

Sam O'Leary- PLC Assignment 1

by Michael Sam O'Leary

FILE	2718_MICHAEL_SAM_OLEARY_SAM_OLEARY-_PLC_ASSIGNMENT_1_85309_1677102012.PDF (1.91M)		
TIME SUBMITTED	11-FEB-2018 08:41PM (UTC+0000)	WORD COUNT	5791
SUBMISSION ID	83062309	CHARACTER COUNT	31345

2018

PLC P.I.D Control System

MICHAEL SAM O'LEARY (P152108)

Abstract

This report contains the design of a P.I.D control system using a Siemens 1214 PLC for a linear positioning system. The linear positioning system is intended to control a drill used to machine holes in sheet metal. The program language used is ladder logic and also incorporates a HMI. A transfer function is created using MATLAB to create a response within a specification provided. The report is fully detailed with graphs showing how the transfer function and program was created. Advantages and disadvantages of using PWM over analogue is also included.

Contents

1.0	Introduction	4
2.0	Design of HMI Screens	5
3.0	PLC Program	7
4.0	Warnings & Fault Codes	16
4.1	Warnings	16
4.2	Errors	17
5.0	PID Control	18
6.0	PWM vs. Analogue	23
7.0	Operator Flow Chart	24
8.0	Conclusion	25

Figures

Figure 1 – HMI Root Screen	5
Figure 2 –HMI Settings Screen	5
Figure 3 – HMI Warnings Screen.....	6
Figure 4 – HMI Errors Screen	6
Figure 5 – Screenshot of PLC Tags	7
Figure 6 – Program Network 1 & 2	13
Figure 7 – Program Network 3 & 4	13
Figure 8 – Program Network 5.....	13
Figure 9 – Program Network 6.....	14
Figure 10 – Program Network 7	14
Figure 11 – Program Network 8.....	14
Figure 12 – Program Network 9.....	15
Figure 13 – Program Network 10.....	15
Figure 14 – Initial Control Parameters.....	18
Figure 15 - Open Loop Response Output Data	18
Figure 16 - Double Variable added to Control Parameters.....	18
Figure 17 - Step Response Setup	19
Figure 18 - One Pole Manual Step Response.....	19
Figure 19 - One Pole Transfer Function	19
Figure 20 - One Pole Auto Step Response	20
Figure 21 – Two Real Poles Manual Step Response.....	20
Figure 22 - Two Real Poles Auto Step Response.....	20
Figure 23 - Underdamped Pair Manual Step Response.....	20
Figure 24 - Underdamped Pair Auto Step Response	21
Figure 25 - Underdamped Pair & Real Pole Manual Step Response	21
Figure 26 - Underdamped Pair & Real Pole Auto Step Response	21
Figure 27 - Underdamped Pair & Real Pole Controller Performance	21
Figure 28 - Final Response Curve & Performance Data.....	22
Figure 29 - Final Transfer Function	22
Figure 30 - [2].....	23
Figure 31 - Operator Flow Chart	24

Tables

Table 1 – Table of PLC Tags.....	9
Table 2 – Table of HMI Tags.....	12
Table 3 – Table of Warning Codes	16
Table 4 – Table of Error Codes	17

1.0 Introduction

Contained in this report is the design of a P.I.D control for a servo motor driven linear positioning system using a Siemens PLC. The linear positioning system contains a drill head which is used to drill holes in heavy metal sheets. The Siemens based program incorporates a HMI which provides visual indications of the systems status including any present warnings and/or errors. Warnings and errors are generated when the system has exceeded minimum and maximum values set in the higher level access settings. A warnings and errors screen is included so the operator can see exactly why a warning or error was generated. An operator control screen allows the operator to start/stop the system and manually enter the required position of the drill while also displaying the current state of the system including speed and current position. Using MATLAB a transfer function of the system is generated within specific parameters and therefore the values of K_p , K_i , and K_d are produced for the P.I.D controller. A brief literature review is carried out which explains the operation of PWM and the advantages and disadvantages of using PWM over analogue.

2

2.0 Design of HMI Screens

This project must include various screens to display information of the system and to allow the user to enter set points for controlling the position of the drill. Figure 1 shows the root screen or 'Home' screen which is the main screen where the operator can start/stop the system, enter the set point, view the speed and position of the drill and view the 3-term control variables (PID). This screen also shows when there is an error or warning as the error or warning lights will illuminate. By selecting the error or warning icon the user will be taken to the corresponding screen to display the current warnings or errors that are present. The home screen also has a settings button which would be used for 'higher level' access to adjust various settings and limits within the program and is shown in figure 2. It was decided to include an emergency stop button for safety which will latch so that the system will not restart even if the start button is pressed. Pressing the E-stop will generate an error and only when the E-stop is reset and then the error is reset can the system be started again.

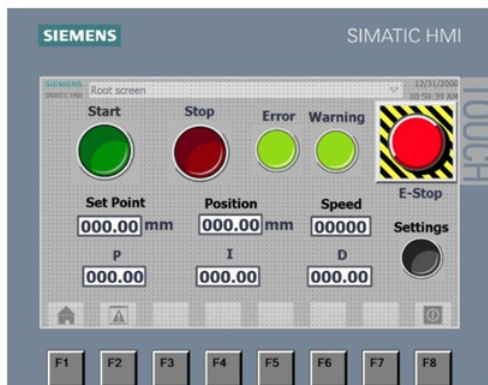


Figure 1 – HMI Root Screen

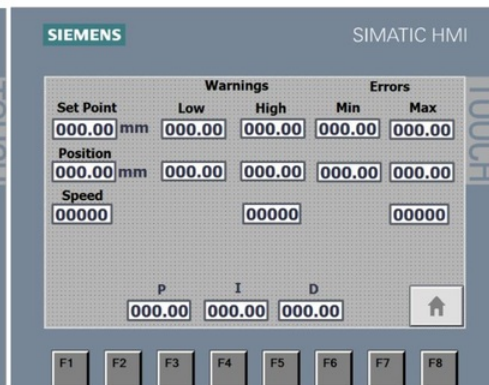


Figure 2 –HMI Settings Screen

The settings screen shown in figure 2 allows the higher level user to enter limits for the absolute min/max travel distance of the drill to prevent the system from operating outside of safe limits. An error will be generated either when the drill position or the entered set point is outside of the min/max limits set in the settings screen which stops the system. A warning will be generated either when the drill position or the entered set point is outside of the low/high warning values. For convenience the set point is also displayed on this screen and can be set to easily test the limits. The drill position, speed and PID values are also displayed. The home button in the bottom right corner allows the user to return to the home screen.

