IMPLEMENTATION AND BUILDING OF CHEAPER SIGNAL CONDITIONERS & A PID CONTROLLER BLOCK USING A PLC

by

ARCHIFORD NDHLOVU R161631A

Submitted in partial fulfilment for the degree of

BSc Honours in applied physics and instrumentation & control

Department of Telecommunications in the Faculty of Science and Technology at the

Midlands State University

Gweru, Zimbabwe

November 2018

Supervisor: Mr G Manjengwa

RELEASE FORM

NAME OF AUTHOR: NDHLOVU ARCHIFORD

PROJECT TITLE: IMPLEMENTATION AND BUILDING CHEAPER SIGNAL CONDITIONERS & PID CONTROLLER BLOCKS USING A PLC

PROGRAMME FOR WHICH

WAS PRESENTED: Bachelor of Science in Applied Physics and Instrumentation

YEAR GRANTED: 2018

SIGNED: **……**

ADDRESS : A14 Glencairn Flats ,Bulawayo, Zimbabwe

DATE: DEC 2018

APPROVAL FORM

The undersigned certify that they have read and recommend to the Midlands State University for acceptance; a dissertation entitled, Automated Contact Thermometer Calibration System.

Submitted by ARCHIFORD NDHLOVU in partial fulfilment of the requirements of the Bachelor of Science in Applied Physics and Instrumentation Honours Degree.

…… ……

SUPERVISOR DATE

…… ..……

PROGRAMME/ SUBJECT COORDINATOR DATE

……. ……

EXTERNAL EXAMINER DATE

Abstract

The proportional, integral and derivative (PID) technique is a reliable and widely used art to control and regulate complex processes. The PID block and signal conditioners are constructed by hand inside a small memory Schneider PLC without inbuilt process control loops using software. Analogue input module of the PLC receives the analogue unconditioned signal from the rtd/thermocouple sensor, passes it through the signal conditioning phase to the pid block. The PID block executes the control logic with two degrees of freedom and then sends it to the output signal conditioning block then output PLC output module. Signal conditioning is done between the input module interface & the PID block module and also between the PID block and the output module. The processed output signal shall be sent to the final control element with the use of the PWM technique. Precise temperature control is difficult to achieve using simple inexpensive techniques, it can easily be achieved by using highly sophisticated expensive PLCs that come with inbuilt process control loops. This project aims to show that with great ingenuity, cheaper PLCs without inbuilt process control loops can be equally used to precisely control complex processes by manually building PID control blocks and signal conditioners.

DEDICATION

TO

Praiseworth, Sasha, Ethan

DECLARATION

I NDHLOVU ARCHIFORD, hereby declare that I am the sole author of this thesis, I authorise the Midlands State University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

Signature …… Date …….

ACKNOWLEDGEMENTS

I would like to sincerely thank the following personnel for their contribution towards the completion of this study;

- Mr G Manjengwa for the guidance and support.
- Eng. A. Bhila for availing the modicon M340 PLC for the project.
- The Midlands State University for providing the necessary platform and resources to be able to undertake the project.
- My beautiful loving wife Praiseworth Muchibwa, my beautiful kids Ethan & Sasha Ndhlovu, family and friends for their financial, physical and emotional support towards completion of this study.

TABLE OF CONTENTS

3 METHODS AND TECHNIQUES

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

CPU	CENTRAL PROCESSING UNIT
PLC	PROGRAMMABLE LOGIC CONTROLLER
PID	PROPORTIONAL INTEGRAL DERIVATIVE CONTROLLER
SP	SETPOINT
PV	PROCESS VARIABLE
MV	MANIPULATED VARIABLE
D_{min}	DEVICE MINIMUM
D_{max}	DEVICE MAXIMUM
SMP	NUMBER OF SAMPLES
E_Max	ENGINNERING MAXIMUM
E_Min	ENGINEERING MINIMUM
Mm_SP	SCADA SETPOINT
Mm_MV	SCADA MANIPULATED VARIABLE
FRA	FORWARD REVERSE ACTION
PWMF	PULSEWIDTH MODULATION FORWARD
PWMR	PULSEWIDTH MODULATION REVERSE
OFFTM	OFF TIME
ONTM	ON TIME
CTim	PULSEWIDTH MODULATION CYCLE TIME
AM	AUTO/MANUAL

CHAPTER 1

INTRODUCTION

1.1 Background

Most PLCs come with an already inbuilt customised Process control loop blocks (PID) to control specific processes which in turn makes such PLCs very expensive to acquire for a starting or developing low capital firm. These high costs of acquiring such PLCs makes it impossible for smaller firms to operate with accurate and reliable technology, hence the technique of building such PID blocks by hand cuts significantly on the costs making it possible for basic & cheapest PLCs on the market to control complex processes. A lot of small indigenous entrepreneurial firms have lost on projects due to the failure of acquiring such expensive Plc. This approach can save companies thousands of dollars if not millions if implemented well. The diagrams below clearly show the difference in PLC sizes and their functions.

GENERAL CHARACTERSITICS OF THE M340 PROCESSORS.

Selection guide

Modicon® M340™ automation platform
Modicon M340 processors

1

Modicon® M340™ platform for Unity Pro™ software offer		BMX 34 10 Standard processor	BMX 34 20 Performance processors
			
Racks	Number of racks	2 (4, 6, 8 or 12 slots)	4 (4, 6, 8 or 12 slots)
	Max. number of slots (excluding power supply module)	24	48
Inputs/Outputs	In-rack discrete I/O *(1)*	512 channels (modules with 8, 16, 32 or 64 channels)	1024 channels (modules with 8, 16, 32 or 64 channels)
	In-rack analog I/O *(1)*	128 channels (modules with 2, 4, 6 or 8 channels)	256 channels (modules with 2, 4, 6 or 8 channels)
	Distributed I/O	Limited depending on the type of medium: Over Ethernet Modbus/TCP network via network module (63 devices with I/O Scanning function), over Modbus link (32 devices)	
In-rack application-specific channels	Max. number of channels (counter, motion control and serial link)	20	36
	Counter *(1)*	BMX EHC 0200, 60 kHz 2 channels or BMX EHC 0800, 10 kHz 8 channels modules	
	Motion control *(1)*	BMX MSP 0200, 200 kHz 2 channels with PTO outputs *"Pulse Train Output"* module for servo drives	
	Process control, programmable loops	Process control EFB library	
Integrated communication ports	Ethernet Modbus*/TCP network	–	
	CANopen Master machine and installation bus	–	

Fig. 1.1 M340 processor specifications

As can be seen in the table [1, Fig.1.1], the M340 modicon PLC CPU processor family does not come with inbuilt process control loops (PID blocks) as can be seen in the in-rack application-specific channels under the process control, programmable loops column. This rather comes with a process control EFB library which is basically a tool that allows one to be able to build defined functions and control function blocks. These blocks can be used in all Concept IECs programming languages for automation. The above modicon PLC costs an average of USD$6500.

General characteristics of the TSX P57 454 processors

TSX P57 454 processor

The following table gives the general characteristics of the TSX P57 454 processor.

Characteristics			TSX P57 454
Maximum configuration	Maximum number of TSX RKY 12EX racks		8
	Maximum number of TSX RKY 4EX/6EX/8EX racks		16
	Maximum number of slots		111
	Maximum number of simultaneous communication EF		64
Functions	Maximum number of channels	In-rack discrete I/O	2048
		In-rack analog I/O	256
		Expert	64
	Maximum number of connections	Built-in Uni-Telway (terminal port)	1
		Network (ETHWAY, Fipway, Modbus Plus)	4
		Master Fipio (built-in): No. of devices	127
		Third party field bus	4
		AS-i field bus	8
	Savable real-time clock		yes
	Process control channels		20
	Process control loops		60
Memory	Savable internal RAM		440K8
	PCMCIA memory card (maximum capacity)		2048K8
Application structure	Master task		1
	Fast task		1
	Event processing (1 has priority)		64

Fig 1.2 TSX series hybrid processor specifications

[2, Fig 1.2] shows that the TSX P57 454 CPU processor comes with inbuilt process control loops which makes it very easy for one to implement PID control logic. This PLC processor type comes at very expensive price tag of around USD$30 000 -$40 000 hence it's not cheaper for small and medium enterprises.

1.2 PROBLEM STATEMENT.

PLC manufacturers mostly use the number of inherent PLC PID controllers to size and price their PLCs.PLCs with most inherent controllers will cost more than PLCs with less or none at all. Practically it can be expressed that the effectiveness of a PID controller is dependent on the PLC scan cycle time and the PLC available memory.

The above-mentioned parameters are usually left unexploited to the maximum even after deployment of all available manufacturer developed controllers inherent in the PLC. Value for money in these scenarios is usually never realised.

1.3 SIGNIFICANCE OF STUDY.

Designing and construction of PID blocks & signal conditioners by hand will effectively solve complex automation control problems that are usually achieved by using very expensive hybrid PLC models that come with inherent inbuilt process control loops(PID blocks).the technique will effectively and significantly reduce the costs of any project that would have been done using expensive hybrid PLCs.This approach is very relevant to the technological development of Zimbabwe since the country is categorised as a developing nation. This approach is a more realistic and affordable approach to the indigenous business community of the nation. This technique can be applied in control of temperature in high pressure vessels in power plants, incubators for the hospitals and farmers, incinerators and many more processes that require precise control.

1.4 HYPOTHESIS.

The author employs the technique of PID control to precisely monitor and control temperature. The PID equation is given below as follows;

$$u(t) = K\left(e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau + T_d\frac{de(t)}{dt}\right)$$

Where u(t) is the controller output.

e is the error.

The control variable is the sum of the proportional term(P), integral term(I) and the derivative term(D). The controller parameters are proportional gain $\boldsymbol{K}$, integral time $\boldsymbol{T_i}$ and derivative time $\boldsymbol{T_d}$.[3]

This control method ensures that the error is always zero meaning that the output and process value will have the same value. The author shall input this equation in the PID block design within the PLC to dictate the control algorithm.

1.5 AIMS AND OBJECTIVES.

- Build a practical PID controller.
- Build supporting signal conditioners.
- Test the control loop and prove its effectiveness.

1.6 THESIS OUTLINE.

Chapter 1: this chapter introduces the main concept of the project and the project overview into an imaginable picture. It clarifies the problem statement, the aim & objectives, project scope and the hypothesis inorder to offer a clearer direction of the project.

Chapter 2: this chapter is all about literature review, it gives an in-depth analysis and detailed theoretical analysis of each and every instrument used in the project. The instruments include the modicon PLC model modicon M340,24V relays, rtd, water bath, heating element, HMI, temperature control unit. books, journals, articles and any other source of information are used as reference to support the project.

