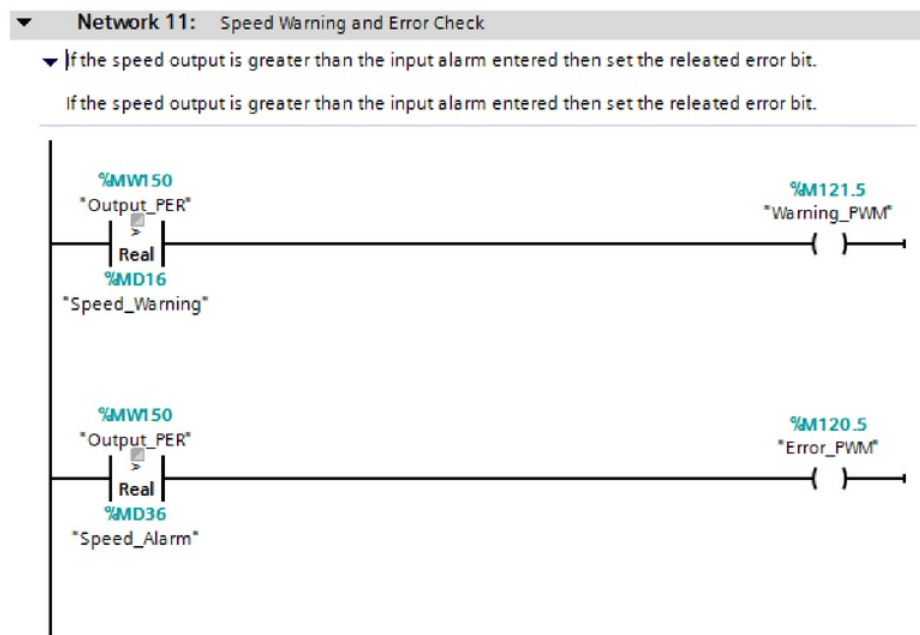


Figure 44: Network 11

The network operates by using comparison blocks to compare the 'Output_PER' of the PID block with the set speed and warning thresholds. It then outputs and sets the corresponding error/warning bits.

5.3.12. Network 12

The final network is shown below (figure 45).

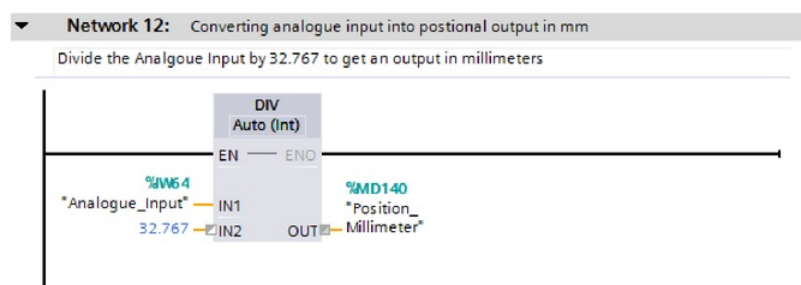


Figure 45: Network 12

The network is used to divide the analogue input by 32.767. This is to convert the analogue value of the input to a millimetre output.

5.4. PID Controller Design

The PID controller was designed using 'Matlab'. The stages taken are outlined in the following section.

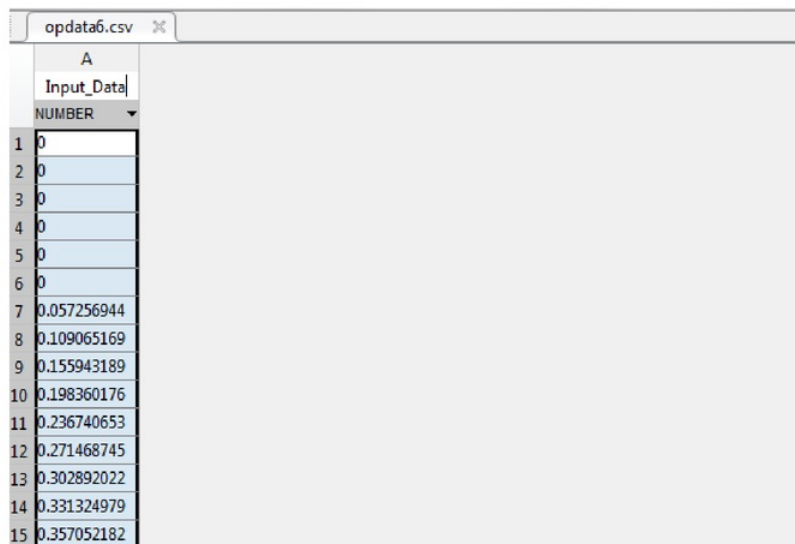
5.4.1. Matlab Set up

The first stage was to set up 'Matlab' which is shown below (figure 46). An output file was supplied with the desired output from an open loop step response. To design the controller graph matching tuning was applied using the supplied output file.

```
Academic License
>> kp = 1
kp =
    1
>> ki = 1
ki =
    1
>> kd = 1
kd =
    1
>> n = 100
n =
   100
```

Figure 46: Initial Value Allocation

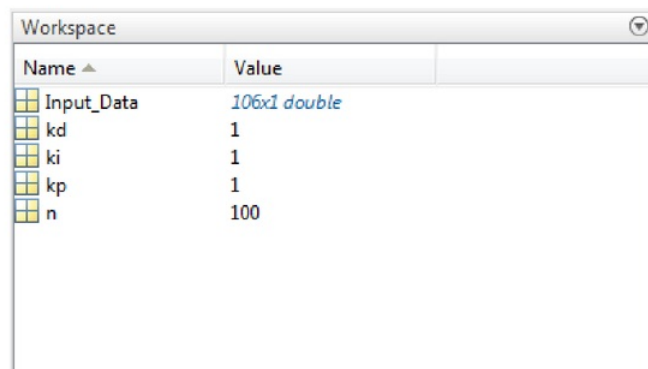
Kp, Ki and Kd are all initially set to one with n set to 100. After this the PID controller tool was called.



opdata6.csv	
A	
Input_Data	
	NUMBER
1	0
2	0
3	0
4	0
5	0
6	0
7	0.057256944
8	0.109065169
9	0.155943189
10	0.198360176
11	0.236740653
12	0.271468745
13	0.302892022
14	0.331324979
15	0.357052182

Figure 47: Output File Import Data

After this the data from the supplied file was imported to the controller tool (figure 47).



Workspace	
Name	Value
Input_Data	106x1 double
kd	1
ki	1
kp	1
n	100

Figure 48: Workspace Values

The workspace information was checked for the correct values (figure 48).

Import Step Response

Output Signal
Specify as a double vector, timeseries or an iddata object containing one output signal.

Input_Data

Name: Output (y)

Input Signal

Amplitude (A): 1

Offset (u_0): 0

Onset Lag (T_Δ): 0

Name: Input (u)

Time Vector

Start Time (T_0): 7

Sample Time (ΔT): 1

Units: seconds

Import

Figure 49: Step Response Import

The next step was to import the step response from the file (figure 49). The timing vector was changed to a seven second start time. This response was then imported to the control tool. After this was completed graph matching could begin.

5.4.2. Single Pole

The first method attempted was to match using a single pole structure.

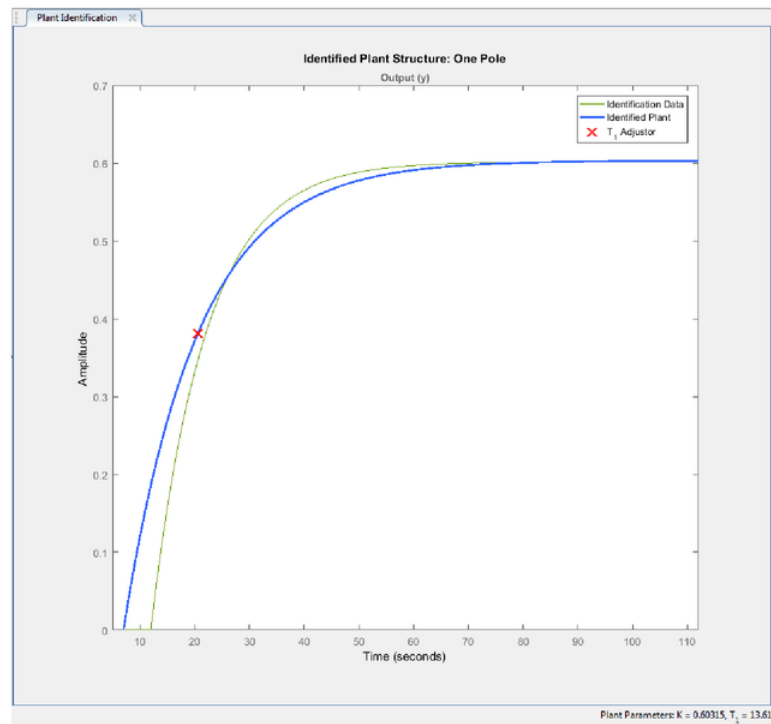


Figure 50: Single Pole Manual

Manual graph matching was first applied and is shown above (figure 50).