The warnings screen shown in figure 3 shows all the possible warnings within the program even if they aren't 'active'. If a warning is active then the icon next to the warning will change from green to red. Warnings are only active for as long as the warning is present and so they cannot be 'cleared'. For example: if the operator enters a set point that is greater than the 'warning high' limit then the 'set point close to maximum limit' will remain active until the set point is changed. Once the set point has been changed then the warning will automatically be cleared.

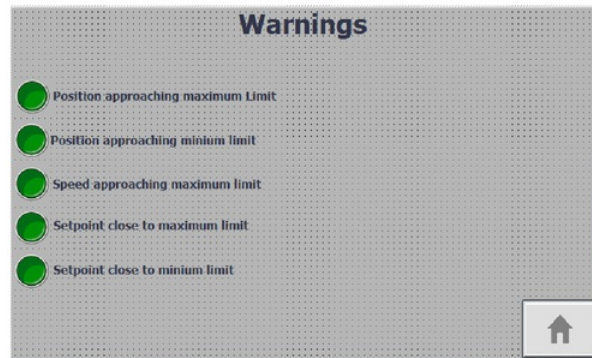


Figure 3 – HMI Warnings Screen

The Error screen shown in figure 4 shows all the possible errors within the program and the green icon will be red to indicate the currently active error or an error that occurred in the past that hasn't been cleared. Unlike the warnings which automatically clear, the errors remain active until the error has cleared and the home button on the errors screen is pressed. The home button on the error screen clears an error bit flag before going back to the home screen and so resets the errors providing the errors are no longer true.

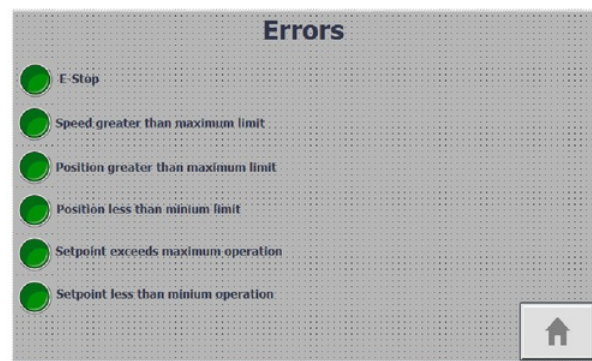


Figure 4 – HMI Errors Screen

3.0 PLC Program

The program language used to code the system is ladder logic but before the program can be written variables need to be declared. The screenshot below shows all the PLC tags;

	Name	Tag table	Data type	Address	Retain	Vioibl.	Acces
1	Optic 1	Default tag table	Bool	%I0.4			
2	Red Button	Default tag table	Bool	%I0.5			
3	Green Button	Default tag table	Bool	%I0.6			
4	Operate	Default tag table	Bool	%Q0.3			
5	Red Light Old	Default tag table	Bool	%I1.6			
6	Green Light	Default tag table	Bool	%I1.7			
7	estop	Default tag table	Bool	%M0.0			
8	Speed Max Error	Default tag table	Bool	%M0.1			
9	Position Max Error	Default tag table	Bool	%M0.2			
10	Position Min Error	Default tag table	Bool	%M0.3			
11	Setpoint Max Error	Default tag table	Bool	%M0.4			
12	Setpoint Min Error	Default tag table	Bool	%M0.5			
13	Error 6	Default tag table	Bool	%M0.6			
14	errorflag	Default tag table	Bool	%M0.7			
15	errors	Default tag table	Byte	%M0.8			
16	Position High Warning	Default tag table	Bool	%M1.0			
17	Position Low Warning	Default tag table	Bool	%M1.1			
18	Speed High Warning	Default tag table	Bool	%M1.2			
19	Set Point High Warning	Default tag table	Bool	%M1.3			
20	Set Point Low Warning	Default tag table	Bool	%M1.4			
21	Warning 5	Default tag table	Bool	%M1.5			
22	Warning 6	Default tag table	Bool	%M1.6			
23	Warning Flag	Default tag table	Bool	%M2.0			
24	Error Light	Default tag table	Bool	%M2.1			
25	Warning Light	Default tag table	Bool	%M2.2			
26	Warnings	Default tag table	Byte	%M81			
27	Analog Input	Default tag table	Int	%IM64			
28	Position (mm)	Default tag table	Real	%MD4			
29	Red Light	Default tag table	Bool	%Q0.5			
30	OUTPUT_PER	Default tag table	Int	%M88			
31	Setpoint High Warning Value	Default tag table	Real	%MD22			
32	Setpoint Low Warning Value	Default tag table	Real	%MD10			
33	Position High Warning Value	Default tag table	Real	%MD14			
34	Position Low Warning Value	Default tag table	Real	%MD18			
35	Setpoint Max Error Value	Default tag table	Real	%MD26			
36	Setpoint Min Error Value	Default tag table	Real	%MD30			
37	Position Max Error Value	Default tag table	Real	%MD34			
38	Position Min Error Value	Default tag table	Real	%MD38			
39	Speed High Warning Value	Default tag table	Real	%MD42			
40	Speed Max Error Value	Default tag table	Real	%MD46			

Figure 5 – Screenshot of PLC Tags

The table below shows all the PLC tags with a description next to each;

Tag N ^o	Name	Data Type	Address	Description
1	Optic 1	Bool	%I0.4	Optic 1 was a sensor that is no longer used in this project.
2	Red Button	Bool	%I0.5	Red button is used as a stop button which when active stops the system.
3	Green Button	Bool	%I0.6	Green button is used to start the system.
4	Operate	Bool	%Q0.3	Operate is an output which allows the system to latch 'on' after the green button is pressed.
5	Red Light Old	Bool	%I1.6	No longer used in this project.
6	Green Light	Bool	%I1.7	Green Light is used to show the operator that the system is running.
7	Estop	Bool	%M0.0	Estop is the emergency stop that must stop the system when active.
8	Speed Max Error	Bool	%M0.1	Error generated when the drill speed exceeds maximum limit.