Chapter3: this chapter outlines the methodology used, system design, interfacing of each and every component and the data collection method and procedure.

Chapter 4: the chapter presents the collected results, the tuning techniques used and an analysis of the results. challenges, alternatives and solutions employed are clearly stated.

Chapter 5: conclusion and recommendations are clearly articulated in this chapter. Recommendations for further study are given in this chapter.

REFERENCES

[1] Schneider Electric, “Premium and Atrium Using Unity pro,”, 05/2010.

[2] Schneider Electric,” Modicon M340 Automation Platform,” 2010.

[3] Astrom, K.J., Hagglund, T., PID Controllers: Theory, Design and Tuning,2nd Edition,1995.

CHAPTER 2

LITERATURE REVIEW

2. INTRODUCTION.

The chapter gives an in-depth and detailed analysis of the study material used in this project. This chapter reviews on PID control.

2.1.1 FEEDBACK CONTROL SYSTEM

Virtually all feedback controllers determine their output by observing the error between the set point and a measurement of the process variable[5]. Errors can occur when an operator changes the set point or when a disturbance or a load on the process changes the process variable. The controller's role is to eliminate the error automatically. [7] Feedback control is actually essential to keep process variable close to the desired value in spite of disturbances and variations of process dynamics, and the development of feedback control methodologies has had a tremendous impact in different fields of engineering. [9] Negative feedback, on the other hand, works toward restoring balance. [14] Fig 2.1 shows a feedback system.

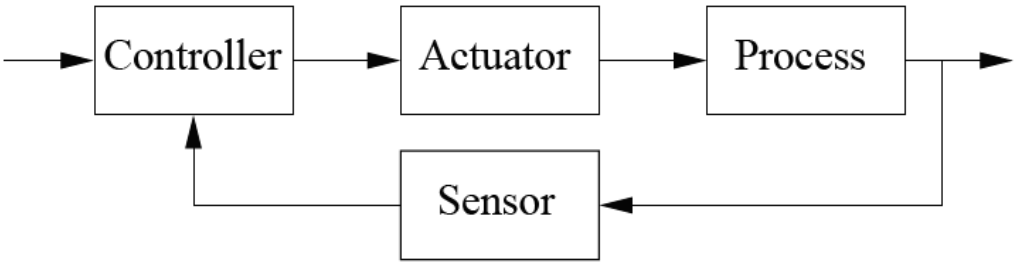


Fig 2.1 Feedback control system diagram

2.1.2 PID CONTROLLER

Undoubtedly, since 140[12,13], PID controllers are the option most frequently used in different process control applications. PID controllers are widely used in industries due to their reliability and accuracy. A PID controller employees the technique of closed loop control to calculate the error signal which is then used to derive the corrective action signal. The PID controller employs proportional (P), Integral (I) and Derivative (D) (PID) action(algorithm) to achieve accurate control action and is termed three term control action as shown in Fig 2.2.

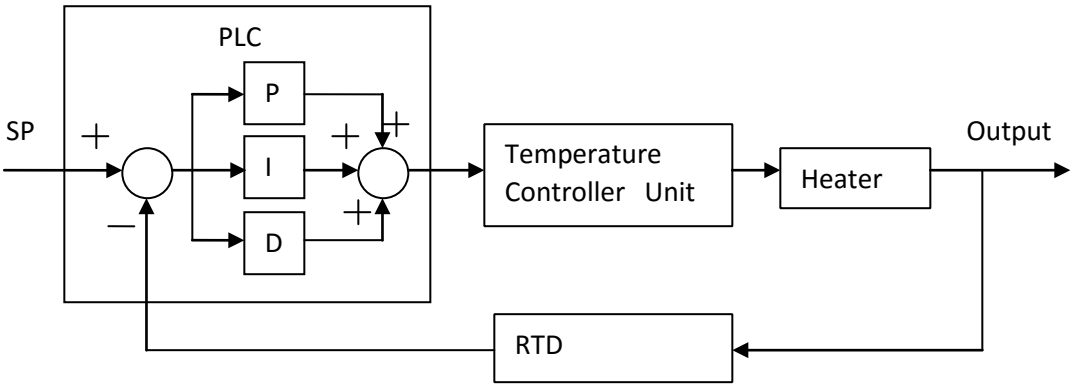


Fig 2.2 PID control loop

$$\mathrm{u(t)} = \mathrm{MV(t)} = K_p e(t) + K_i \int_0^t e(\tau)\,d\tau + K_d \frac{de}{dt}$$

Where u(t) is the PID controller output.

K_p: proportional gain constant, a tuning parameter.

K_i: integral gain constant, a tuning parameter.

K_d: derivative gain constant, a tuning parameter.

e: Error = SP – PV.

t: time or instantaneous time

MV: Manipulated variable.

Feedback loops have been controlling continuous processes since 1700's. [1].

Today, there are several more controllers, but most of all derivates from the PID controller. "The PID controller is by far the most common control algorithm. Most feedback loops are controlled by this algorithm or minor variations of it.it is implemented in many different forms, as a stand-alone controller or as part of the DDC (Direct Digital Control) package (….). Many thousands of instrument and control engineers worldwide are using such controllers in their daily work." [2]

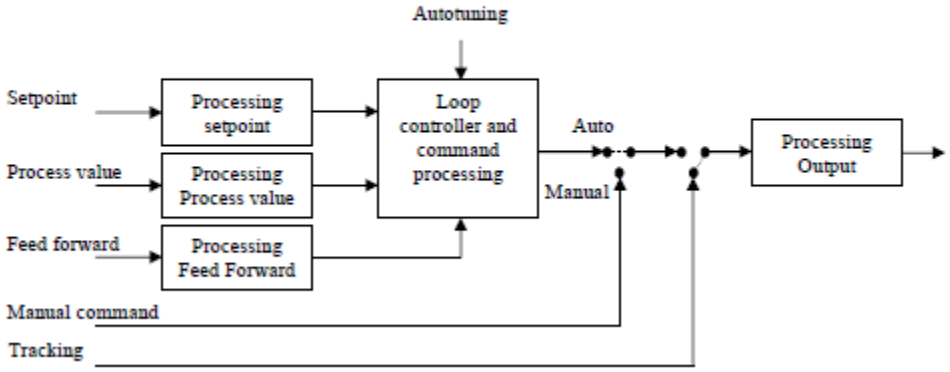


Fig 2.3 PID function block diagram

2.1.3 PROPORTIONAL

Proportional control is the type of action employed by a controller when the rate of control is proportional the size of the error signal of the process. The letter P in the PID equation represents the proportional control action. The equation below represents proportional control. The proportional output of the controller is a result of the product error (e) and proportional term (P).

$$P_{\text{out}} = K_p\, e(t)$$

Where P_{out} =the proportional output term

K_p = proportional gain

e = error (set point - process variable)

t = instantaneous time

The proportional (P) mode alone is the simplest linear control algorithm. It is characterized by a continuous relationship between the controller input and output. [22] The adjustable parameter of the proportional mode called the proportional gain. It is frequently expressed in terms of percentage proportional band, PB. The proportional controller responds only to the present.it cannot consider the past history of the error or possible future consequences of an error trend. [22]

2.1.4. INTEGRAL

The “I” in the term PID represents integral action. Integral action is used when the controller is needed to correct steady state offset from a constant reference value of a signal. This type of control over comes and solves the weakness of the proportional control by totally removing the offset without taking into consideration the controller gain.it is also called the reset action.

$$I_{\text{out}} = K_i \int_0^t e(\tau)\, d\tau$$

Where I_{out} = integral term (output)

K_i = integral gain

e = error (SP-PV)

t = time constant

the integral (I) control mode is also sometimes called reset mode because after a load change it returns the controlled variable to set point and eliminates the offset, which the plain proportional controller cannot do. The integral mode has been introduced inorder to eliminate the offset that plain proportional control must result in an offset is because it regards the past history of the error. The integral mode, on the other hand, continuously looks at the total past history of the error by continuously integrating the area under the curve and eliminates the offset by forcing the addition of energy that should have been added in the past. [3]

2.1.5 DERIVATIVE

The rate of change of the process error is calculated by determining the first derivative with respect to time and multiplying this rate of change by the derivative gain. The output control action is termed the derivative gain, $K_{D.}$

$$D_{\text{out}} = K_d \frac{de}{dt}$$

Where D_{out} =derivative term (output)

K_d =derivative gain

e =error

t =time

derivative action is used to reduce the magnitude of the overshoot and also slow down the transient response.

The purpose of derivative action is to improve the closed-loop stability. The instability mechanism can be described intuitively as follows. Because of process dynamics, it will take some time before a change in the control variable is noticeable in the process output. Thus, the control system will be late in correcting for an error. [2]

2.1.6 FEATURES AND ADVANTAGES OF PID CONTROL BLOCKS

The PID block possesses numerous functions which gives huge advantages to other control modes, these are listed below;

- Calculating the proportional, integral and differential component of its incremental form.
- Antiwind up measures.
- Displays actual value, set point and output in physical units.
- Direct or reverse action.
- Differential component to process value or deviation.
- Parametering the transfer gain of the differential component.
- Reduces overrun
- Possibility of upgrading a block external integral component.
- Feed forward component for disturbance compensation.
- Dead zone deviation.
- Incremental value and absolute value output.
- Upper and lower limit on the output signal.
- Output offset.
- Selecting manual and automatic mode.
- Tracking mode.
- Upper and lower set point limit.

- Auto tuning
- Selection of internal or external set point. [15]

2.2 PLC

2.2.1 HISTORY OF THE PLC.

A new means to modify control circuitry was needed. The development and testing ground for this new means was the U.S. auto industry. The time period was late 1960's and early 1970's and the result were the programmable logic controller or PLC. The PLC developed during this time was not very easy to program. [4]

The term PLC stands for Programmable Logic Controller. The PLC was developed and it was mainly of two types namely modular and compact. Modular consists of separate specialized modules whilst compact has inbuilt modules. The PLC mainly consists of the following components namely the CPU, Memory, Power supply, I/O modules.