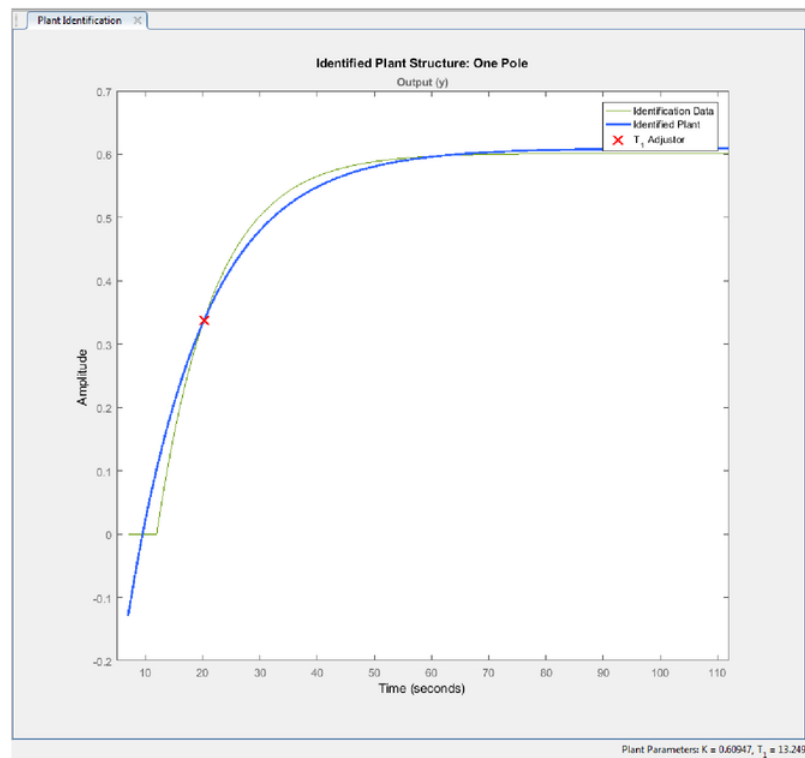


Figure 51: Automatic Single Pole

Using the automatic matching tool the response has been improved (figure 51). The wanted output follows the rise time. However, there is a visible steady state error meaning the one pole structure does not equal the set point. The transfer function for one pole structure is shown below (figure 52).

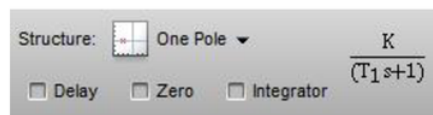


Figure 52: Single Pole Transfer Function

5.4.3. Two Real Poles

The second structure used was two real poles.

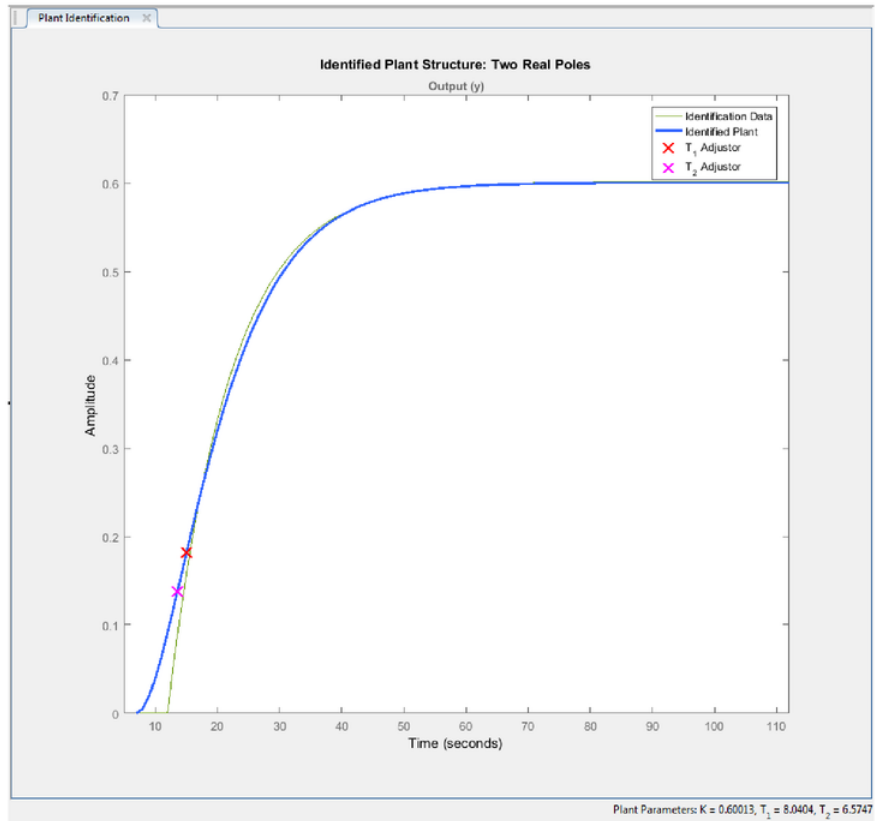


Figure 53: Two Real Poles Manual

The manual matching result is shown above (figure 53).

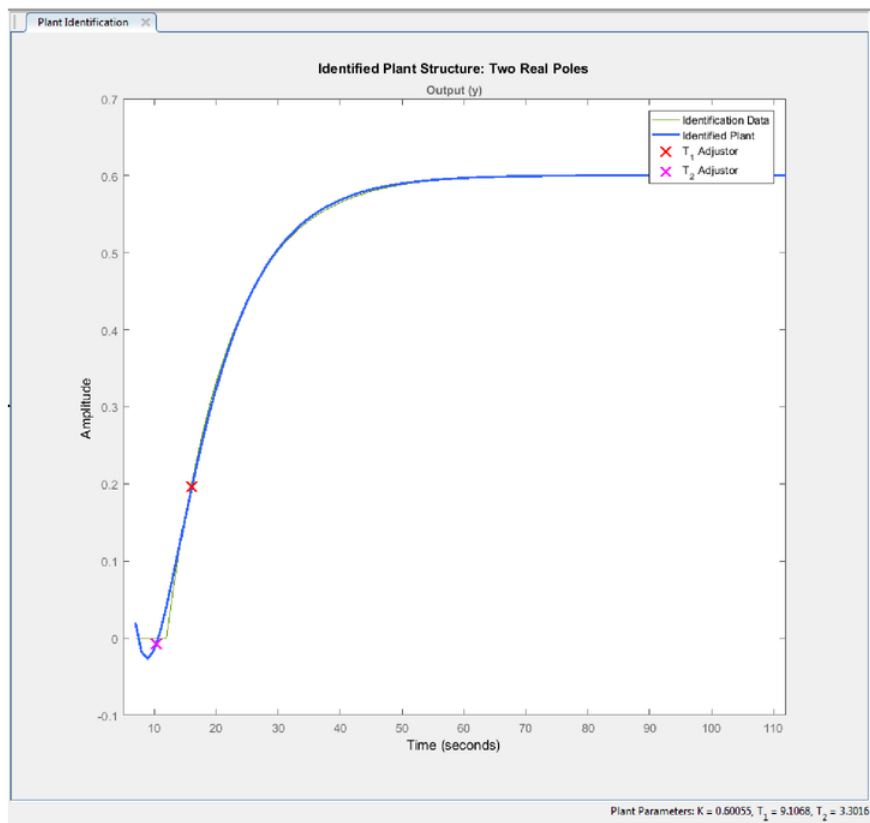


Figure 54: Two Real Poles Automatic

The automatic response presents a slightly improved response than the manual attempt (figure 54). The step response in comparison with the actual outcome (PV) has minimal deviation and therefore gives the most optimal outcome. This structure has an improved response over single pole. The transfer function for this is shown below (figure 55).



Figure 55: Two Real Poles Transfer Function

5.4.4. Underdamped Pair

The third structure attempted was an underdamped pair.