P.I.D Control System

9	Position Max Error	Bool	%M0.2	Error generated when position exceeds maximum limit.
10	Position Min Error	Bool	%M0.3	Error generated when position exceeds minimum limit.
11	Setpoint Max Error	Bool	%M0.4	Error generated when operators setpoint exceeds the maximum limit.
12	Setpoint Min Error	Bool	%M0.5	Error generated when operators setpoint exceeds the minimum limit.
13	Error 6	Bool	%M0.6	An Error code reserved for future use.
14	Errorflag	Bool	%M0.7	Used to latch the errors so they are not automatically cleared.
15	Errors	Byte	%MB0	This tag is assigned to the MB0 byte and so is active when any of the error bits are true.
16	Position High Warning	Bool	%M1.0	Warning generated when position exceeds high limit.
17	Position Low Warning	Bool	%M1.1	Warning generated when position exceeds low limit.
18	Speed High Warning	Bool	%M1.2	Warning generated when speed exceeds high limit.
19	Set Point High Warning	Bool	%M1.3	Warning generated when operators setpoint exceeds high limit.
20	Set Point Low Warning	Bool	%M1.4	Warning generated when operators setpoint exceeds low limit.
21	Warning 5	Bool	%M1.5	A warning code reserved for future use.
22	Warning 6	Bool	%M1.6	A warning code reserved for future use.
23	Warning Flag	Bool	%M2.0	Use to control the warning light.
24	Error Light	Bool	%M2.1	Error light is active when any errors are true to give a visual indication to the operator that an error is present.
25	Warning Light	Bool	%M2.2	Warning light is active when any warnings are true to give a visual indication to the operator that a warning is present.
26	Warnings	Byte	%MB1	This tag is assigned to the MB1 byte and so is active when any of the warning bits are true.
27	Analog_Input	Int	%IW64	This is an analog input from the position sensor
28	Position (mm)	Real	MD4	This is a real number used to convert the analog input from the position sensor into millimeters so the position of

P.I.D Control System

				the drill can be displayed in millimeters.
29	Red Light	Bool	%Q0.5	Red Light is used to show the operator that the system is stopped.
30	OUTPUT_PER	Int	%MW8	This is an integer value from the P.I.D controller PER outout that monitors the speed.
31	Setpoint High Warning Value	Real	%MD22	This is a byte used to compare the 'setpoint high warning' value with the setpoint.
32	Setpoint Low Warning Value	Real	%MD10	This is a byte used to compare the 'setpoint low warning' value with the setpoint.
33	Position High Warning Value	Real	%MD14	This is a byte used to compare the 'Position High Warning value with the position.
34	Position Low Warning Value	Real	%MD18	This is a byte used to compare the 'Position Low Warning' value with the position.
35	Setpoint Max Error Value	Real	%MD26	This is a byte used to compare the 'Setpoint Max Error' value with the setpoint.
36	Setpoint Min Error Value	Real	%MD30	This is a byte used to compare the 'Setpoint Min Error' value with the setpoint.
37	Position Max Error Value	Real	%MD34	This is a byte used to compare the 'Position Max Error' value with the position.
38	Position Min Error Value	Real	%MD38	This is a byte used to compare the 'Position Min Error' value with the position.
39	Speed High Warning Value	Real	%MD42	This is a byte used to compare the 'Speed High Warning' value with the speed.
40	Speed Max Error Value	Real	%MD46	This is a byte used to compare the 'Speed Max Error' value with the speed.

Table 1 – Table of PLC Tags

P.I.D Control System

Below is a screenshot of the HMI tags;

HMI tags					
Name	Tag table	Data type	Connection	PLC name	PLC tag
Analog_Input	Default tag table	Int	HMI_Conne...	PLC_1	Analog_Input
Conveyor	Default tag table	Bool	HMI_Conne...	PLC_1	Operate
Error 0	Default tag table	Bool	HMI_Conne...	PLC_1	estop
Error 1	Default tag table	Bool	HMI_Conne...	PLC_1	"Speed Max Error"
Error 2	Default tag table	Bool	HMI_Conne...	PLC_1	"Position Max Error"
Error 3	Default tag table	Bool	HMI_Conne...	PLC_1	"Position Min Error"
Error 4	Default tag table	Bool	HMI_Conne...	PLC_1	"Setpoint Max Error"
Error 5	Default tag table	Bool	HMI_Conne...	PLC_1	"Setpoint Min Error"
Error 6	Default tag table	Bool	HMI_Conne...	PLC_1	"Error 6"
Error Light	Default tag table	Bool	HMI_Conne...	PLC_1	"Error Light"
errorflag	Default tag table	Bool	HMI_Conne...	PLC_1	errorflag
Millimeters	Default tag table	Real	HMI_Conne...	PLC_1	"Position (mm)"
OUTPUT_PER	Default tag table	Int	HMI_Conne...	PLC_1	OUTPUT_PER
PID_Compact_1_Retain_CtrlPar...	Default tag table	Real	HMI_Conne...	PLC_1	PID_Compact_1_Retain Ct...
PID_Compact_1_Retain_CtrlPar...	Default tag table	Real	HMI_Conne...	PLC_1	PID_Compact_1_Retain Ct...
PID_Compact_1_Retain_CtrlPar...	Default tag table	Real	HMI_Conne...	PLC_1	PID_Compact_1_Retain Ct...
PID_Compact_1_Setpoint	Default tag table	Real	HMI_Conne...	PLC_1	PID_Compact_1_Setpoint
Position High Warning	Default tag table	Real	HMI_Conne...	PLC_1	"Position High Warning V...
Position Low Warning	Default tag table	Real	HMI_Conne...	PLC_1	"Position Low Warning V...
Position Max Error	Default tag table	Real	HMI_Conne...	PLC_1	"Position Max Error Value"
Position Min Error	Default tag table	Real	HMI_Conne...	PLC_1	"Position Min Error Value"
Setpoint High Warning	Default tag table	Real	HMI_Conne...	PLC_1	"Setpoint High Warning V...
Setpoint Low Warning	Default tag table	Real	HMI_Conne...	PLC_1	"Setpoint Low Warning V...
Setpoint Max Error	Default tag table	Real	HMI_Conne...	PLC_1	"Setpoint Max Error Value"
Setpoint Min Error	Default tag table	Real	HMI_Conne...	PLC_1	"Setpoint Min Error Value"
Speed High Warning	Default tag table	Real	HMI_Conne...	PLC_1	"Speed High Warning Val...
Speed Max Error	Default tag table	Real	HMI_Conne...	PLC_1	"Speed Max Error Value"
Tag_ScreenNumber	Default tag table	UInt	<Internal tag>		<Undefined>
Warning 0	Default tag table	Bool	HMI_Conne...	PLC_1	"Position High Warning"
Warning 1	Default tag table	Bool	HMI_Conne...	PLC_1	"Position Low Warning"
Warning 2	Default tag table	Bool	HMI_Conne...	PLC_1	"Speed High Warning"
Warning 3	Default tag table	Bool	HMI_Conne...	PLC_1	"Set Point High Warning"
Warning 4	Default tag table	Bool	HMI_Conne...	PLC_1	"Set Point Low Warning"
Warning 5	Default tag table	Bool	HMI_Conne...	PLC_1	"Warning 5"
Warning 6	Default tag table	Bool	HMI_Conne...	PLC_1	"Warning 6"
Warning Light	Default tag table	Bool	HMI_Conne...	PLC_1	"Warning Light"