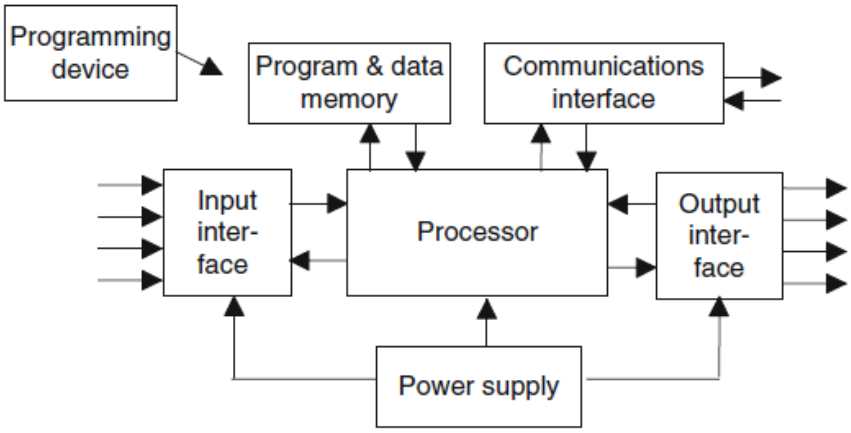


FIG 2.4 The PLC System

2.2.2 CPU

This is called the central processing unit (CPU) and it consists of system memory, microprocessor, communication interfaces for external programming devices, HMI display devices and printers. This component of the PLC has the capacity to perform complex

mathematical algorithms including complex control commands and PID control loop functions. In some cases, the CPU usually comes attached with the power supply.

The central processing unit (CPU) governs all PLC activities. The following three components constitute a CPU. [5]

- The processor
- Memory system
- System power supply. [5]

During the PLC operation, the CPU completes three processes:

1) it reads, or accepts, the input data from field devices via input interfaces.
2) It executes, or performs, the control program stored in the memory system.
3) It writes, or updates, the output devices via the output interfaces. [5]

This whole process is known as the PLC scan cycle. [5] One common way of rating how a PLC performs these tasks is its scan time. [10]

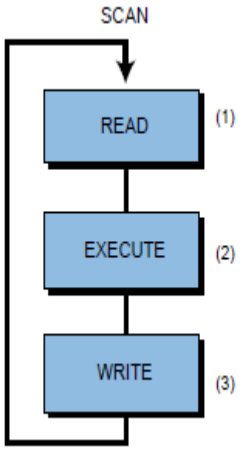


Fig 2.5 PLC scan cycle

2.2.3 INPUT & OUTPUT MODULES

The most common type of I/O interface module is its discrete type. This type of interface connects field input devices of the ON/OFF nature such as selector switches, pushbuttons and limit switches. Likewise, the output control is limited to devices such as lights, relays, solenoids and motor starters that require simple ON/OFF switching. The classification of discrete I/O covers bit-oriented inputs and outputs.in this type of input or output; each bit represents a complete information element in itself and provides the status of some external contact or advises of the presence or absence of power in a process circuit. [7]

The input/output unit provide isolation and signal conditioning functions so that sensors and actuators can often directly be connected to them without need for the other circuitry. Electrical isolation from the external world is usually means of opts isolators. [23]

2.2.4 POWER SUPPLY

The power supply unit is needed to convert the mains AC voltage to low DC voltage (5V) necessary for the processor and the circuits in the input and output interface modules. [23] This is the component of the PLC that provides enough power to the various components of the PLC. The specification of the power supply should always be such that the power supply should be adequate enough to provide enough operational power to the rest of the PLC components even when external devices like programming machines are connected without failure.

2.2.5. MEMORY

The total memory system in a PLC is actually composed of two different memories [5];

- The executive memory
- Application memory.

The executive memory is a collection of permanently stored programs that are considered part of the PLC itself. These supervisory programs direct all systems activities, such as execution of control program and communication with peripheral devices. The executive section is the part of PLC's memory where the systems available instruction software is stored.[5] This area of memory is not accessible to the user. The application memory provides storage for the user programmed instructions that form the application program. The application memory area is composed of several areas, each having a specific function and usage. [5]

Two types namely RAM and ROM are mainly found as subsections of the application and executive memory. RAM memory is the memory stored alive in the CPU through the use of batteries or power supply. This is where the main program for running the PLC instructions is kept, although newer PLCs now come with EEPROM which doesn't require batteries for storage. Battery type used for such purposes are nickel-cadmium type which store charge for a very long time. ROM memory contains the running program information and makes it possible for the CPU to execute and act on the ladder program instructions stored in RAM memory.

2.2.6 ADVANTAGES OF PLCS OVER RELAY LOGIC.

The advantages of PLCs over relay logic are listed below;

- Flexibility and ease of service - changing features to relay logic requires change in wiring but with PLCs it's only a change in software program.
- Cost-relatively cheaper on big projects compared to hardwiring.
- Space advantages-PLCs are small and can be the size of a house brick whilst relays will require huge panels.
- Communication and interoperability-Scada packages can be easily linked and communication integration with various communication protocols is made easy. [16]

2.3.0 SIGNAL CONDITIONING

Signal conditioning is a basic component of all measurement devices.it converts incoming measurements into a form acceptable to digitization hardware. [17] We need signal conditioning to ensure accuracy, transmission and signal type are compatible with the rest of the control system [18] Some of the conditioning features are listed below;

- Transducer excitation
- Isolation
- Cold-junction compensation
- Filtering
- Amplification/attenuation
- Linearization
- Multiplexing
- Bridge completion
- Shunt calibration
- Switching relays
- Offsetting
- Signal Conversion
- Scaling

But for the sake of this study I will further explain the techniques specifically employed for this purpose.

2.3.1 ISOLATION

Isolation is a technique used to separate physical field analogue signals from the digital equipment as they might not be compatible or might cause damage to the equipment. The signal is however converted into a form the digital equipment understands.

2.3.2 FILTERING

This is the process that blocks unwanted signal frequencies arising from external noise sources. [17] This also helps to prevent anti-aliasing.

2.3.3 LINEARIZATION

This is the process of mapping a relationship between the sensor's signal value and the physical quantity that is being measured such the incremental changes that occur in the physical quantity corresponds exactly to the signal change. [17]

2.3.4 SIGNAL CONVERSION

This is a process of changing an analogue signal from one electrical form to another, which allows equipment with dissimilar signal to communicate and accept equivalent signals for processing.

2.3.5 OFFSETTING

Offsetting is the removal of any fixed bias from the signal allowing any remaining signal to be processed. [19]

2.3.6 SIGNAL SCALING

If the amplitude of any internal signal in a fixed-point implementation is allowed to exceed the dynamic range, overflow will occur and the output signal will be severely distorted. Therefore, for optimum filter performance suitable signal scaling must be employed to adjust the various signal levels. [20]

$$\text{OUT} = (\text{IN} - \text{in\_max}) * \frac{(out\_\max - out\_\min)}{in\_max - in\_min} + \text{out\_min} \quad [21]$$

2.3.7 PULSEWIDTH MODULATION

This is a process of converting analogue control signals into digital pulse control signals.

OUT_POS is given by T_on = period * (IN/in_max) and

OUT_NEG is given by T_on = period * ([IN]/in_max) [21]

2.4. ZIEGLER NICOLS PID TUNING METHODS

2.4.1 THE STEP RESONSE METHOD

The first design method presented by Ziegler and Nichols is based on a registration of the open-loop step response of the system, which is characterised by two parameters. The parameters are determined from a unit step response of the process. The point where the slope of the step response has its maximum is first determined and the tangent at this point is drawn. The intersections between the tangent at this point is drawn. The intersections between the tangent and the coordinate axes give the parameters K and L. [6]

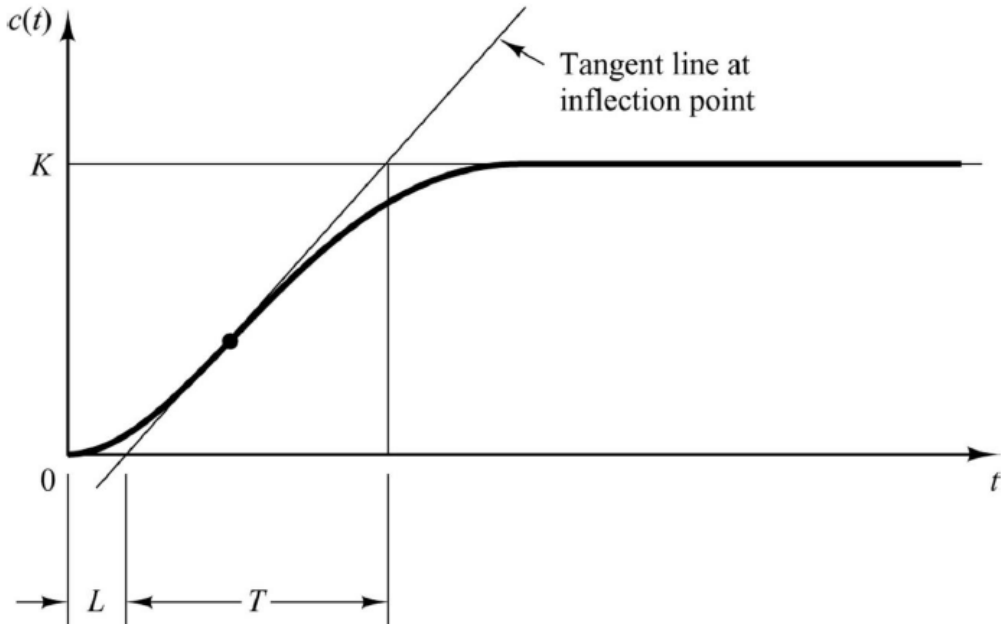


Fig 2.6 Ziegler Step response graph

For a function,

$$G_S = \frac{1}{(S+1)3}$$

Measurements at step response give the parameters K= 0,218 and L = 0.806.the controller parameters can then be determined using the table below. [6]

Table 2.1 Ziegler Nichols Tuning for Step Response method

CONTROLLER	K	T_i	T_d	T_p
P	1/a			4L
PI	0.9/a	3L		5.7L
PID	1.2/a	2L	L/2	3.4L

2.4.2 THE FREQUENCY RESPONSE METHOD

This method is also based on a simple characterization of the process dynamics. The design is based on knowledge of the point on the Nyquist curve of the process transfer function where the Nyquist curve intersects the real axis. [6] This method is used in plants where they the plant or equipment can become unstable under proportional control.