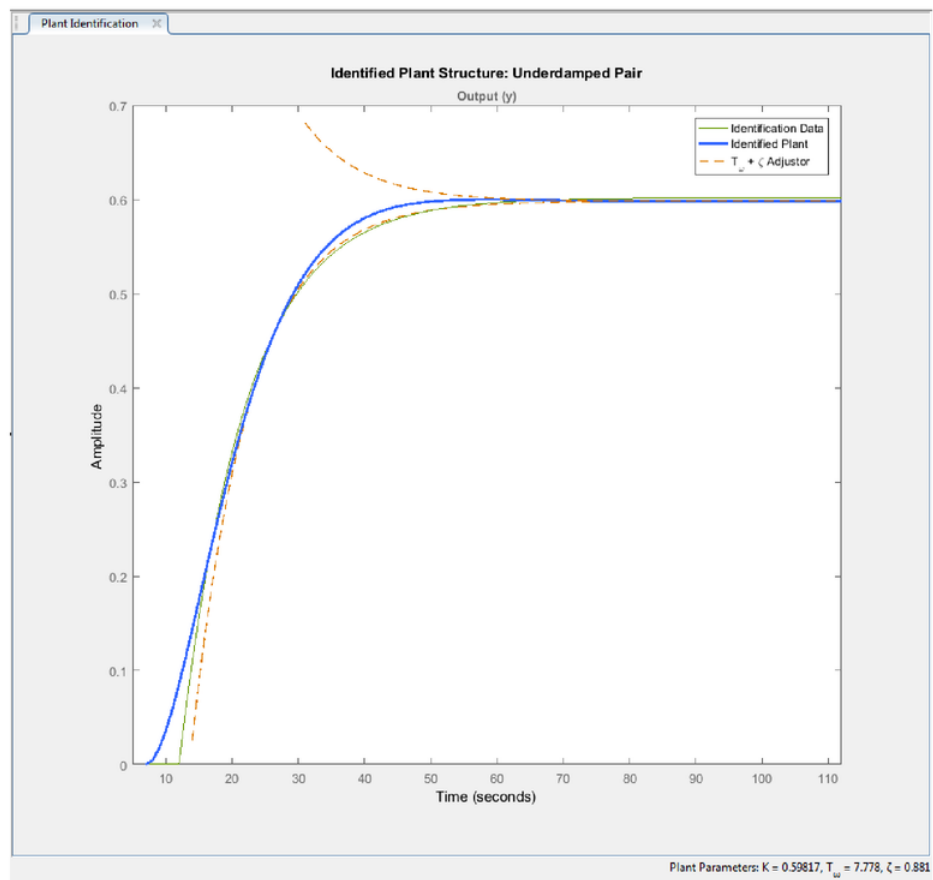


Figure 56: Underdamped Pair Manual

The manual attempt is shown above (figure 56).

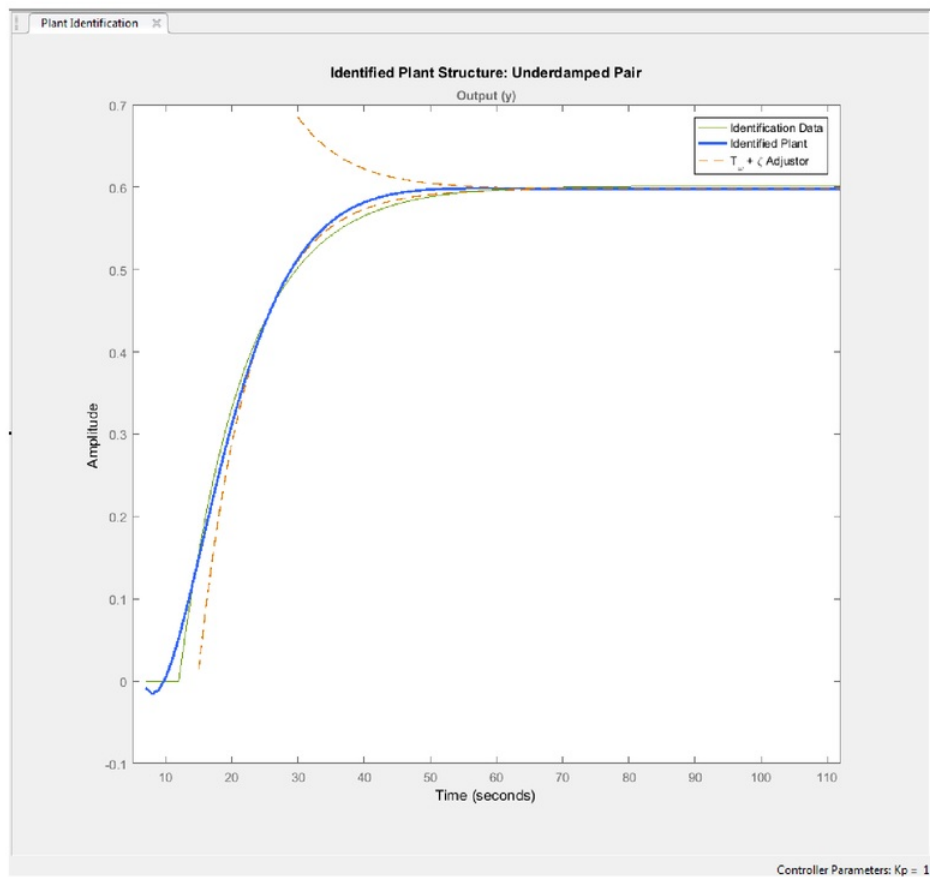


Figure 57: Underdamped Pair Automatic

The automatic match has improved the response slightly (figure 57). The start time of the above system is improved compared with the manual attempt. The response also is stable as it reaches the set point. However the response is less accurate than that of two real poles. The transfer function is shown below (figure 58).

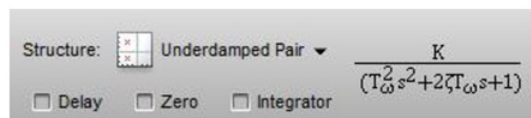


Figure 58: Underdamped Pair Transfer Function

5.4.5. Underdamped Pair and Real Poles

The final structure used was the underdamped pair and real poles.

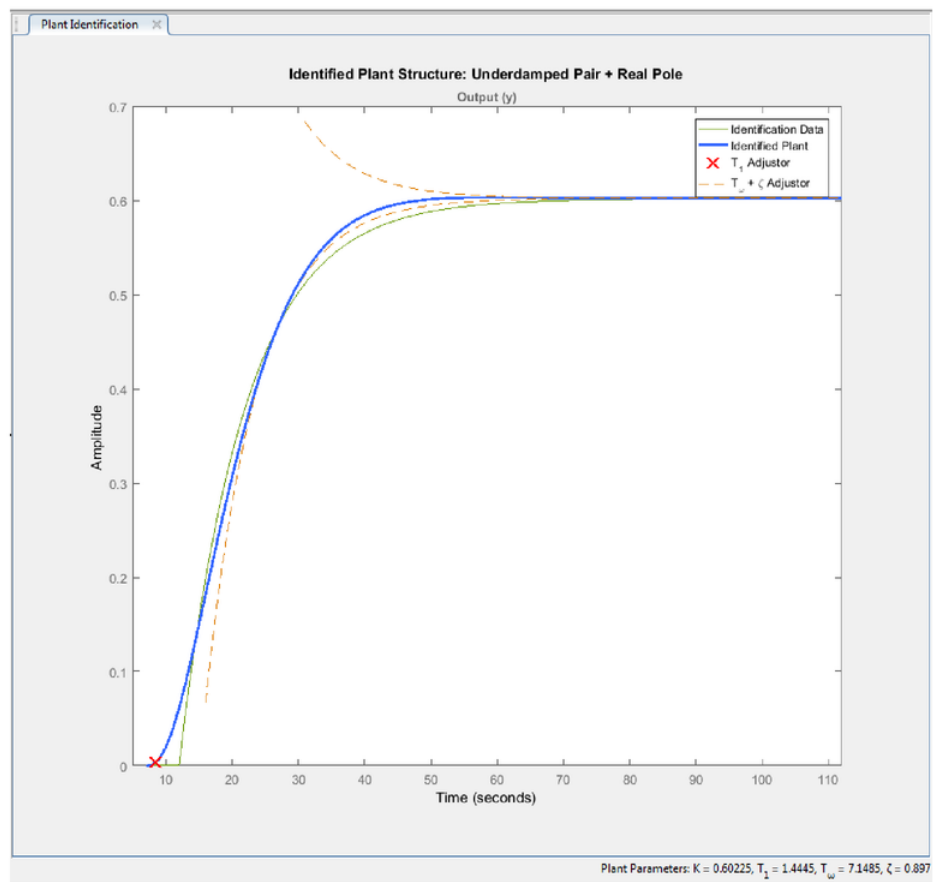


Figure 59: Underdamped Pair and Real Poles Manual

The manual attempt is shown above (figure 59).