It can be seen that each HMI tag is linked with a PLC tag which is how each icon on the HMI screens are integrated with the PLC program. Below is a table showing the HMI tags with a description next to each;

Tag N°	Name	Data Type	PLC Tag	Description
1	Analog_Input	Int	Analog_Input	This was used to display the analog input from the position sensor during testing but is no longer displayed on the HMI except in millimeters (See Tag N° 12).
2	Conveyor	Bool	Operate	Originally named 'Conveyor' is used to light up the start icon on the root screen to show the system is running.
3	Error 0	Bool	Estop	Error 0 is the first error and is active when the Estop has been pressed.
4	Error 1	Bool	Speed Max Error	Error 1 is the second error and is active when the speed has exceeded the value entered in the 'Speed Max Error' box in the settings screen.
5	Error 2	Bool	Position Max Error	Error 2 is the third error and is active when the position has exceeded the value entered in the 'Position Max Error' box in the settings screen.
6	Error 3	Bool	Position Min Error	Error 3 is the fourth error and is active when the position has

P.I.D Control System

				exceeded the value entered in the 'Position Min Error' box in the settings screen.
7	Error 4	Bool	Setpoint Max Error	Error 4 is the second error and is active when the set point has exceeded the value entered in the 'Setpoint Max Error' box in the settings screen.
8	Error 5	Bool	Setpoint Min Error	Error 5 is the sixth error and is active when the set point has exceeded the value entered in the 'Setpoint Min Error' box in the settings screen.
9	Error 6	Bool	Error 6	Error 6 is reserved for future use.
10	Error Light	Bool	Error Light	Error light is the error icon on the root screen and is active when any errors are true.
11	Errorflag	Bool	Errorflag	Used so when the home button is pressed on the error screen the errorflag gets cleared.
12	Millimeters	Real	Position (mm)	Displays the position in mm on both the root screen and settings screen.
13	OUTPUT_PER	Int	OUTPUT_PER	Displays the speed on both the root screen and settings screen.
14	PID_Compact_1_Retain_CtrlPar...	Real	PID_Compact_1_Retain_Ct...	These three tags are used to display the P, I and D values on both the root screen and settings screen.
15	PID_Compact_1_Retain_CtrlPar...	Real	PID_Compact_1_Retain_Ct...	
16	PID_Compact_1_Retain_CtrlPar...	Real	PID_Compact_1_Retain_Ct...	
17	PID_Compact_1_Setpoint	Real	PID_Compact_1_Setpoint	Takes the users entered set point and inputs into the P.I.D controller.
18	Position High Warning	Real	Position High Warning Value	Takes the entered input from the 'position high warning' box in the settings screen.
19	Position Low Warning	Real	Position Low Warning Value	Takes the entered input from the 'Position Low Warning' box in the settings screen.
20	Position Max Error	Real	Position Max Error Value	Takes the entered input from the 'Position Max Error' box in the settings screen.
21	Position Min Error	Real	Position Min Error Value	Takes the entered input from the 'Position Min Error' box in the settings screen.
22	Setpoint High Warning	Real	Setpoint High Warning Value	Takes the entered input from the 'Setpoint High Warning' box in the settings screen.

P.I.D Control System

23	Setpoint Low warning	Real	Setpoint Low Warning Value	Takes the entered input from the 'Setpoint Low warning' box in the settings screen.
24	Position Max Error	Real	Setpoint Max Error Value	Takes the entered input from the 'Position Max Error' box in the settings screen.
25	Position Min Error	Real	Setpoint High Warning Value	Takes the entered input from the 'Position Min Error' box in the settings screen.
26	Speed High Warning	Real	Speed High Warning Value	Takes the entered input from the 'Speed High Warning' box in the settings screen.
27	Speed Max Error	Real	Speed Max Error Value	Takes the entered input from the 'Speed Max Error' box in the settings screen.
28	Tag_ScreenNumber	UInt	Undefined	No longer used in this project.
29	Warning 0	Bool	Position High Warning	Warning 0 is the first warning and is active when the position has exceeded the value entered in the 'position high warning' box in the settings screen.
30	Warning 1	Bool	Position Low Warning	Warning 1 is the second warning and is active when the position has exceeded the value entered in the 'Position Low Warning' box in the settings screen.
31	Warning 2	Bool	Speed High Warning	Warning 2 is the warning and is active when the speed has exceeded the value entered in the 'Speed High Warning' box in the settings screen.
32	Warning 3	Bool	Set point High Warning	Warning 3 is the fourth warning and is active when the set point has exceeded the value entered in the 'Set point High Warning' box in the settings screen.
33	Warning 4	Bool	Set point Low Warning	Warning 4 is the fifth warning and is active when the set point has exceeded the value entered in the 'Set point Low Warning' box in the settings screen.
34	Warning 5	Bool	Warning 5	Warning 5 reserved for future use.
35	Warning 6	Bool	Warning 6	Warning 6 reserved for future use.
36	Warning Light	Bool	Warning Light	Warning light is the warning icon on the root screen and is active when any warnings are true.

Table 2 – Table of HMI Tags

P.I.D Control System

Firstly, a start/stop latching control was implemented so that the start/stop buttons can start and stop the system with just one touch. When the green button is pressed on the root screen the normally open (NO) contact closes which sets the 'operate' output on. This causes the 'operate' NO contact to close which by-passes the 'green button' contact. When the operator lets go of the green button the 'operate' contact is keeping the system running until either the 'red button' or 'errorflag' normally closed (NC) contacts open. By using the errorflag the system will stop regardless of which error has occurred. Network 2 allows the error and warning flags to be set by using an error and warning byte to set a flag bit. If any of the error or warning bits are set then so will the error and warning bytes which then set the corresponding flags.