The steps for tuning PID controllers via this method are listed below;

- Reduce derivative and integrator gains to zero.
- Increase K_p from a low value, it may vary depending on the system, to some critical value $K_p=K_{cr}$ as oscillations occur. If it does not occur, then another method has to be applied.
- K_r and the corresponding period of sustained oscillation, $P_{cr.}$

Table 2.1 Ziegler Nichols Tuning for frequency Response method

PID Type	K_p	T_i	T_d
P	$0.5K_{cr}$	∞	0
PI	$0.45K_{cr}$	$\frac{P_{cr}}{1.2}$	0
PID	$0.6K_{cr}$	$\frac{P_{cr}}{2}$	$\frac{P_{cr}}{8}$

2.5 UNITY PRO XL

EcoStructure Control Expert (Unity Pro XL) is a unique software Platform to increase design productivity and performance of Modicon M340, M580, Momentum, Premium and Quantum. [11] This is a design and programming software environment offered by Schneider electric, it is a unique and high performance multi task software. Unity pro offers IEC61131 programming language standards, where each section of plc program can be programmed in any language of the programmer's choice. The unity (EFB)Elementary Function Block tool kit, enables one to develop new basic functions to in C-language to supplement the Unity PRO library. The main benefit of UNITY PRO is the complete set of functions and tools enabling modelling of application structure on any machine or process.

REFERENCES

[1] Gene F Franklin, J Davied Powell, Abbas Emami-Naeini," Feedback Control of Dynamic Systems",6th edition,2010,

[2] Astrom, K.J., E Hagglund, T., PID Controllers: Theory, Design and Tuning, 2nd edition,1995

[3] Astrom, K. J.," Control System Design",2002.

[4] John R. Hackworth, Frederick D. Hackworth, "The programmable Logic Controller. "in PLC Programming Methods and Applications: Prentice Hall.

[5] L. A Bryan, E.A. Bryan, "Programmable Controllers, Theory and Implementation",2nd Edition.

[6] Astrom, K.J, J., Hagglund, T., "PID Controllers: Theory, Design and Tuning", 2nd edition,2005.

[7] Frank D. Petruzella, "Programmable Logic Controllers",4th Edition: McGraw-Hill.

[8] W. Bolton, "Programmable Logic Controllers",5th Edition, Oxford.

[9] Antonio Visioli, Practical PID Control: Springer,2006.

[10] Dilip Patel, Introduction Practical PLC (Programmable Logic Controller) Programming: German National Library,2017.

[11] Schneider Electric, EcoStructuxure-IEC Programming Software for Modicon PAC[online].Available at https://www.schneider-electric.com/en/product-range/548-ecostruxure%E2%84%A2-control-expert/ .[Accessed 1 December 2018] .

[12] Babb, Pneumatic instruments gave birth to automatic control. Control Eng,1990.

[13] Bennet, S, The Past of PID controllers. In IFAC Digital Control: Past, Present and Future of PID Control, Terrasa,2000.

[14] F.G. Shinskey, "Process Control Systems, Application Design Adjustment"2nd Edition McGraw-Hill,1979.

[15] Schneider Electric," Unity Pro Control Block Library," April 2015.

[16] Metroid Electrical Engineering, Control Methods vs Relay Logic [online] .Available at https://www.metroid.net.au/engineering/control-methods-plc-vs-relay-logic/.[accessed 1December 2018].

[17] Control Engineering, Basics Of Signal Conditioning[online] Available at https://www.controleng.com/articles/basics-of-signal-conditioning-2/[accessed 2December 2018]

[18] ISA, Automation Basics [online] Available at https://www.isa.org/link/SigConSolve/ [accessed 2 December 2018]

[19] John Errington's Data Conversion Website, Signal Conditioning [online] Available at http://www.skillbank.co.uk/SignalConversion/conditioning.htm [accessed 2 December 2018]

[20] Andreas Antoniou, Digital Signal Processing: McGraw-Hill,2006.

[21] Schneider Electric," Unity Pro Control Block Library", June 2005.

[22] Hordeski, M.F.," HVAC Control In The New Millennium", The Fairmont Press,2001.

[23] W.Bolton."PLC Systems", Elsevier BV,2004.

CHAPTER 3

METHODS AND TECHNIQUES

3.1 INTRODUCTION

An accurate PID temperature-controlled process needs to employ various techniques to ensure accurate signal conditioning and PID output control. Techniques such as A/D conversion, signal scaling, signal filtering, signal offsetting, D/A conversion, signal transmission & display and good PID control logic will ensure desired accurate control. These are cheap and feasible techniques that any engineer or technician can employ to control engineering practice for over seven decades and have been suggested as the second most important decision and control instrument of the 20th Century [1].

3.2 OVERVIEW OF THE METHODOLOGY

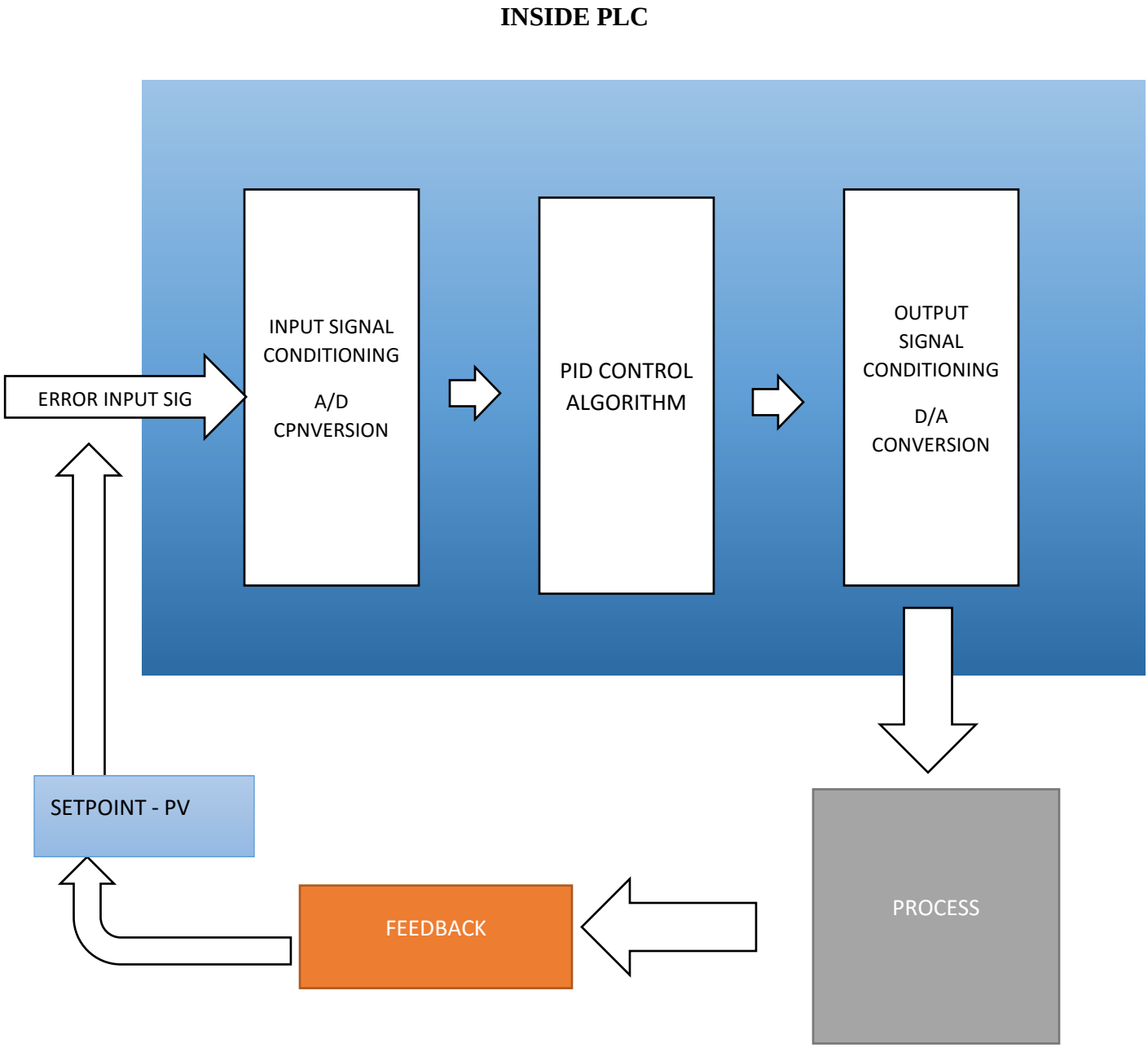


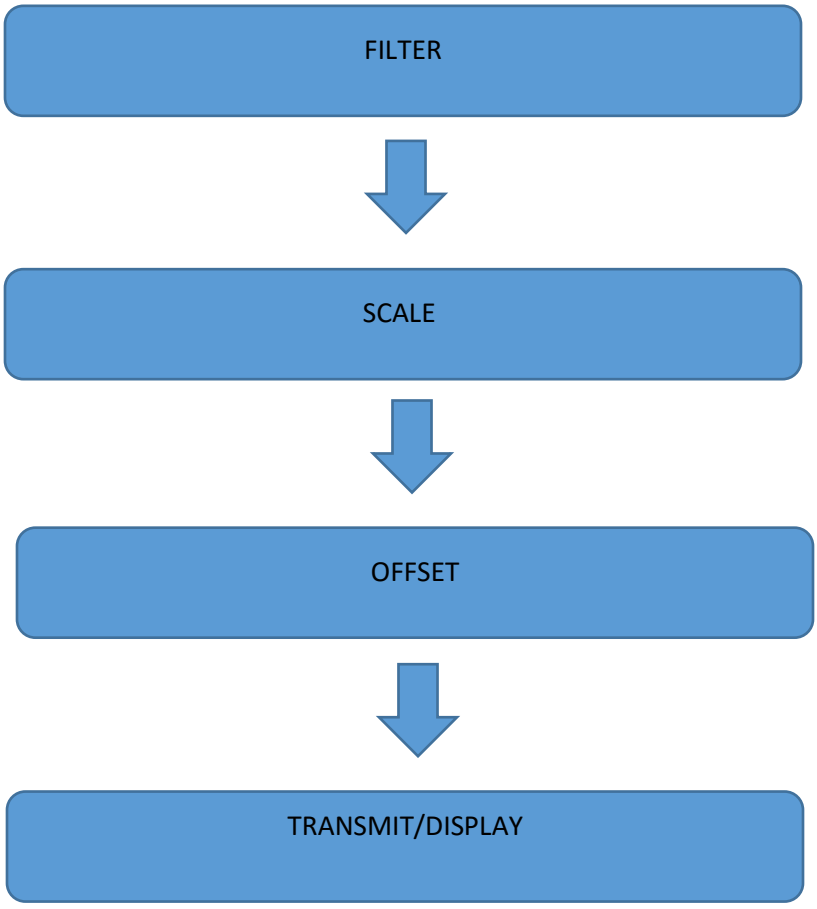
Fig 3.1 Overview of methodology

3.3 PLC INPUT CARD

The PLC input card takes a 4-20mA input signal from the field temperature measuring instrument, analogue to digital conversion is done. The signal is scaled within a digital range of the set device minimum D_{min} and device maximum D_{max} of 0-10 000.