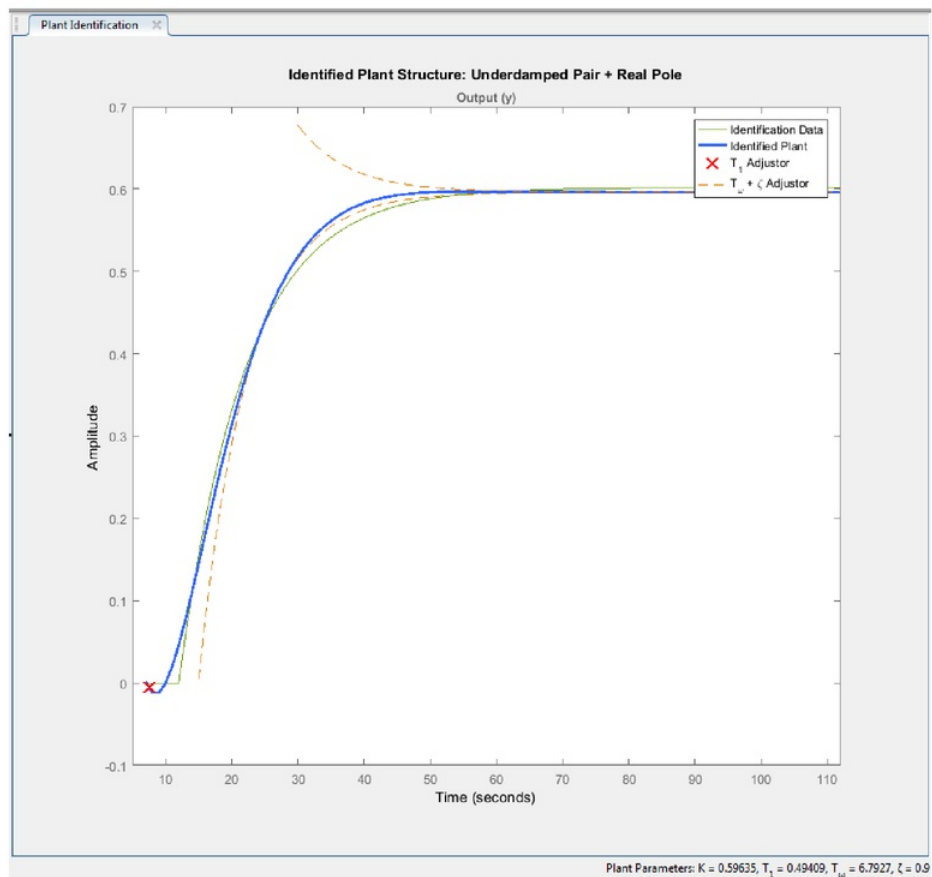


Figure 60: Underdamped Pair and Real Pole Automatic

The automatic match provides an improvement over the manual. The system follows the desired rise time. However the system never reaches the set point and is just under with a slight steady state error present. The transfer function is shown below (figure 61).

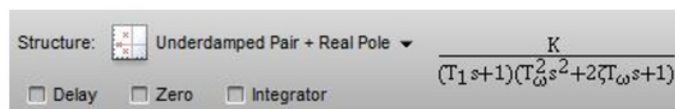
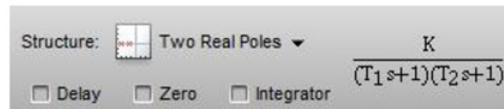


Figure 61: Underdamped Pair and Real Pole Transfer Function

5.4.6. Final Design

The final design which was selected was to use a two real poles structure. The system then had to be tuned using the PID controller tool to achieve the desired overshoot of under 5% and the desired response time of 20 seconds in total. The transfer function used is the two poles function (figure 55).



5.4.6.1. Proportional Only

Firstly a proportional only control was used (figure 62).

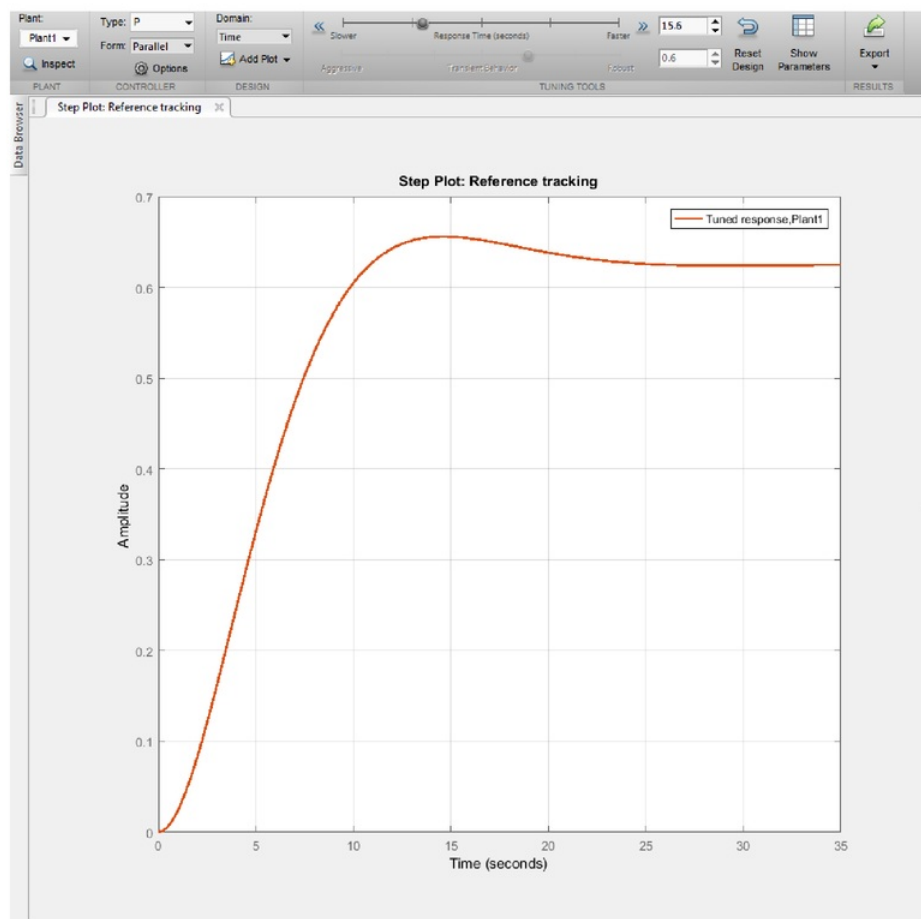


Figure 62: Proportional Only Control

Controller Parameters	
	Tuned
Kp	2.779
Ki	n/a
Kd	n/a
Tf	n/a
Performance and Robustness	
	Tuned
Rise time	7.06 seconds
Settling time	20.1 seconds
Overshoot	4.9 %
Peak	0.656
Gain margin	Inf dB @ Inf rad/s
Phase margin	108 deg @ 0.128 rad/s
Closed-loop stability	Stable

Figure 63: Proportional Control Performance

Using proportional only does not meet the design specification. The overall response time is 27.16 seconds which is out of specification limits. The overshoot is within the design specification of 4.9% (figure 63).

5.4.6.2. Proportional and Integral

Shown below is the proportional and integral control tuning (figure 64).

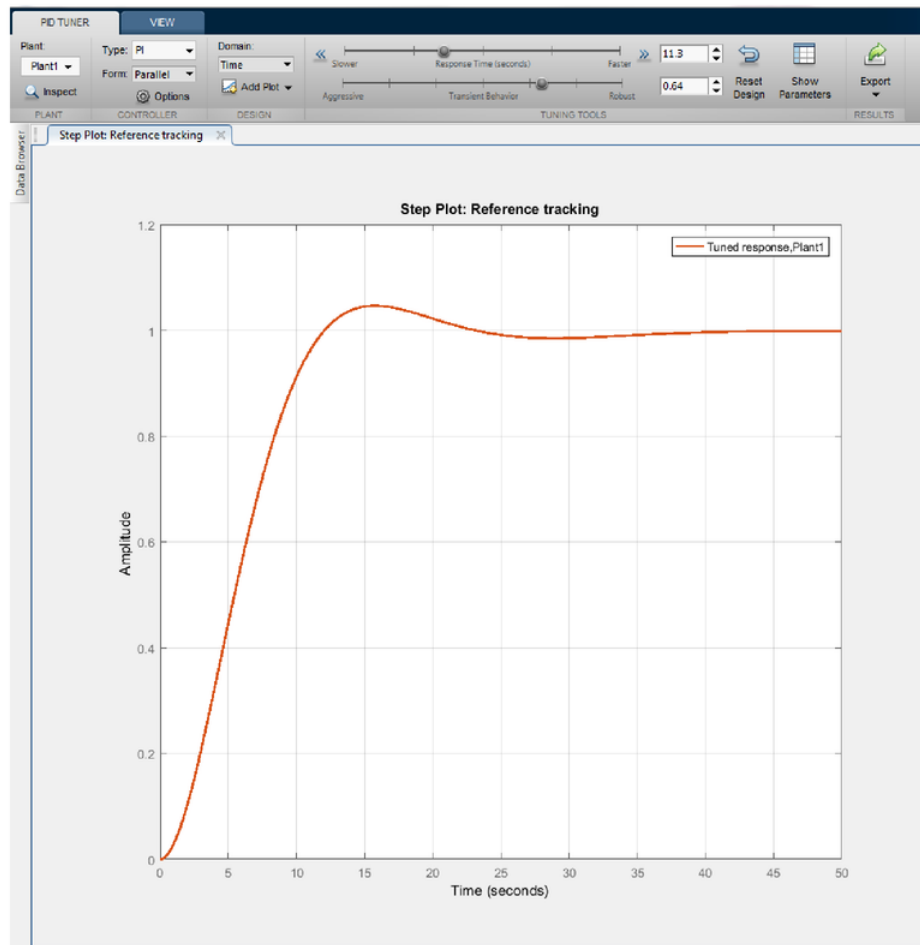


Figure 64: PI Control

Controller Parameters	
	Tuned
Kp	3.2426
Ki	0.2991
Kd	n/a
Tf	n/a
Performance and Robustness	
	Tuned
Rise time	7.84 seconds
Settling time	20.2 seconds
Overshoot	4.64 %
Peak	1.05
Gain margin	Inf dB @ Inf rad/s
Phase margin	64 deg @ 0.177 rad/s
Closed-loop stability	Stable

Figure 65: PI Control Performance

Using proportional and integral control does not meet the design specification. The overshoot of the system has however decreased slightly in comparison to proportional only control. However the response time is still out of specification (figure 65).