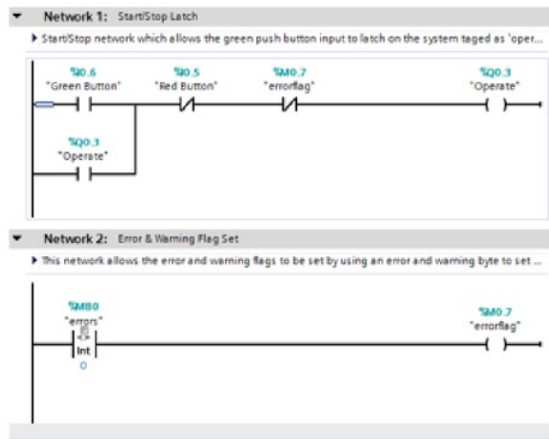


Figure 6 – Program Network 1 & 2

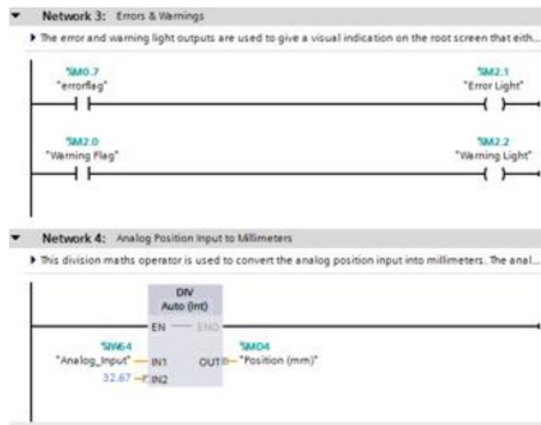


Figure 7 – Program Network 3 & 4

Network 5 is where the PID controller is used which takes the position of the drill and compares it to the set point of where the drill needs to move and creates an error rate. The further the value is from the set point the larger the error rate. The PID controls the motor which moves the drill to the required position and as it gets closer the error rate decreases. When the drill is in the correct position there will be no error and so the movement of the drill stops.

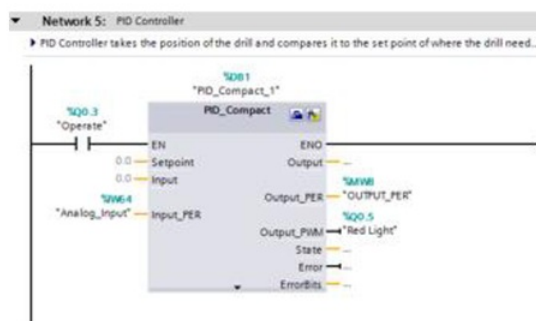


Figure 8 – Program Network 5

P.I.D Control System



Figure 9 – Program Network 6

Network 6 is where the set point min and max warnings are generated. Comparators are used to compare the set point with the 'Setpoint High Warning Value' and 'setpoint Low Warning Value' that are set in the settings screen. If the set point is greater than or equal to 'Setpoint High Warning Value' then the 'Set Point High Warning' is generated. If the set point is less than or equal to the 'Setpoint Low Warning Value' then the 'Set Point Low Warning' is generated.

Network 7 is where the set point min and max errors are generated. Just like in network 6 comparators are used to compare the set point with the min and max values set in the settings screen. If the set point is greater than or equal to 'Setpoint Max Error Value' then 'Setpoint Max Error' is generated. If the set point is less than or equal to 'Setpoint Min Error Value' then 'Setpoint Min Error' is generated.

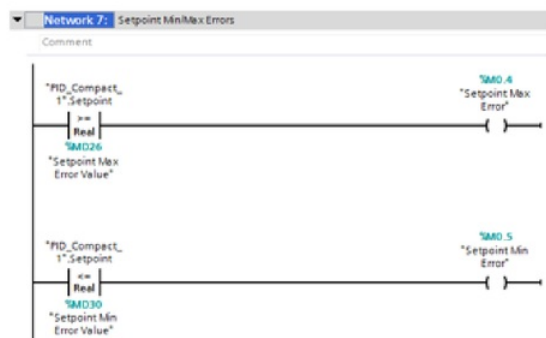


Figure 10 – Program Network 7



Figure 11 – Program Network 8

Network 8 is where the position min and max warnings are generated, again by using comparators to compare the position with the values entered in the settings screen. If the 'position (mm)' is greater than or equal to 'Position High Warning Value' then 'Position High Warning' is generated. If 'Position (mm)' is less than or equal to 'Position Low Warning Value' then 'Position Low Warning' is generated.

P.I.D Control System



Figure 12 – Program Network 9

Finally, network 10 is where the speed high warning and max error is generated. If 'OUTPUT_PER' is greater than or equal to 'Speed High Warning Value' then 'Speed High Warning' is generated. If 'OUTPUT_PER' is greater than or equal to 'Speed Max Error Value' then 'Speed Max Error' is generated.

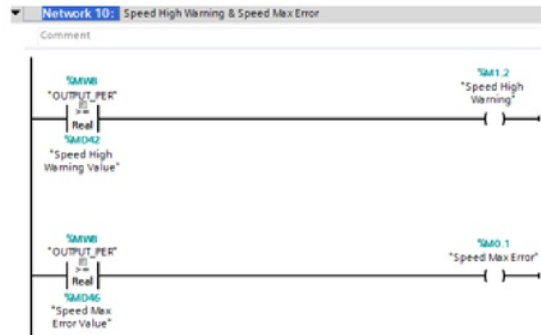


Figure 13 – Program Network 10

4.0 Warnings & Fault Codes

4.1 Warnings

Below is a table showing all the warning codes implemented in the software and what they mean from an operators perspective;

Warning	Description
Position approaching maximum limit	The drill position is getting close to the maximum position limit – care should be taken to ensure the drill position does not go into dangerous boundaries. Check the drill position is in the correct location.
Position approaching minimum limit	The drill position is getting close to the minimum position limit – care should be taken to ensure the drill position does not go into dangerous boundaries. Check the drill position is in the correct location.
Speed approaching maximum limit	The speed of the system is getting close to the maximum limit - The speed should be carefully monitored and the system should be checked for any abnormalities.
Setpoint close to maximum limit	The entered set point is close to the maximum limit -If possible the set point should be decreased. If it is not possible to decrease the set point then care should be taken while operating the system to ensure the drill position does not enter dangerous boundaries.
Setpoint close to minimum limit	The entered set point is close to the minimum limit - If possible the set point should be increased. If it is not possible to increase the set point then care should be taken while operating the system to ensure the drill position does not enter dangerous boundaries.

Table 3 – Table of Warning Codes

Although the system will continue to operate during a warning, the warning codes should still be checked and if the warnings codes are persistent then investigations should be carried out.

4.2 Errors

Error	Description
E-Stop	The emergency stop has been activated and the system will not operate until it has been reset – the emergency stop can be reset by pressing it again. Before resetting the emergency stop ensure it is safe to do so.
Speed greater than maximum limit	The speed of the system has reached the maximum limit and has stopped the system for safety– Ensure the system has been correctly maintained.
Position greater than maximum limit	The drill position has hit the maximum position limit and has stopped the system for safety. The drill must be manually moved out of the dangerous boundary before the system can be reset.
Position less than minimum limit	The drill position has hit the minimum position limit and has stopped the system for safety. The drill must be manually moved out of the dangerous boundary before the system can be reset.
Setpoint exceeds maximum operation	The entered set point is greater than the maximum limit allowed – Decrease the set point.
Setpoint less than minimum operation	The entered set point is less than the minimum limit allowed – Increase the set point.