3.4 INPUT SIGNAL CONDITIONING TECHNIQUE AND DESIGN (X _SIG)

X_SIG BLOCK FLOW CHART



The digital signal is accepted from the PLC input card, it goes through filtering, scaling, offsetting and finally retransmission to the PID configuration block. It should be noted that this is a linear feedback circuit. An averaging filter is used to perform the filtering. On the scaling of

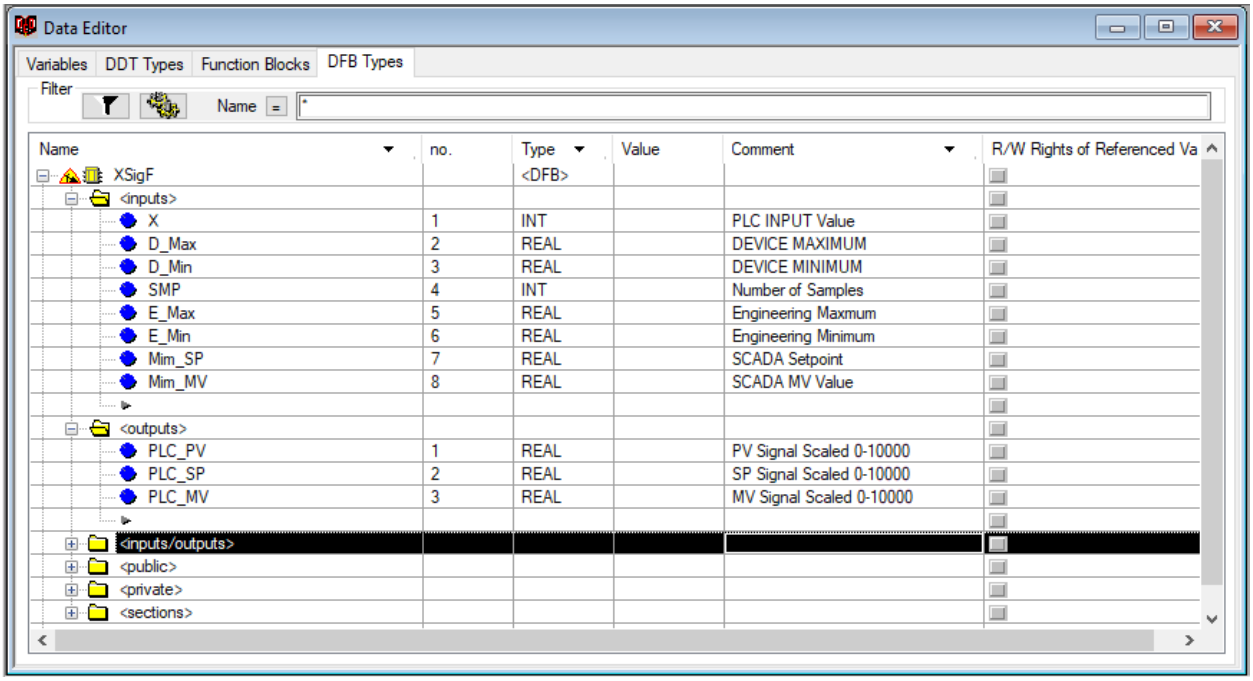
the signal it has to be noted that a linear relationship is used hence we employ the equation below;

$$y = mx + c$$

Where m is the gradient

c is the offset

the signal is then transmitted to the PID block and a standard range of real numbers between 0 and 10 000 is used to represent a scale of 0 - 100% of the signal. It should be noted that a range of 0 to 10 000 is chosen so as to increase the resolution of the signal. Any range can be used. The other output signal is sent to the Scada unit for display.



Name	no.	Type	Value	Comment	R/W Rights of Referenced Va
XSigF		<DFB>			
<inputs>					
X	1	INT		PLC INPUT Value	
D_Max	2	REAL		DEVICE MAXIMUM	
D_Min	3	REAL		DEVICE MINIMUM	
SMP	4	INT		Number of Samples	
E_Max	5	REAL		Engineering Maxmum	
E_Min	6	REAL		Engineering Minimum	
Mim_SP	7	REAL		SCADA Setpoint	
Mim_MV	8	REAL		SCADA MV Value	
<outputs>					
PLC_PV	1	REAL		PV Signal Scaled 0-10000	
PLC_SP	2	REAL		SP Signal Scaled 0-10000	
PLC_MV	3	REAL		MV Signal Scaled 0-10000	
<inputs/outputs>					
<public>					
<private>					
<sections>					

Fig 3.2 Declared input and output parameters 1for X_SIG

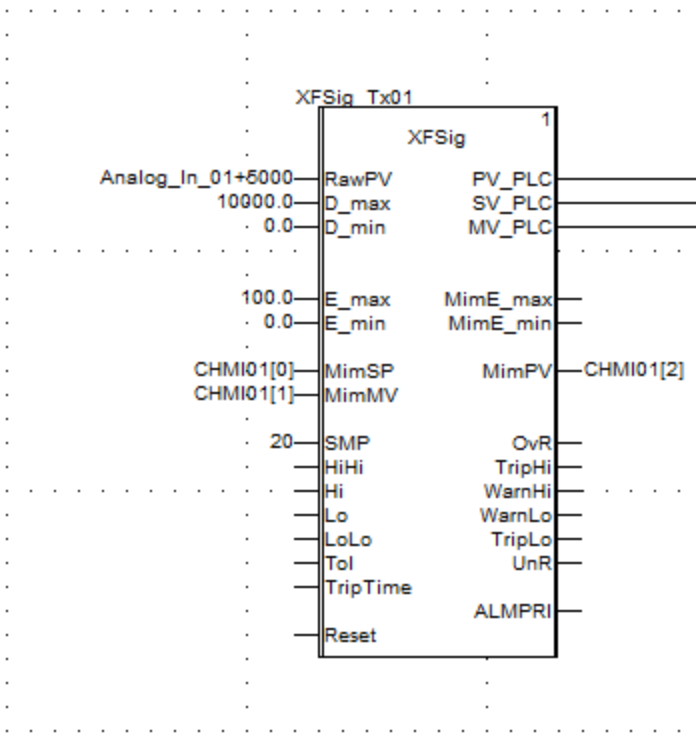


Fig 3.3 X_SIG Function block

The inputs to the X_SIG block are D_max, D_Min, SMP, E_Max, E_Min, Mm_SP, Mm_MV. The. The X_SIG block outputs three signals to the PLC PID block with the ranges of 0 – 10 000 namely process variable (PV), signal set point (SP) and manipulated variable (MV).

3.4.1 LINEARISATION AND OFFSETTING

```
Dpoint:=SAVG;
Dgrad:=((E_max-E_min)/(D_max-D_min));
DCmin:=(E_min-(((E_max-E_min)/(D_min-D_max))* D_min));
DCmax:=(E_max-(((E_max-E_min)/(D_min-D_max))* D_max));
DCres:=(E_min-(((E_max-E_min)/(D_min-D_max))* D_min));

MimPV:=dgrad*dpoint+DCmin;
```

Fig 3.4 Linearization Equation

The general equation of a line is implemented in the form of

$$Y = mx + c,$$

Where y is represented by MimPV

m is represented by dgrad

x is represented by dpoint

c is represented by DCmin

this equation of a line relationship is established so as to determine the offset of the signal and nature of behaviour of the signal.

3.4.2 SIGNAL SAMPLING

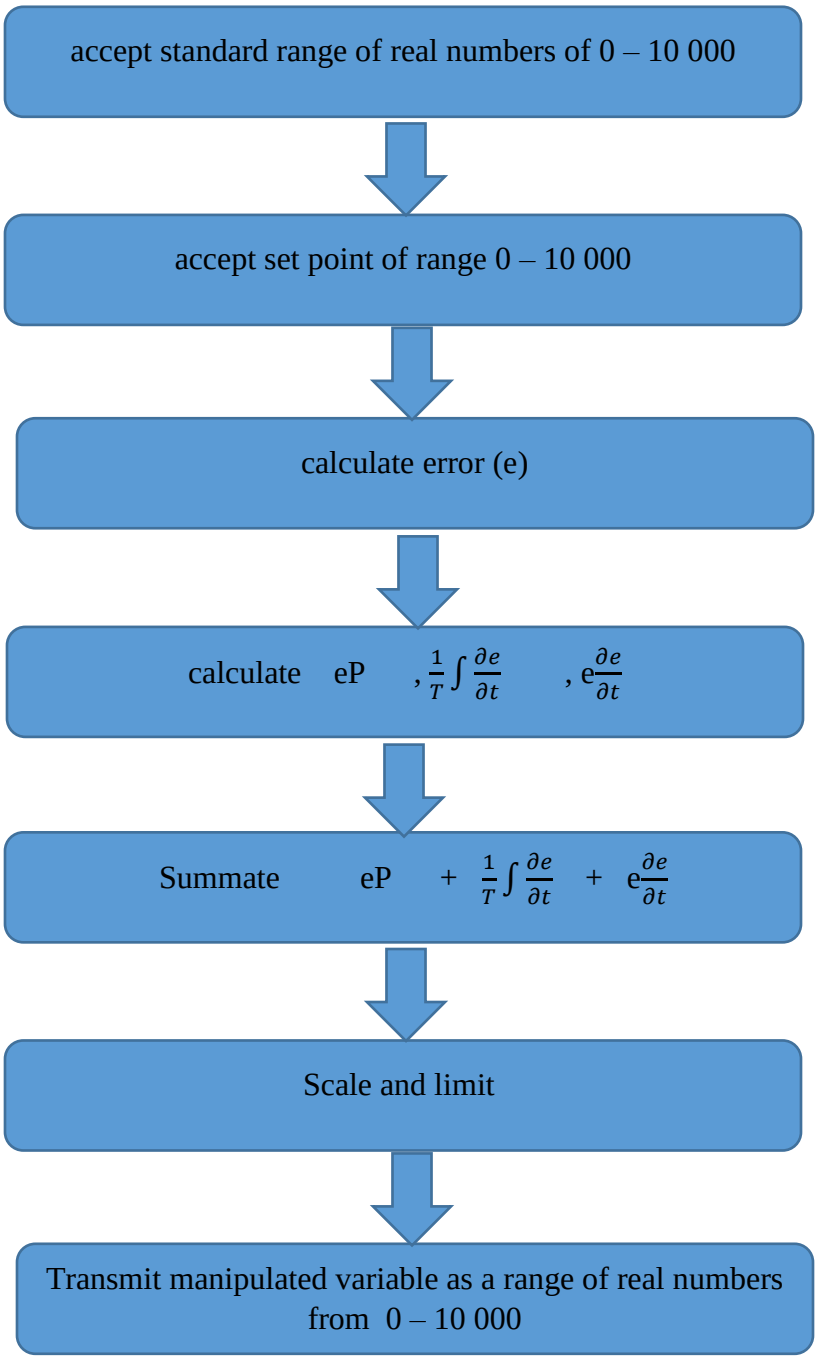
```
(* Loading the Xsignal ALgorithm into the Functional Block *)
(* Calculating Mimic and Display Value *)

T:=0;
IF SMP < 1 THEN SMPL:=1; ELSE SMPL:=SMP; END_IF;
IF SMP >= 200 THEN SMPL:=200; END_IF;

N:=SMPL;
STOT:=0.0;

FOR T:=0 TO SMPL DO
        SDAT[(N-T)]:=SDAT[((N-1)-(T))];
        STOT:=SDAT[(N-T)]+STOT;
END_FOR;

SDAT[0]:=INT_TO_REAL(RawPV);
SAVG:=STOT/(INT_TO_REAL(SMP)+1.0);
```