5.4.6.3. Proportional, Integral and Derivative

Proportional, Integral and derivative control was the final control used (figure 66).

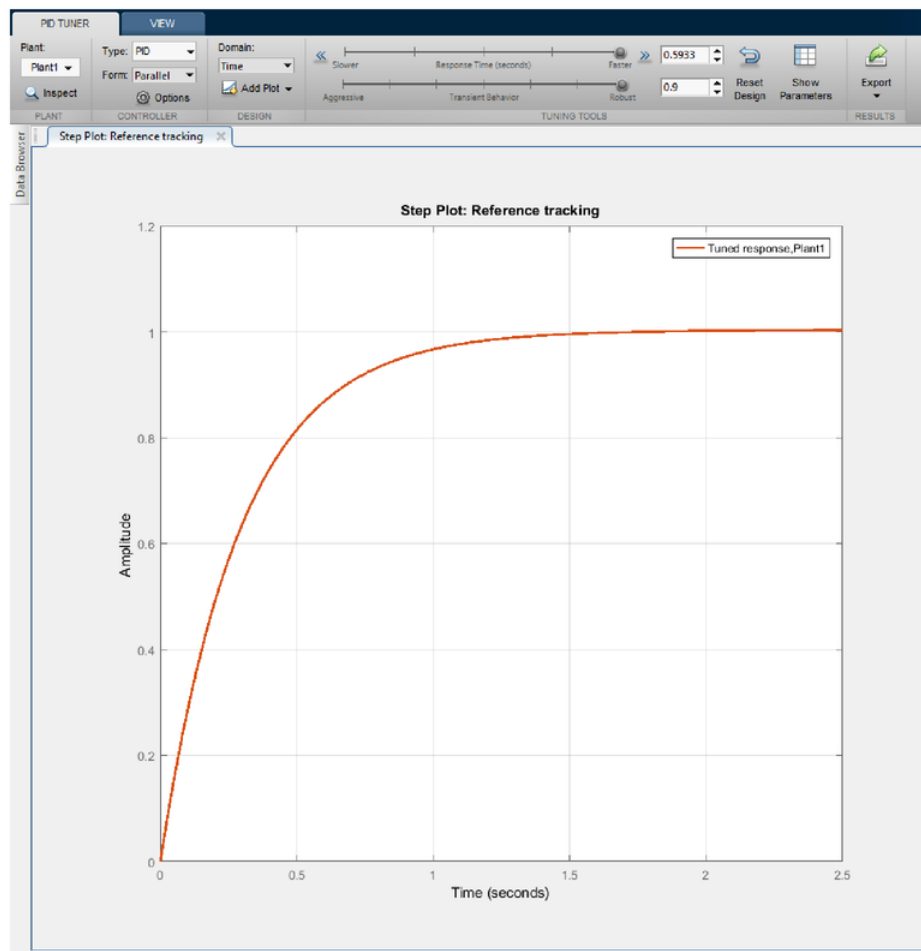


Figure 66: PID Control

Controller Parameters	
	Tuned
Kp	69.6542
Ki	7.1806
Kd	168.9168
Tf	n/a
Performance and Robustness	
	Tuned
Rise time	0.65 seconds
Settling time	1.14 seconds
Overshoot	0.413 %
Peak	1
Gain margin	Inf dB @ NaN rad/s
Phase margin	90 deg @ 3.37 rad/s
Closed-loop stability	Stable

Figure 67: PID Control Performance

The PID control meets the design specification. The total response time is quicker than both P and PI control with a total response time of just 1.79 seconds, undercutting the specification by just over 18 seconds. The overshoot is also decreased compared to other control methods with an overshoot of just 0.413%. This also is meeting the max overshoot specification of 5%.

Below is an outline of the final Kp, Ki and Kd values which will be entered in the PID controller.

$$K_p = 69.6542$$

$$K_i = 7.1806$$

$$K_d = 168.9168$$

5.4. Functionality Testing

Below shows photos of functionality testing by uploading the program to the PLC and screens.

5.4.1. Stop/Start

The first part tested was the stop start function of the system (figure 68 to 69).

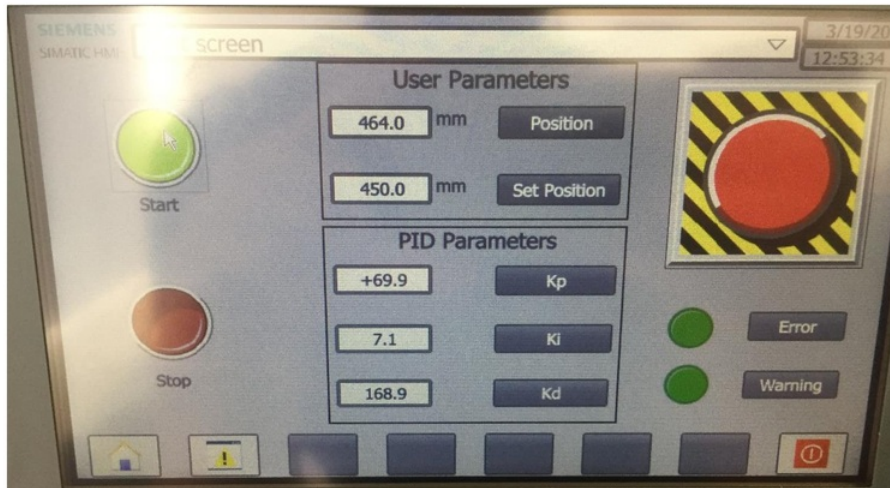


Figure 68: Start Function Testing



Figure 69: Stop Function Testing

This function worked accordingly and turned on and off the system with the indicators functioning as intended.

5.5.2. Analogue Input and Positional Output

The next test was to check the functionality of the analogue input and positional change by twiddling the potentiometer (figure 70). This was functioning successfully with the positional change in millimetres with a max reading of around 1000mm.

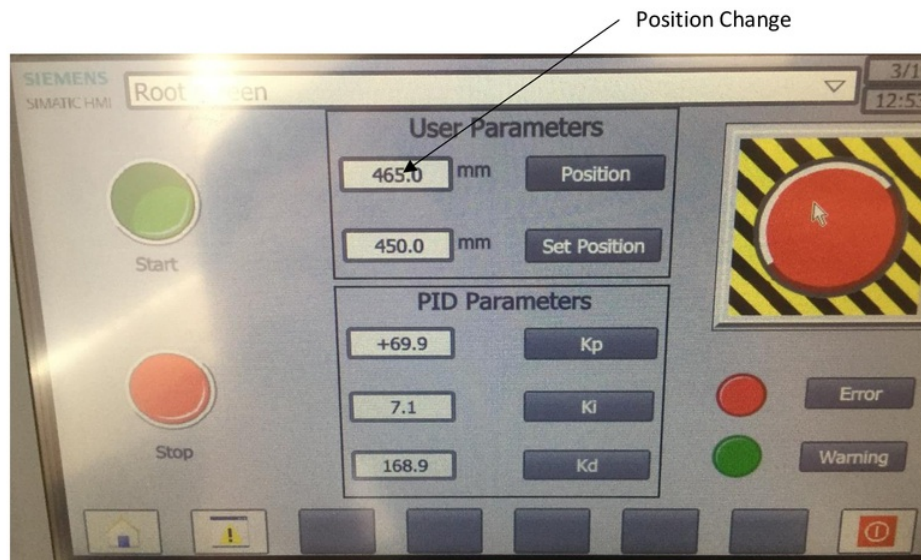


Figure 70: Analogue Positional Test

5.5.3. Emergency Stop

The emergency stop function was checked by starting the system then engaging the e-stop switch to check for immediate shut down. This was functioning successfully with the relevant light indicator switching on (figure 71 and 72).