Table 4 – Table of Error Codes

All error codes cause the system to stop. Investigations must be carried out after an error code as safety may be compromised. Once investigations have been carried out and it is safe to reset the system then the home button can be pressed in the error screen which will reset the error(s).

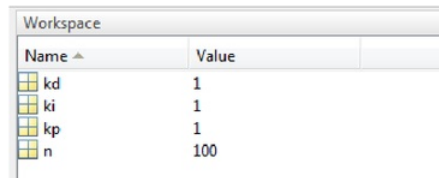
5.0 PID Control

A transfer function is to be generated using MATLAB with the parameters to produce a control system that operates within the following specification;

- Absolute Max Response time to steady state = 20secs.
- Absolute Max Over-shoot to a step-change = 5%

The initial parameters were set as follows;

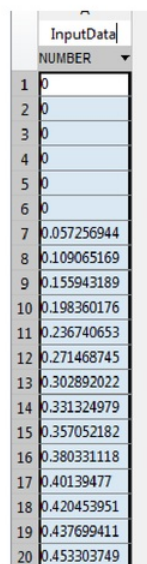
- $k_d = 1$
- $k_i = 1$
- $k_p = 1$
- $n = 100$



Name	Value
kd	1
ki	1
kp	1
n	100

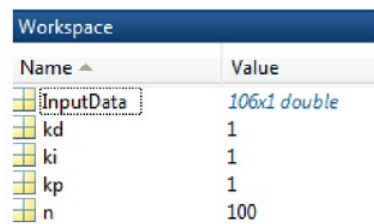
Figure 14 – Initial Control Parameters

A file was then given which contains the output data from an open loop response to a unity step change of the system input. This data was imported into MATLAB and added as a double variable;



NUMBER	InputData
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
16	0.380331118
17	0.40139477
18	0.420453951
19	0.437699411
20	0.453303749

Figure 15 - Open Loop Response Output Data



Name	Value
InputData	106x1 double
kd	1
ki	1
kp	1
n	100

Figure 16 - Double Variable added to Control Parameters

P.I.D Control System

The PID tool in MATLAB was launched where a step response can be created;

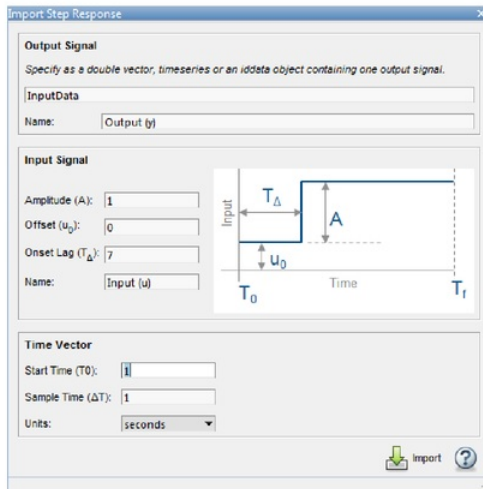


Figure 17 - Step Response Setup

As can be seen in figure 16 there is a lag of 7 so the Onset Lag as shown in figure 17 was set to 7. The amplitude was set to 1, offset to 0, start time and sample time both set to 1 second.

one pole step response was A created and it can be seen in figure 18 that the green line is the target and the blue line is the actual response. There is one pole that can be manually adjusted and the screenshot shows the best adjustment that could be made manually without any significant overshoot.

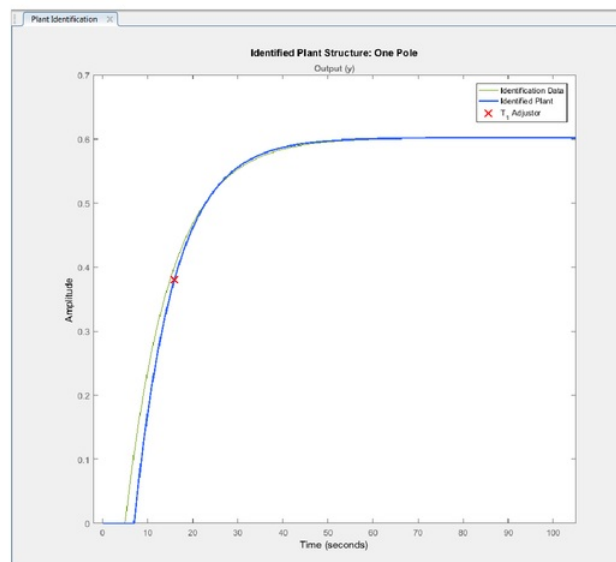


Figure 18 - One Pole Manual Step Response

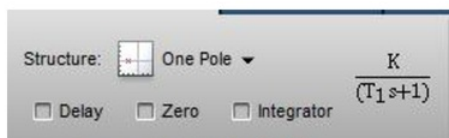


Figure 19 - One Pole Transfer Function

Figure 19 shows the transfer function for the one pole step response.

MATLAB has an 'auto estimate' tool which uses software to automatically adjust the response to as close as it can. Figure 20 shows the result;

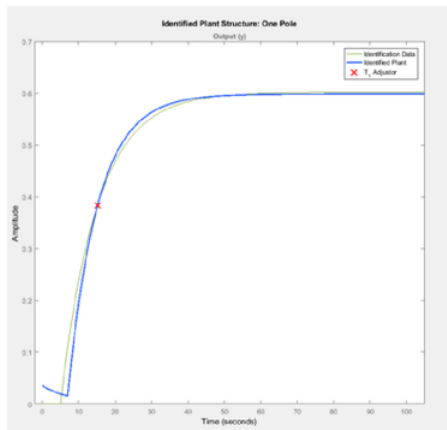


Figure 20 - One Pole Auto Step Response

In an attempt to get better results, the structure was changed from one pole to 'two real poles' and figure 21 shows the result using manual adjustment;

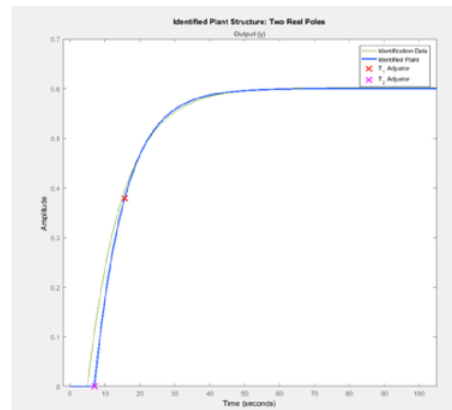


Figure 21 – Two Real Poles Manual Step Response

Figure 22 shows the result of the auto estimate when using two real poles;