Fig 3.5 Sampling equation

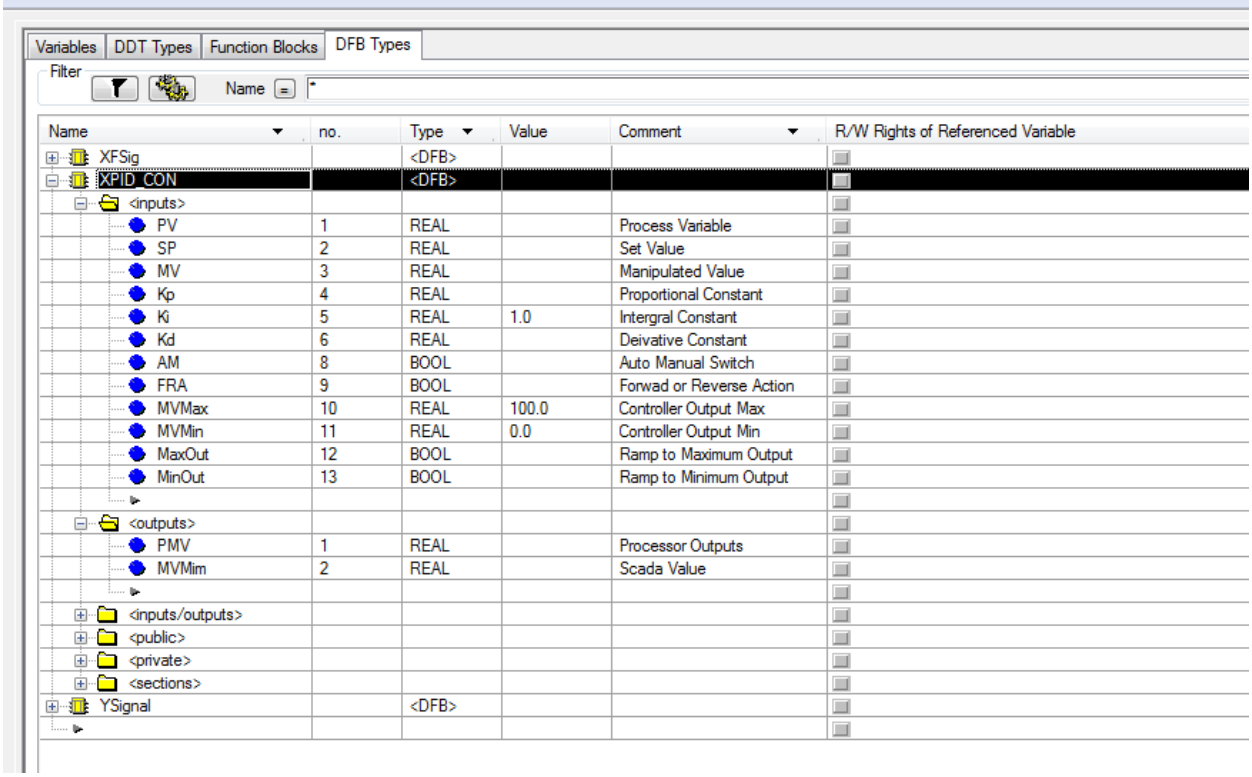
Signal sampling being a vital part of analogue to digital conversion, it is set at a slightly higher frequency to avoid aliasing of the signal. A number of 10 samples per unit time is chosen so as to avoid overloading of the PLC CPU and scan time.

3.5 PID CONTROL TECHNIQUE AND DESIGN(PID_CON)

PID_CON BLOCK FLOW DIAGRAM



This PID_CON block takes inputs MV, SP, K_p, K_i, K_d, AM, FRA, MVMax, MVMim, MaxOut, MinOut. All these inputs are vital and are used in the control algorithm stored inside the block to achieve the required PID output.



Name	no.	Type	Value	Comment	R/W Rights of Referenced Variable
XFSig		<DFB>			
XPID_CON		<DFB>			
<inputs>					
PV	1	REAL		Process Variable	
SP	2	REAL		Set Value	
MV	3	REAL		Manipulated Value	
Kp	4	REAL		Proportional Constant	
Ki	5	REAL	1.0	Intergral Constant	
Kd	6	REAL		Deivative Constant	
AM	8	BOOL		Auto Manual Switch	
FRA	9	BOOL		Forwad or Reverse Action	
MVMax	10	REAL	100.0	Controller Output Max	
MVMin	11	REAL	0.0	Controller Output Min	
MaxOut	12	BOOL		Ramp to Maximum Output	
MinOut	13	BOOL		Ramp to Minimum Output	
<outputs>					
PMV	1	REAL		Processor Outputs	
MVMim	2	REAL		Scada Value	
<inputs/outputs>					
<public>					
<private>					
<sections>					
YSignal		<DFB>			

Fig 3.6 Declared input/output parameters for PID_Block

The PID_CON block then outputs PMV and the MVMin, the PMV is the processor output and the MVMin is the Scada value taken for display.

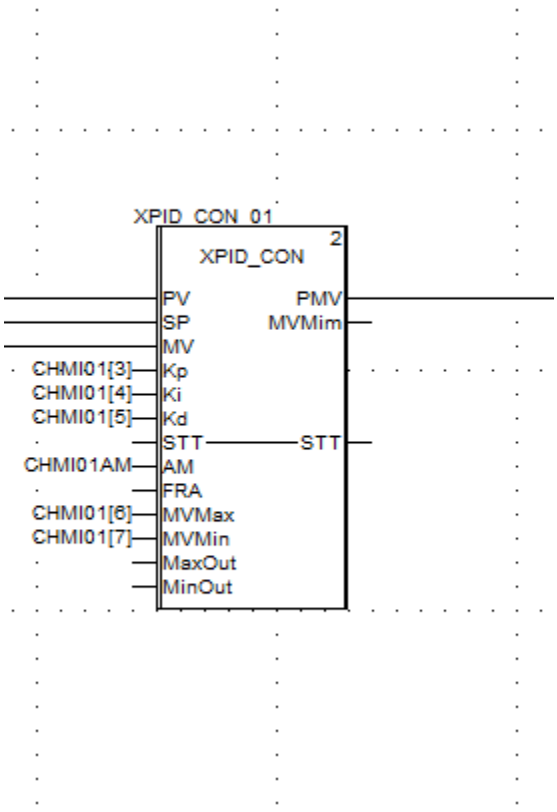


Fig 3.7 PID function block

3.5.1 PID BLOCK DETERMINATION OF ERROR FOR FORWARD OR REVERSE OPERATION

Fig 3.8 Forward reverse logic

After the PID block receives the input signal from PV, it then computes the error signal to determine whether it is positive or negative so as to be able to determine the next control action. If the error is negative it prompts for the corrective action to be forward whereas if the error is positive, it prompts the control logic to be reverse. The logical equation is used is shown below

$$e = SP - \ PV$$

Where e is the error [4]

SP is the set point

PV is the process variable [4]

3.5.2 PID BLOCK DETERMINATION OF SCAN CYCLE

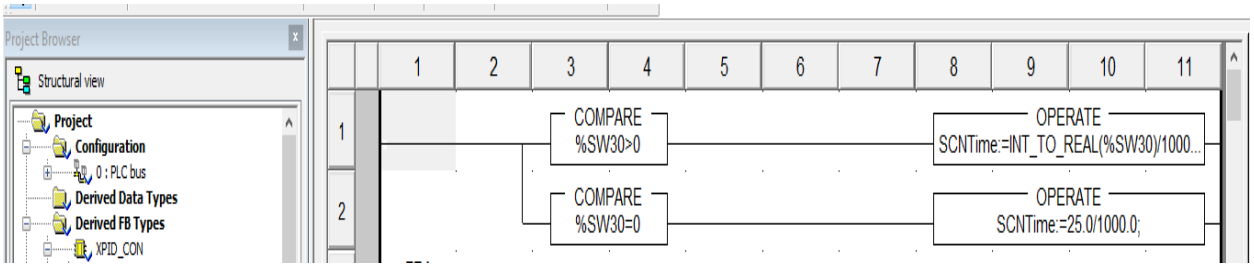


Fig 3.9 PID scan cycle

During this part of programming, a scan cycle of 30% dead band time is set so as to allow optimum update of the PLC inputs and outputs thus increasing efficiency. During this process, an instruction to convert integer numbers to real numbers is ushered inorder to allow use of fractions in the logic.

3.5.3 PID BLOCK BUMPLESS TRANSFER

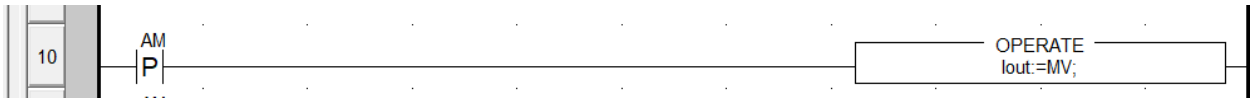


Fig 3.10 PID bumpless Transfer

This operation where the integral component is equated to the manipulated variable helps to disable a bumpless transfer. When the integral component ($_{Iout)}$ is matched with the manipulated variable (MV), this technique makes sure that no changes will occur in the output when the system switches over from auto to manual and vice versa.