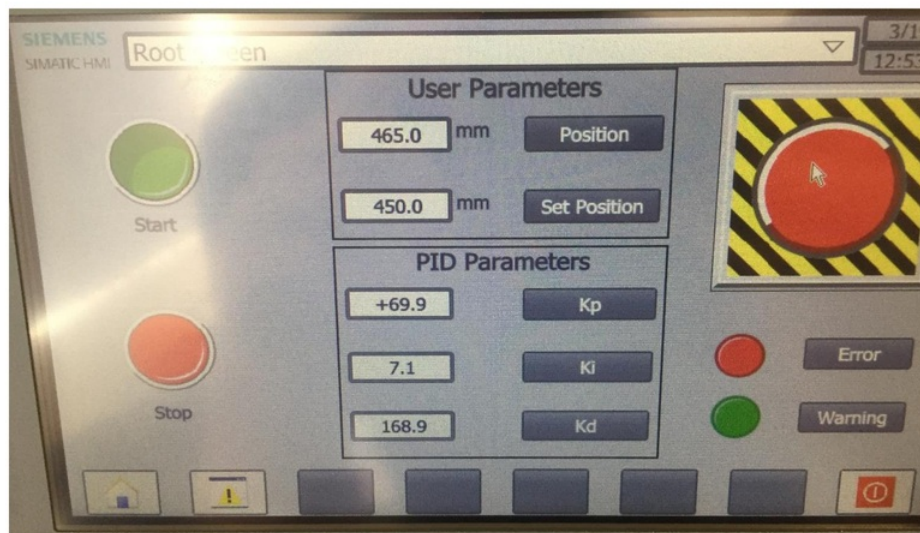


Figure 71: E-Stop Activation

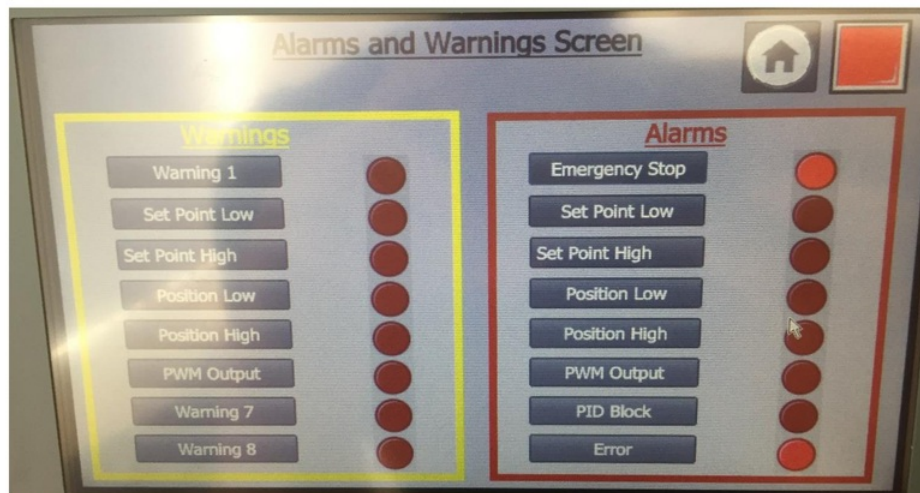


Figure 72: E-Stop Alarm Activation

5.5.4. Set point Warning and Error

Checks were made on the set point errors and warnings. This was achieved by setting the engineering screen up as shown below (figure 73).

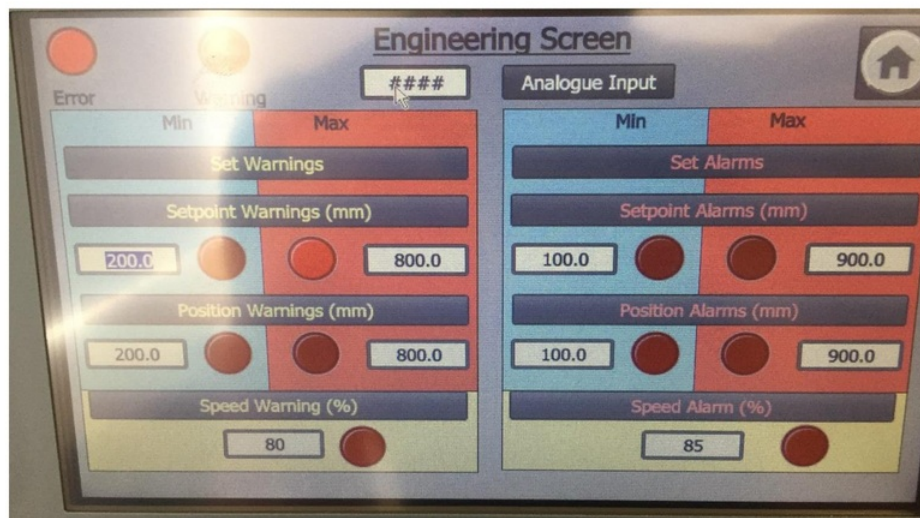


Figure 73: Setting Error and Warning Thresholds

After this the set point was entered and changed to above and below the values to check for each error and warning. This was achieved successfully (figure 74).

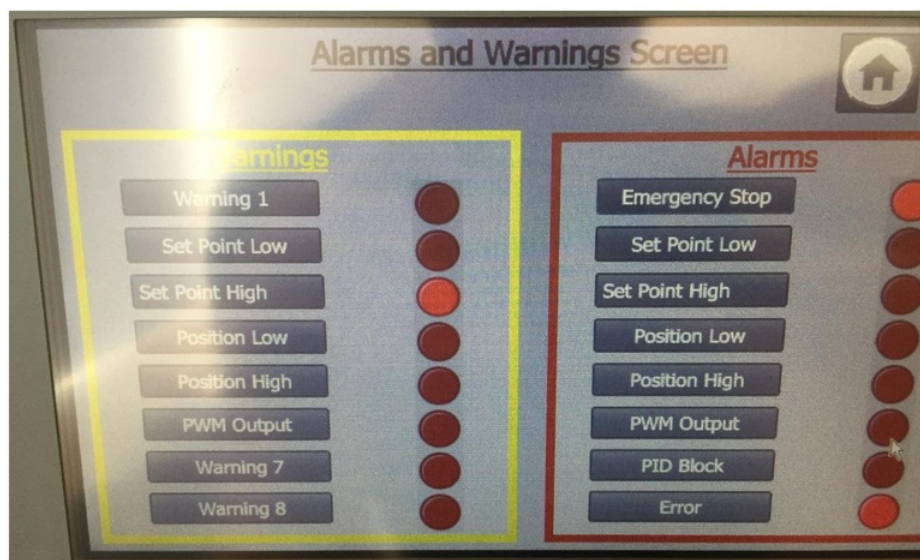


Figure 74: Set Point High Warning Activated

5.5.5. Position Input Error and Warnings

The next test was to check the positional error and warnings by twiddling the potentiometer so the position was below and above the set values. This was achieved successfully (figure 75 and 76).

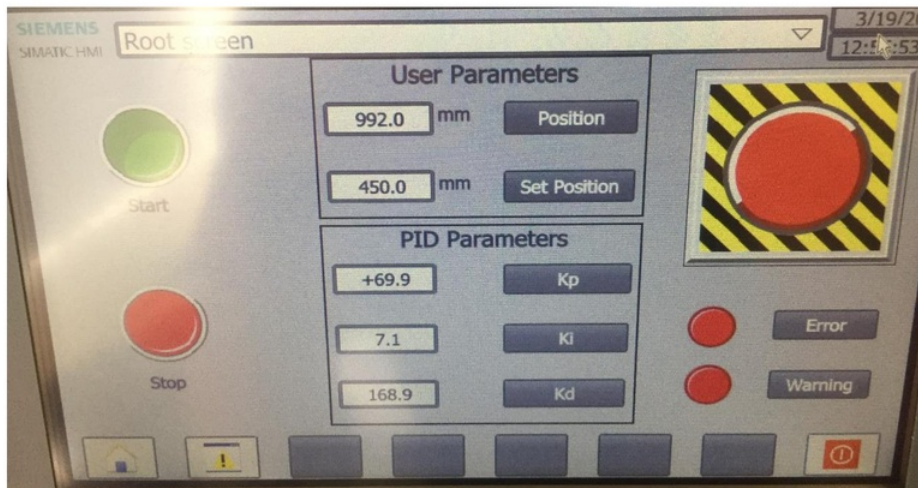


Figure 75: Position Greater than 900mm

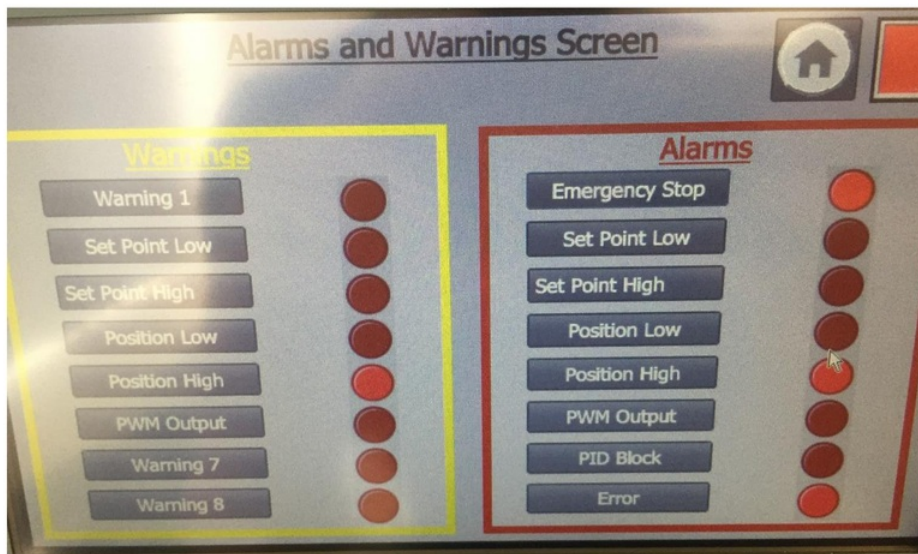


Figure 76: Error Position High

5.5.6. Error Flag Clear

With each error applied the system was checked for start-up. The system did not start when an error was presented. To clear the error the back button must be pressed whilst no other errors are present. This was checked and functioned accordingly (figure 77 and 78).