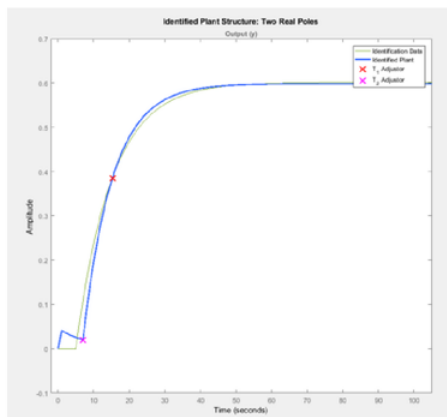


Figure 22 - Two Real Poles Auto Step Response

Figure 23 shows the result from manually adjusting the step response using 'underdamped pair' structure;

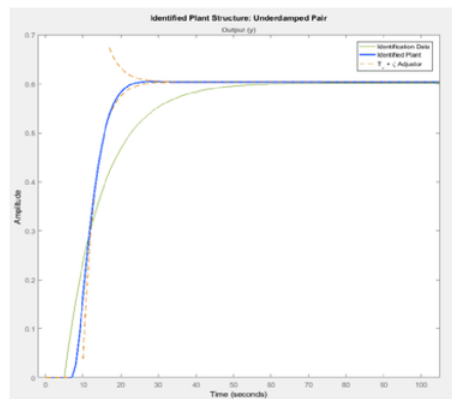


Figure 23 - Underdamped Pair Manual Step Response

Figure 24 shows the underdamped pair auto estimate result;

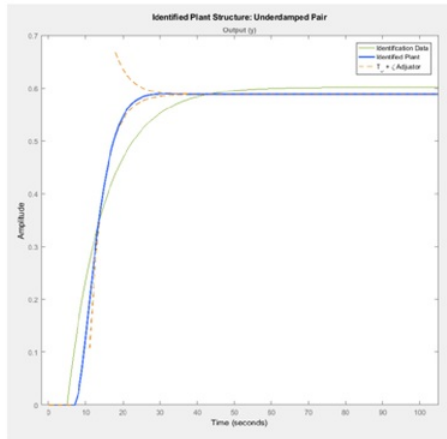


Figure 24 - Underdamped Pair Auto Step Response

Structure changed to 'Underdamped Pair & Real Pole' with manual adjustment;

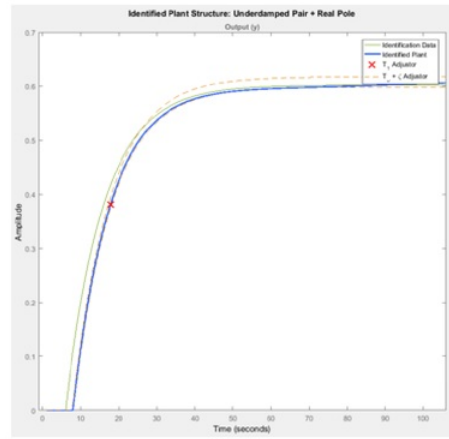


Figure 25 - Underdamped Pair & Real Pole Manual Step Response

Figure 26 shows underdamped pair & real pole auto estimate result;

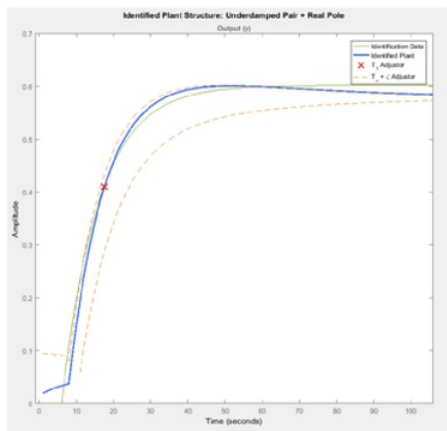


Figure 26 - Underdamped Pair & Real Pole Auto Step Response

Using the underdamped pair & real pole structure, the P, PI and PID parameters were manually adjusted in an attempt to get the response within the required specification. Unfortunately, the overshoot and response time were way over the specification and therefore not suitable;

Controller Parameters	
	Tuned
Kp	0
Ki	0.15736
Kd	0
Tf	n/a
Performance and Robustness	
	Tuned
Rise time	17.5 seconds
Settling time	59.4 seconds
Overshoot	13.6 %
Peak	1.14
Gain margin	59.6 dB @ 5.06 rad/s
Phase margin	54.6 deg @ 0.0742 rad/s
Closed-loop stability	Stable

Figure 27 - Underdamped Pair & Real Pole Controller Performance

P.I.D Control System

The structure was changed back to underdamped pair and the P, PI, and PID parameters were adjusted to achieve an overshoot of just 3.02% and a response time of 17.88 seconds (rise time + settling time);

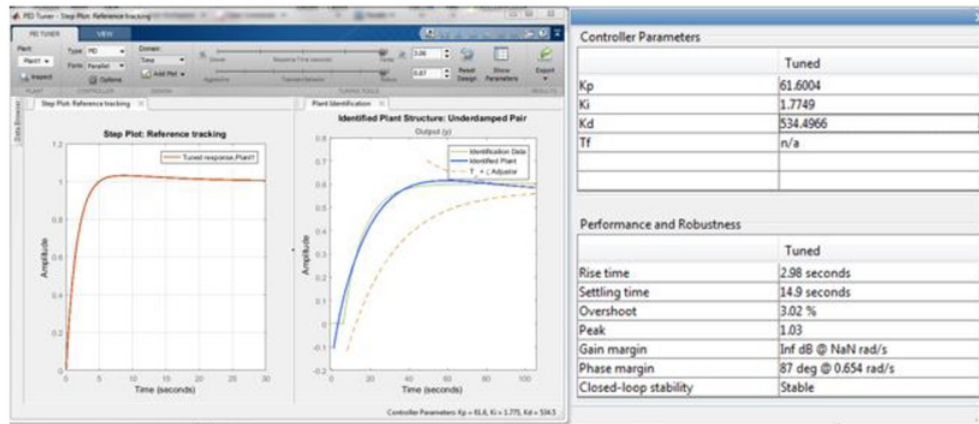


Figure 28 - Final Response Curve & Performance Data

It can be seen from the above screenshot that the final parameters for the PID controller are as follows;

- Kp = 61.6004
- Ki = 1.7749
- Kd = 534.4966

The transfer function of the final step response (underdamped pair) used is shown below;