3.5.4 PID BLOCK DETERMINATION OF PROPORTIONAL COMPONENT (P_{OUT}) AND INTEGRAL COMPONENT (I_{OUT}).

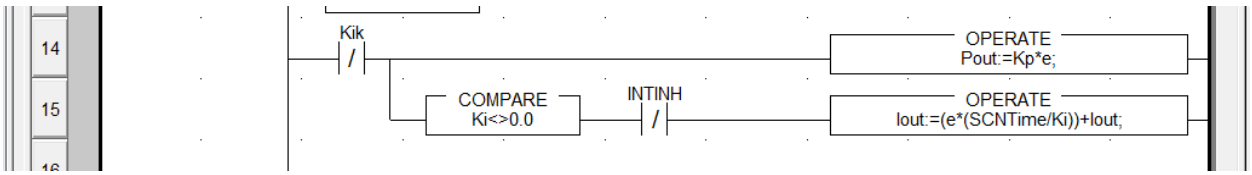


Fig 3.11 P&I determination

At this stage the (Pout) proportional output is computed as a product of the set proportional constant and the error.

$$Pout = Kp * e$$

Where Pout is proportional output

Kp is the proportional constant

e is the error

And the integral output is also initialised to give the integral output (Iout).

$$Iout = \left(e * \left(\frac{scantime}{Ki}\right)\right) + Iout$$

Where I_{out} is the integral output

e is the error

K_i is the integral constant

3.5.5 PID DETERMINATION OF THE DERIVATIVE COMPONENT

Fig 3.12 Determination of derivative constant

The derivative constant of the PID controller is given by the logical equation as shown in the final "OPERATE" block. The equation is given below,

$$D_{out} = K_d * (ST1-ST2)$$

Where D_{out} is the derivative component

K_d is the derivative constant

(ST1 – ST2) is the error

3.5.6 PID BLOCK DISABLE INTEGRAL WINDUP BY CLIPPING I_{OUT},D_{OUT} AND P_{OUT} TO 10 000.

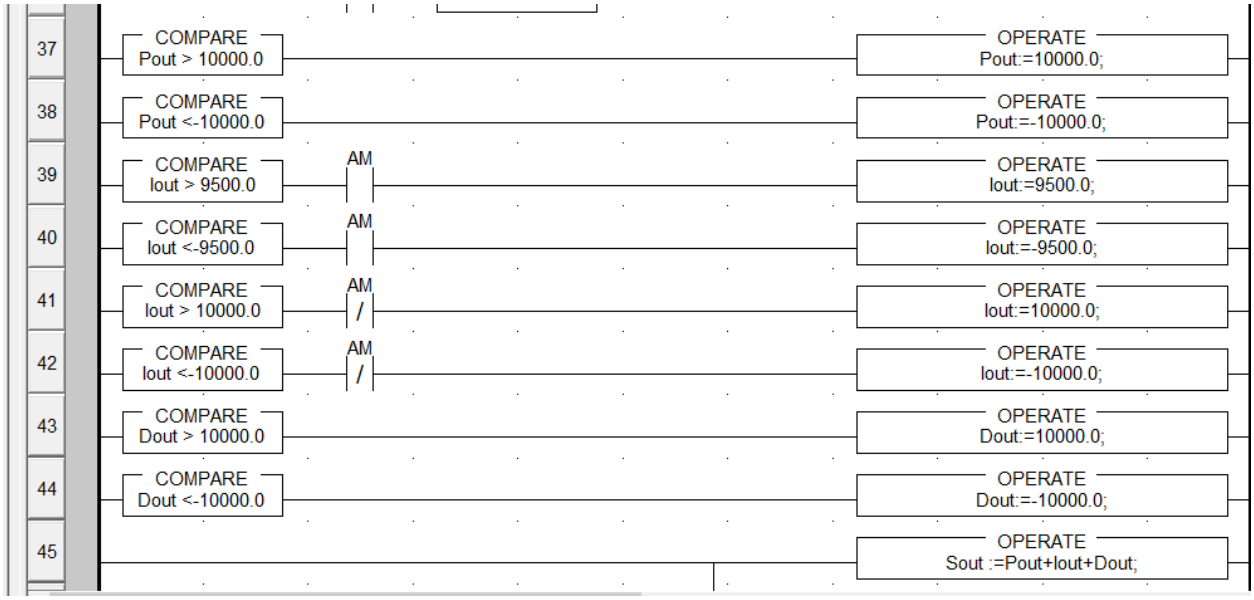


Fig 3.13 Integral windup

In order to avoid integral windup, whenever the output becomes equal or above 10 000, the error signal is clipped to within limits to avoid windup. Also, when the output is less or equal to 0 the error is clipped to also avoid wind up. That way all three constants are kept within the 10 000 limit.

3.5.7 PID BLOCK FINAL OUTPUT (P_{OUT} + I_{OUT} + D_{OUT})

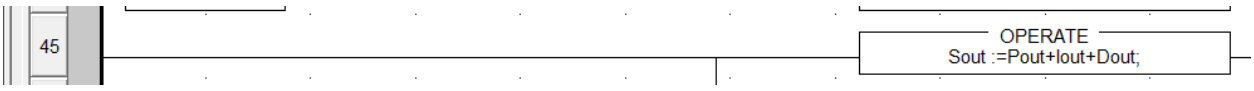


Fig 3.14 PID total output

after determining all the three components namely proportional, integral and derivative, the components are then added together to get the final PID result.

3.5.8 PID BLOCK FINAL OUTPUT CLIPPING

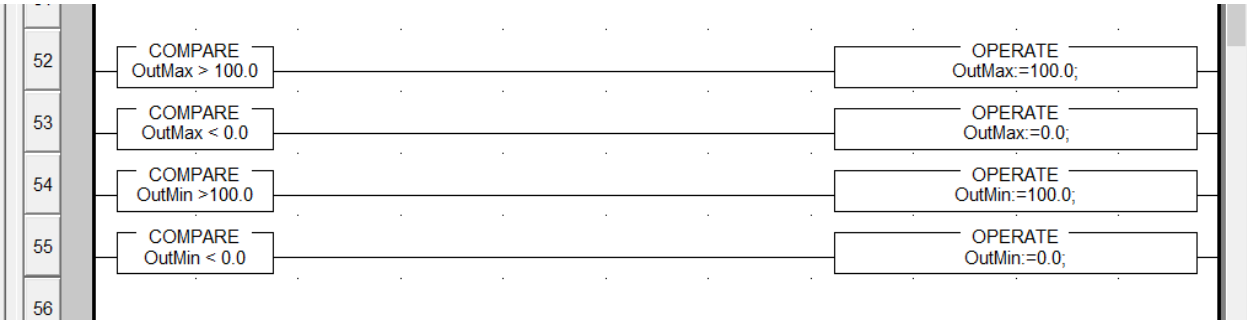
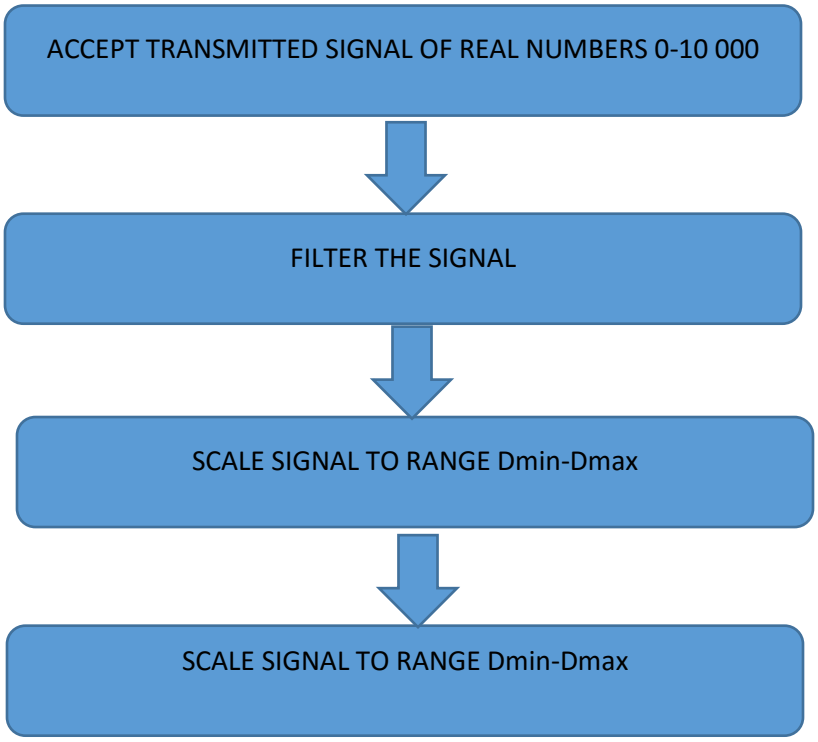


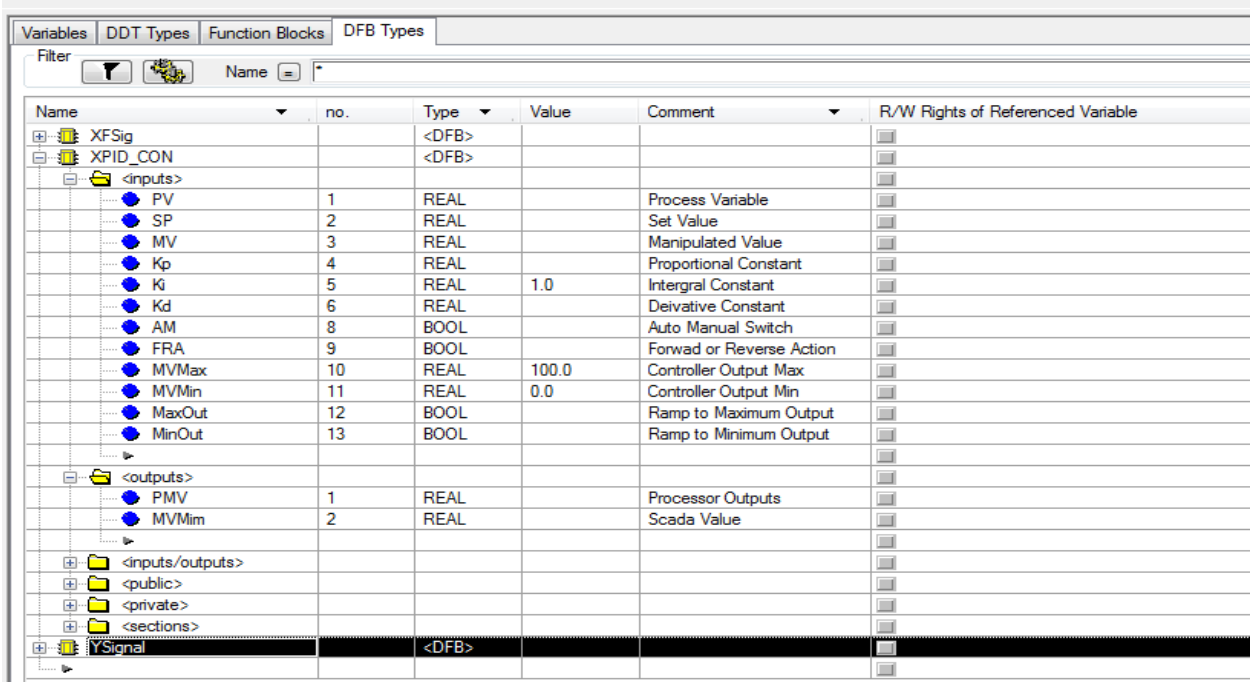
Fig 3.15 PID output clipping

This function of the PID block is the one that clips the final output of the PID block to the ranges of between 0 and 100.