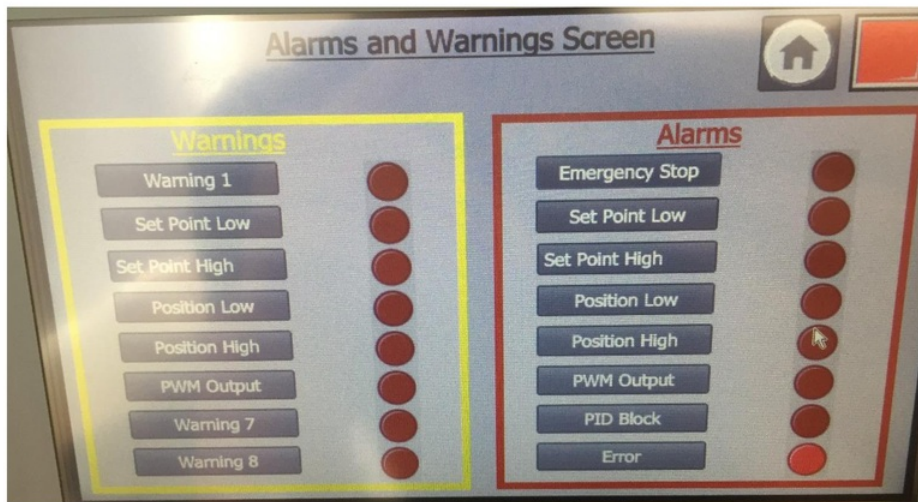


Figure 77: Error Flag Set

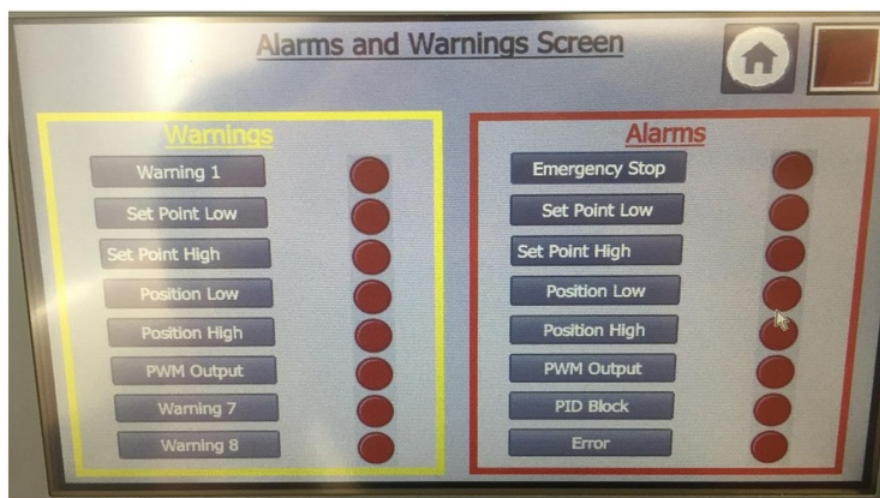


Figure 78: Error Flag Cleared After back is Activated

6. User Documentation

The following section presents details on error and warning audit documentation. Fault codes and warning codes are provided in each relevant sub section.

6.1. Error Codes

Presented below are all associated errors in the system including a recommended solution for rectifying the error (table 5).

Table 5: Error Fault Codes

Error Name	Description of Error	Solution/Checks to be made
Emergency Stop	Emergency Stop button has been activated.	Check to see if emergency has cleared. Release E-Stop to clear error
Set Point Low	Set Point boundary of the drill is less than Set Point min parameter (10%).	Check drill position. If the drill needs to be in that position edit set point min parameter on engineering screen.
Set Point High	Set point boundary of the drill is greater than set point max parameter (90%).	Check drill position. If the drill needs to be in that position edit set point max parameter on engineering screen.
Position Low	The position entered is less than position min error parameter (10%).	Check drill position. Increase position on user screen to be within parameters boundaries.
Position High	The position entered is greater than position max error parameter (90%).	Check drill position. Decrease position on user screen to be within error boundaries.
PWM output	PWM output is greater than 85% of max.	Consult an engineer and check PID controller output.
PID Block	PID error block has generated an error.	Consult engineer to check Kp, Ki and Kd parameters and PID controller.
Error	Error Flag is set due to one of above errors.	Clear all errors, once all cleared review the error screen. To clear flag return to user screen.

6.2. Warning Codes

Presented below are the warning codes associated with the system (table 6).

Table 6: Warnings Fault Codes

Warning Name	Description of Warning	Solution/Checks to be made
Warning 1	Free space for additional warning to added.	No checks to be made.
Set Point Low	Set Point boundary of the drill is less than Set Point min parameter (20%).	Check drill position. If the drill needs to be in that position edit set point min parameter on engineering screen.
Set Point High	Set point boundary of the drill is greater than set point max parameter (80%).	Check drill position. If the drill needs to be in that position edit set point max parameter on engineering screen.
Position Low	The position entered is less than position min error parameter (20%).	Check drill position. Increase position on user screen to be within parameters boundaries.
Position High	The position entered is greater than position max error parameter (80%).	Check drill position. Decrease position on user screen to be within warning boundaries.
PWM output	PWM output is greater than 80% of max.	Consult an engineer and check PID controller output.
Warning 7	Free space for additional warning to added.	No checks to be made.
Warning 8	Free space for additional warning to added.	No checks to be made.

7. Improvements and Recommendations

There are many improvements which may be made to the current system and design specification. These recommendations are outlined below.

7.1. Speed and Positional Data Graphs

The first recommendation to the system is to provide a graph screen to the user. This is believed to be beneficial as it would enhance the presentation of the display to the user, also allowing the user to monitor speed and positional data over a period of time. This could be beneficial to the engineer as if an error occurs they could pin point a moment in time when the systems speed or position is within error boundaries. To implement this increased graphical files and libraries are required. The system would be programmed by using the PWM output from the PID and using the positional millimetre conversion. An example of implementing this is shown below (figure 79).

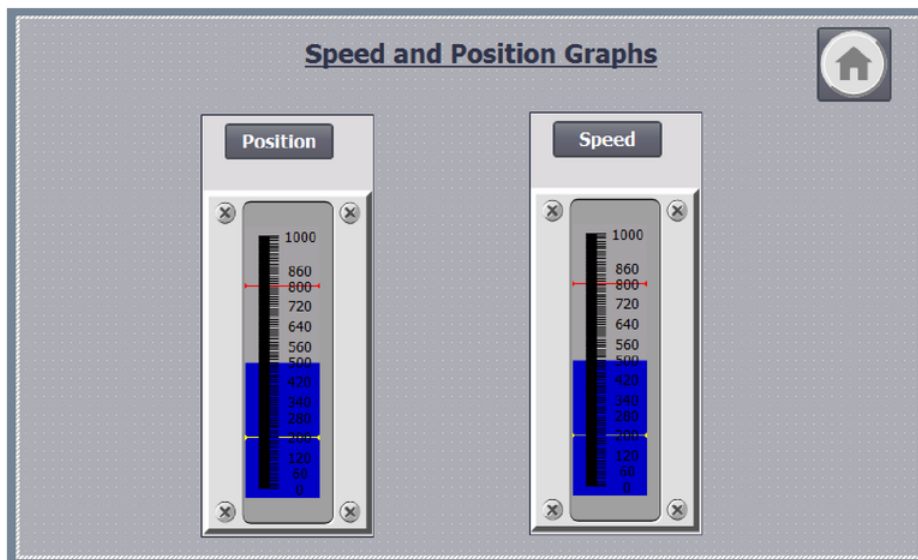


Figure 79: Speed and Position Graphs

7.2. User Login

With the currently designed system the user can access the user screen, engineer screen and alarms and warnings screen. In reality a standard operator should not be able to access the engineering screen. An engineer or an operator with higher clearance should be the only person able to access this screen. This can be achieved either by providing a system login or by a password to access the engineering screen. This can be implemented using an input in which the entered pin would have to match the stored password value. This would effectively lock the engineering screen until the correct password and account name have been entered.

7.3. Drill Position Display

This enhancement is similar to speed and positional graphs. A positional graphic could show the relative drill position. This could also highlight warning areas and error areas. This would improve the user interface and allow operators to view the drills position. Implementing this feature would require an additional screen which could be switched by the user. An example of the graphic layout is shown below (figure 80).