Structure: ☒ Underdamped Pair $\frac{K}{(T_{\omega}^2 s^2 + 2\zeta T_{\omega} s + 1)}$

☐ Delay ☐ Zero ☐ Integrator

Figure 29 - Final Transfer Function

6.0 PWM vs. Analogue

The speed of a DC motor depends on its physical construction, the load it is driving and the voltage applied. A DC motor's speed can be controlled by altering the applied voltage. [1] When controlling the speed of a DC motor a varying voltage can be used which results in large amounts of power dissipation. In some situations using an analogue voltage can cause a motor to stall due to the loss in torque. Using analogue voltages to control motors is much more inefficient compared with using PWM. PWM which stands for 'pulse width modulation' is a technique where the voltage is either on or off and so gives a square wave output. [2] By using PWM to alter the speed of a motor the voltage applied can be averaged which results in a controller which is made smaller, more efficient and cheaper with less heat dissipation. [1] There are many advantages of using PWM over analogue and using PWM allows a motor to operate at its full rated voltage without losing torque and therefore overcoming the possible issue of the motor stalling. As the voltage is pulsed on or off this means when the voltage is off the system is consuming only very little power. For example if the on time is equal to the off time then the power consumption is decreased by half which is why PWM is more efficient than analogue. [2] The diagram below shows a typical PWM signal with a duty cycle of 25%;

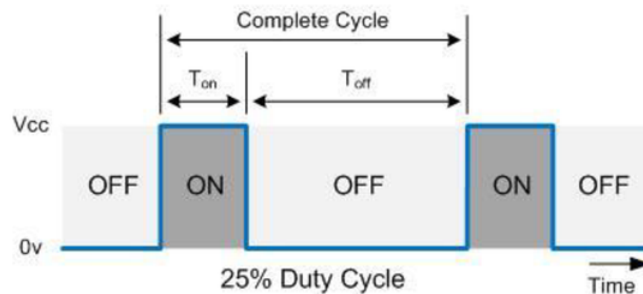


Figure 30 - [2]

The duty cycle of the PWM signal can be varied which directly varies the speed of the motor while still using the maximum rated voltage which allows for accurate speed control while maintaining torque and efficiency. The motor induction causes the current to average out while keeping the torque in proportion to the current. A disadvantage of using PWM is when a frequency is used is either too high or too low then the motor's induction cannot correctly average the current. [2]

Another advantage of using PWM is the signal remains digital from the source such as a PLC to the controlled system without the need for a digital to analog converter (DAC). Noise induced into a digital signal can only really cause problems when the noise level is large enough to change a logic-1 to a logic-0, or vice versa meaning digital signals have better noise immunity. PWM is often used for communications over a long distance, by converting an analogue signal to digital can increase the length of communication which can then be converted back to analogue by using RC or LC networks to filter the modulating high frequency square wave converting the signal back to analogue if so desired. [3] A disadvantage of using PWM for communications is the digital signal requires significantly more bandwidth than using the original analogue signal.

Output signals in a PLC are generated by using either solid state switches such as transistors or TRIACs or an electromechanical relay which can easily be used by switching on/off to create PWM.

7.0 Operator Flow Chart

The below flowchart is intended to be given to the operator so they know how to operate the system;

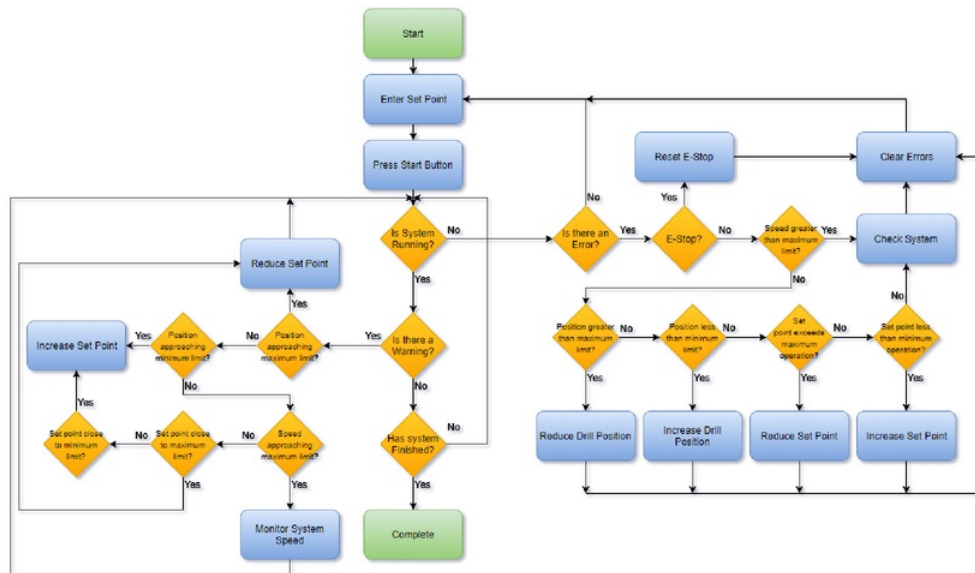


Figure 31 - Operator Flow Chart

8.0 Conclusion

This report has demonstrated the design of a PLC based PID control system to control a servo driven linear positioning system. The system incorporates a HMI display which provides visual indicators of the systems status and allows the operator to enter set points to control the position of the drill. A set of warning and error codes were implemented into the program to ensure safe operation. A higher level access was implemented which allows persons of high authority to set the minimum and maximum values to safeguard the operator and surrounding persons. Tables were included describing the error and warning codes to the operator as well as a flowchart to help aid in the operation and diagnostics of the system. MATLAB was used to create a transfer function so that the system operates within the specified specifications. Although this was more difficult to achieve than expected it can be seen from the graphs that a fairly good response curve was achieved in the end. A brief literature review was carried out detailing the advantages and disadvantages of using PWM over analogue and it can clearly be seen from the review that PWM has many more advantages. Overall this was a very successful project and the system has been fully tested and proven to be working correctly within the required specifications. However, if this project were to be carried out again perhaps more errors and warnings could be implemented such as warnings that can be generated from the PID controller such as overshoot limits etc.

Sam O'Leary- PLC Assignment 1

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

Instructor

68/100

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

MICHAEL SAM O'LEARY**68****RESEARCH-INFORMED LITERATURE**

The literature review is clear with good background information but would have been improved with a few more simple illustrations, along with an advantage/disadvantage table. 65%

9.75**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 but a little cluttered. Warning and error screens are nicely laid out and make it easy to identify faults/warnings and allow for user navigation. Good fault audit table. Limited improvements.65%

13

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.65%

9.75