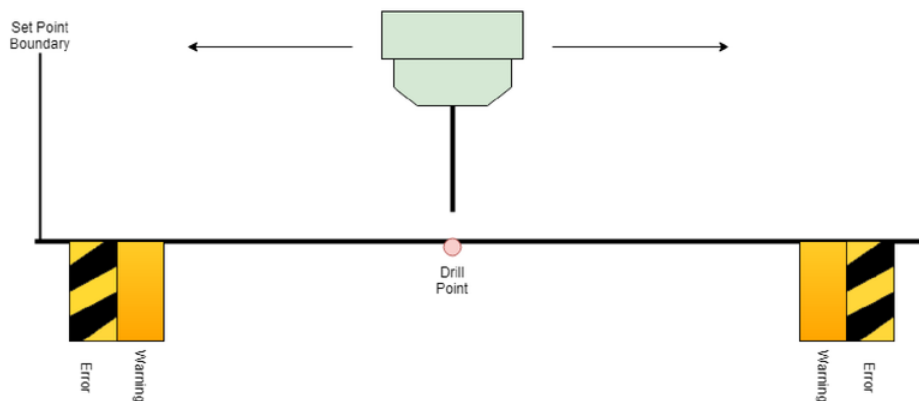


Figure 80: Drill Example Graphic

7.4. Error and Warning Logging

Error and warning logging would enhance the system by providing engineers with information on system performance and performance history. This would provide an aid to engineers to determine why faults may be occurring within the system. This feature could be implemented using the alarms logging function which is featured in TIA portal. This allows any alarm information to be saved to an external storage device.

8. Conclusion

Overall the designed system meets the proposed design specification. The system features all alarms and warnings which were proposed. The HMI screens designed provide the operator with control of the system as a whole.

The user screen provides inputs such as start, stop, E-Stop, positional input, current drill position information and error and warning indicators. Additionally the screen also displays PID controller parameters and are available as an input. The engineer screen allows the user to enter alarm and warning thresholds. The alarm and warnings screen provides a display of all warnings and alarms. In terms of functionality the system outputs correctly and all related errors and warnings are displayed correctly.

Although there are improvements which can be made to the system including a user login and enhanced displays the system functions accordingly to the design specification. Overall the system is regarded as a success with future additions recommended to further improve the system.

9. References

- [1] B. William, Programmable Logic Controllers, 6th, revised ed., Newnes, 2015, p. 424.
- [2] W. K. Edward, Industrial Controls and Manufacturing, Elsevier, 1999, p. 230.
- [3] Siemens AG, "SIMATIC S7-1200 CPUs," Siemens AG, [Online]. Available: <https://c4b.gss.siemens.com/resources/images/articles/dffa-b10053-03-7600.pdf>. [Accessed 22 March 2018].
- [4] V. Antonio, Practical PID Control, illustrated ed., Springer Science & Business Media, 2006.
- [5] K. J. Åström and H. Tore, "Advanced PID control," in *The Instrumentation, Systems, and Automation Society*, 2006.
- [6] National Instruments, "PID Theory Explained," 29 March 2011. [Online]. Available: <http://www.ni.com/white-paper/3782/en/>. [Accessed 3 March 2018].
- [7] B. Goodwine, "University of Notre Dame Aerospace and Mechanical Engineering: Control Systems Engineering," 22 January 2002. [Online]. Available: <https://controls.ame.nd.edu/ame437/S2003/hw1/>. [Accessed 5 March 2018].
- [8] F. S. Jacques, Process Control for Practitioners, OptiControls Inc, 2011.
- [9] K.-S. Tang, F. M. Kim, C. Guanrong and K. Sam, "An Optimal Fuzzy PID Controller," *IEEE transactions on industrial electronics*, vol. 48, no. 4, pp. 757-765, 2001.
- [10] A. J. Michael and H. M. Mohammad, PID Control: New Identification and Design Methods, illustrated ed., Springer Science & Business Media, 2006.
- [11] A. Tarun, "The Working Principle of a PID Controller for Beginners," [Online]. Available: <https://www.elprocus.com/the-working-of-a-pid-controller/>. [Accessed 12 April 2017].
- [12] Control Station, "How Does the Derivative Term Affect PID Controller Performance?," [Online]. Available: <https://controlstation.com/derivative-affect-pid-controller-performance/>. [Accessed 13 April 2018].
- [13] W. Y. Svrcek, D. P. Mahoney and B. R. Young, A Real-Time Approach to Process Control, 2nd ed., John Wiley & Sons, Ltd., 2006.
- [14] H. Finn, DYNAMICS and CONTROL, TechTeach, 2010.
- [15] M. Eric, Power Electronic Converters: PWM Strategies and Current Control Techniques, John Wiley & Sons, 2013.
- [16] S. Kohei, S. Mahesh, K. Jun-Koo, H. Masaki, W. Yifeng, K. Don and H. Jim, "Advantages of high frequency PWM in AC motor drive applications," *IEEE Energy Conversion Congress and Exposition (ECCE)*, 12 November 2012.

- [17] H. D. Grahame and A. L. Thomas, Pulse Width Modulation for Power Converters: Principles and Practice, vol. Volume 18 of IEEE Press Series on Power Engineering, John Wiley & Sons, 2003.
- [18] H. R. Muhammad, Power Electronics Handbook: Devices, Circuits and Applications, 2nd ed., Elsevier, 2010.
- [19] M. Boost and P. Ziogas, "State-of-the-art carrier PWM techniques: a critical evaluation," *IEEE Transactions on Industry Applications*, vol. 24, no. 2, pp. 271-280, 1988.
- [20] F. F. Mazda, Power Electronics Handbook, Elsevier, 1997.

10. Bibliography

- [1] A. D. Gary, Introduction to Programmable Logic Controllers, Cengage Learning, 2005.
- [2] G. Carlos, "Engineering Essentials: What Is a Programmable Logic Controller?," 1 June 2015. [Online]. Available: <http://www.machinedesign.com/engineering-essentials/engineering-essentials-what-programmable-logic-controller>. [Accessed 2018 Febraury 27].
- [3] V. Ramon and V. Antonio, PID Control in the Third Millennium: Lessons Learned and New Approaches, Springer Science & Business Media, 2012.
- [4] D. E. Rivera, M. Manfred and S. Sigurd, "Internal model control: PID controller design," *Industrial & engineering chemistry process design and development*, vol. 25, no. 1, pp. 252-265, 1986.
- [5] J. P. John, "How to best design an HMI system," 20 June 2015. [Online]. Available: <https://www.controleng.com/single-article/how-to-best-design-an-hmi-system/eee1d74711085cfd19cb15622c1f8968.html>. [Accessed 3 March 2018].
- [6] S. BDY, "An introduction to PID control with DC motor," 27 February 2017. [Online]. Available: <https://medium.com/pollenrobotics/an-introduction-to-pid-control-with-dc-motor-1fa3b26ec661>.

PID Control PLC Assignment

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

Instructor

73/100

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

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33

PAGE 34

PAGE 35

PAGE 36

PAGE 37

PAGE 38

PAGE 39

PAGE 40

PAGE 41

PAGE 42

PAGE 43

PAGE 44

PAGE 45

PAGE 46

PAGE 47

PAGE 48

PAGE 49

PAGE 50

PAGE 51

PAGE 52

PAGE 53

PAGE 54

PAGE 55

PAGE 56

PAGE 57

PAGE 58

PAGE 59

PAGE 60

PAGE 61

PAGE 62

PAGE 63

PAGE 64

PAGE 65

PAGE 66

PAGE 67

PAGE 68

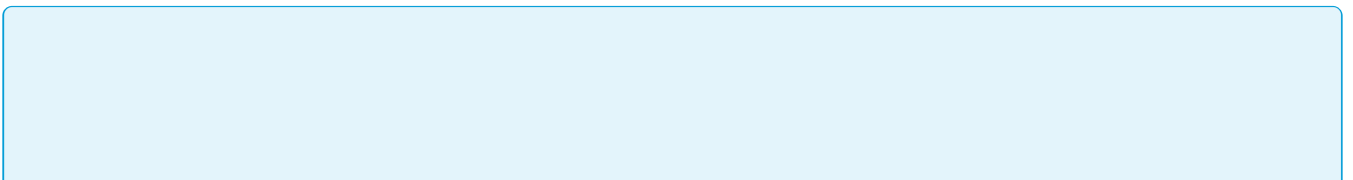
PAGE 69

PAGE 70

PAGE 71

PAGE 72

GRADING FORM: SMU GENERIC_GJ



RESEARCH-INFORMED LITERATURE



The literature review is clear with good background information a good selection of simple illustrations, but would have been improved with an advantage/disadvantage table. 75%

11.25

KNOWLEDGE AND UNDERSTANDING



The ladder logic program is clear and well documented. 70%

17.5

ANALYSIS



The mechanism for generating the transfer function from the plant input output response curve is very good. Your method is clear and well documented with good graphics. 72%

18

PRACTICAL APPLICATION AND DEP.



Your screens are functional and uncluttered and very well documented. All screens are nicely laid out and make it easy to identify faults/warnings and allow for user input and navigation. Good fault audit document. Good thought provoking improvements. 75%

15

SKILLS FOR PROFESSIONAL PRAC.



The report is well presented with the author using sufficient images and illustrations to allow the reader to digest information readily.75%

11.25