3.6 OUTPUT SIGNAL CONDITIONING TECHNIQUE AND DESIGN (Y_SIG)

Y_SIG BLOCK FLOW CHART





Name	no.	Type	Value	Comment	R/W Rights of Referenced Variable
XFSig		<DFB>			
XPID_CON		<DFB>			
<inputs>					
PV	1	REAL		Process Variable	
SP	2	REAL		Set Value	
MV	3	REAL		Manipulated Value	
Kp	4	REAL		Proportional Constant	
Ki	5	REAL	1.0	Intergral Constant	
Kd	6	REAL		Deivative Constant	
AM	8	BOOL		Auto Manual Switch	
FRA	9	BOOL		Forwad or Reverse Action	
MVMax	10	REAL	100.0	Controller Output Max	
MVMin	11	REAL	0.0	Controller Output Min	
MaxOut	12	BOOL		Ramp to Maximum Output	
MinOut	13	BOOL		Ramp to Minimum Output	
<outputs>					
PMV	1	REAL		Processor Outputs	
MVMim	2	REAL		Scada Value	
<inputs/outputs>					
<public>					
<private>					
<sections>					
YSignal		<DFB>			

Fig 3.16 Declared inputs and outputs for Y_SIG

The Y_SIG block takes inputs MV, Dmin, Dmax, CTim and Rev.

The Y_SIG block then outputs the DMV, PWMF, PWMR, MimMV as shown above. The pulse width modulation signal is outputted for both forward and reverse action so that control logic can be fully archived. Another signal MimMV is sent to the Scada display for mimic display.

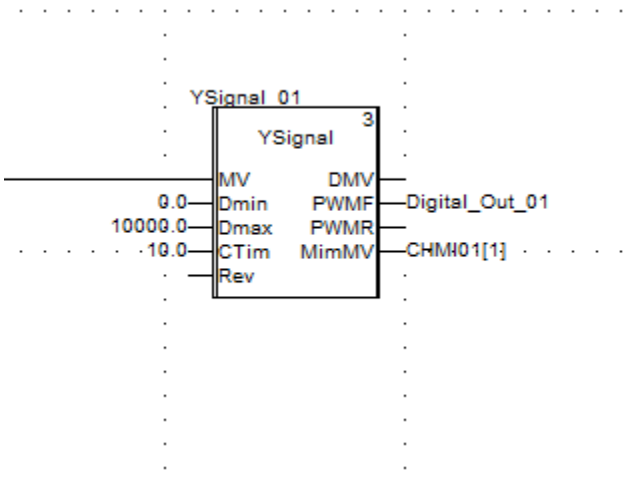


Fig 3.17 Y_SIG Function block

3.6.1 RE-SCALING MV FROM 0-10 000 TO DEVICE SCALE AND CONVERSION OF REAL NUMBERS TO INTEGERS TO MATCH DATA TYPE

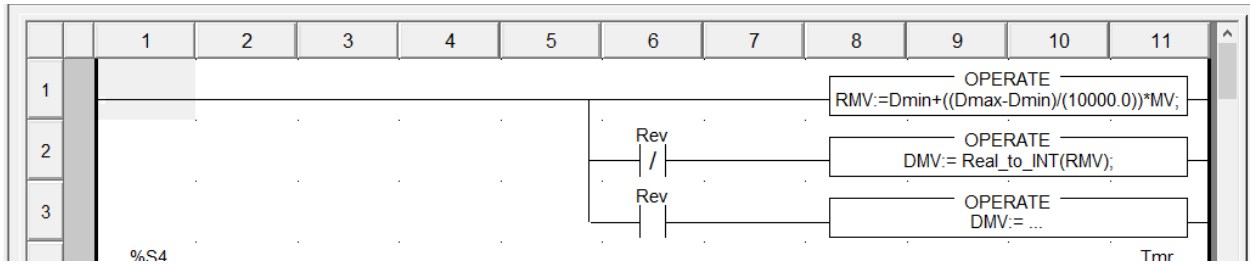


Fig 3.18 Rescaling of output signal

The function block can be used to change the value range of a numerical variable.[3]In this block, the signal is rescaled from values of 0 – 10 000 to ranges of 0 – 100 to match our output device. This is achieved using the formula below,

$$RMV = Dmin + \left(\frac{Dmax - Dmin}{10\ 000.0}\right) * MV$$

Where RMV is the rescaled manipulated variable

D_{min} is the output device minimum

D_{max} is the output device maximum

MV is the manipulated variable

3.6.2 PWM TIME PULSE CREATION

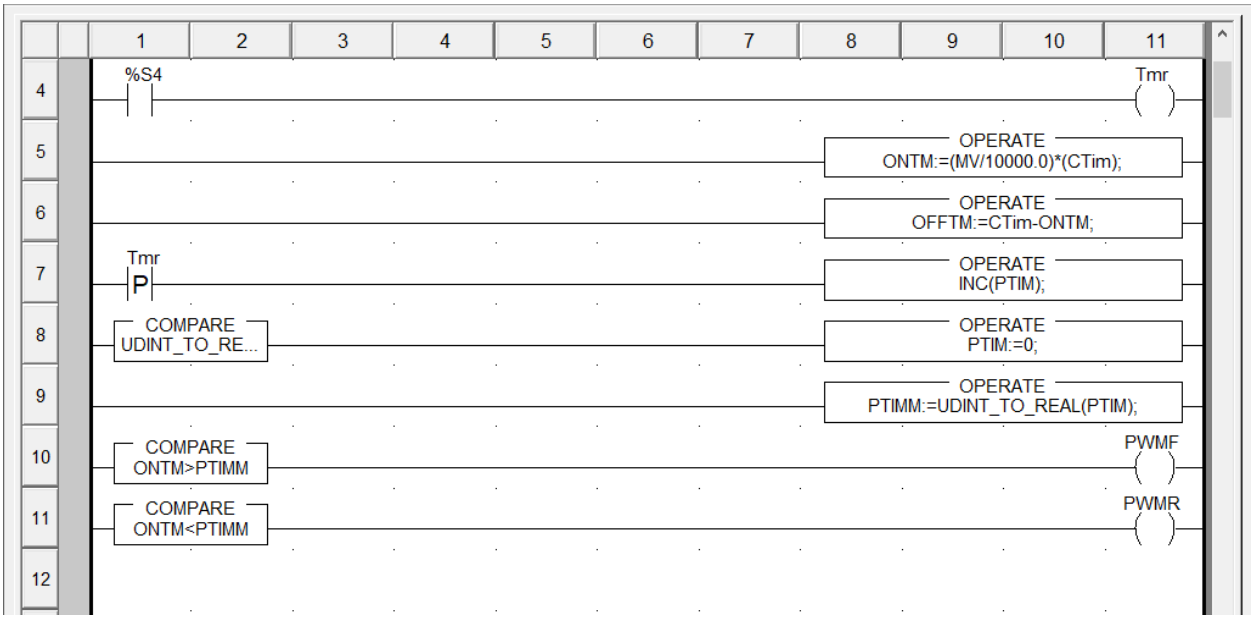


Fig 3.19 PWM generation

The conversion of analogue values into binary output signals is achieved through pulse width modulation [2]. In the Y_SIG block, the PWM pulse is created using the technique of off time (OFFTM) and on time (ONTM).

For the on-time technique, the equation below is used

$$ONTM = \left(\frac{MV}{10000.0}\right) * (CTim$$

Where ONTM is the on-time

MV manipulated variable

CTim is pulsewidth modulation cycle time.

For the off-time technique, the equation below is used

$$OFFTM = CTim - ONTM$$

Where OFFTM is the off-time

CTim is the pulsewidth modulation cycle time

ONTM is the on-time

PTIMM is a timer that controls the on and off period of cycle, it can be set at any percentage for time on and off of the cycle.

So, whenever the ONTM >PTIMM a forward pulse width modulated signal is generated (PWMF) and when the ONTM<PTIMM is a reverse pulsewidth modulated signal is generated (PWMR) in the output.

3.7.0 PLC OUTPUT CARD

The plc output card will take Y_SIG output signal in the form of pulse width modulation signal and convert them to an PWM output for final control element.

3.8 EQUIPMENT USED IN THE STUDY

EQUIPMENT	SPECIFICATIONS	QUANTITY
Magelis Schneider electric HMI	10inch TFT 1280x800pixels	1
Schneider PLC Modicon M340	120/240Vac ,Modbus	1
Schneider Electric Connexium Switch	8-Channel	1
Circuit Breakers	250Vac,50-60Hz,3Amp	2
24V dc Power Supply	100-240V input,2.1amp,50-60Hz	1
24V Relay	220Vac Contacts	1
PT100 RTD	PT100	1
Ethernet Connection cable	Ethernet /Modbus	2
RTD to Current Convetor	4-20mA output	1
Water Bath	Closed Vessel	1
Heating Element	220Vac	1

Table 3.1 Equipment used in the study

REFERENCES

[1] Rhinehart, R.R: The Century's greatest contributions to control practice, ISA Trans,2000.

[2] Schneider Electric," Unity Pro Obsolete Control Block Library", June 2005.

[3] Schneider Electric," Unity Pro Control Block Library", June 2005.

[4] Astrom, K.J., Hagglund, T., PID Controllers: Theory, Design and Tuning",2nd Edition,2005.

CHAPTER 4

4.1 RESULTS AND ANALYSIS

This chapter of the dissertation presents the results and analysis of the signal conditioning and PID control blocks as previously described in the previous chapter. The output results will be critically analysed and results will be displayed in the form of pictorial representations taken from the HMI.The set temperature, set point (SP), process variable(PV) and (MV)manipulated variable were set according to the operator needs.

4.2 Results and Process tuning inputs

Fig 4.1 PID tuning control settings

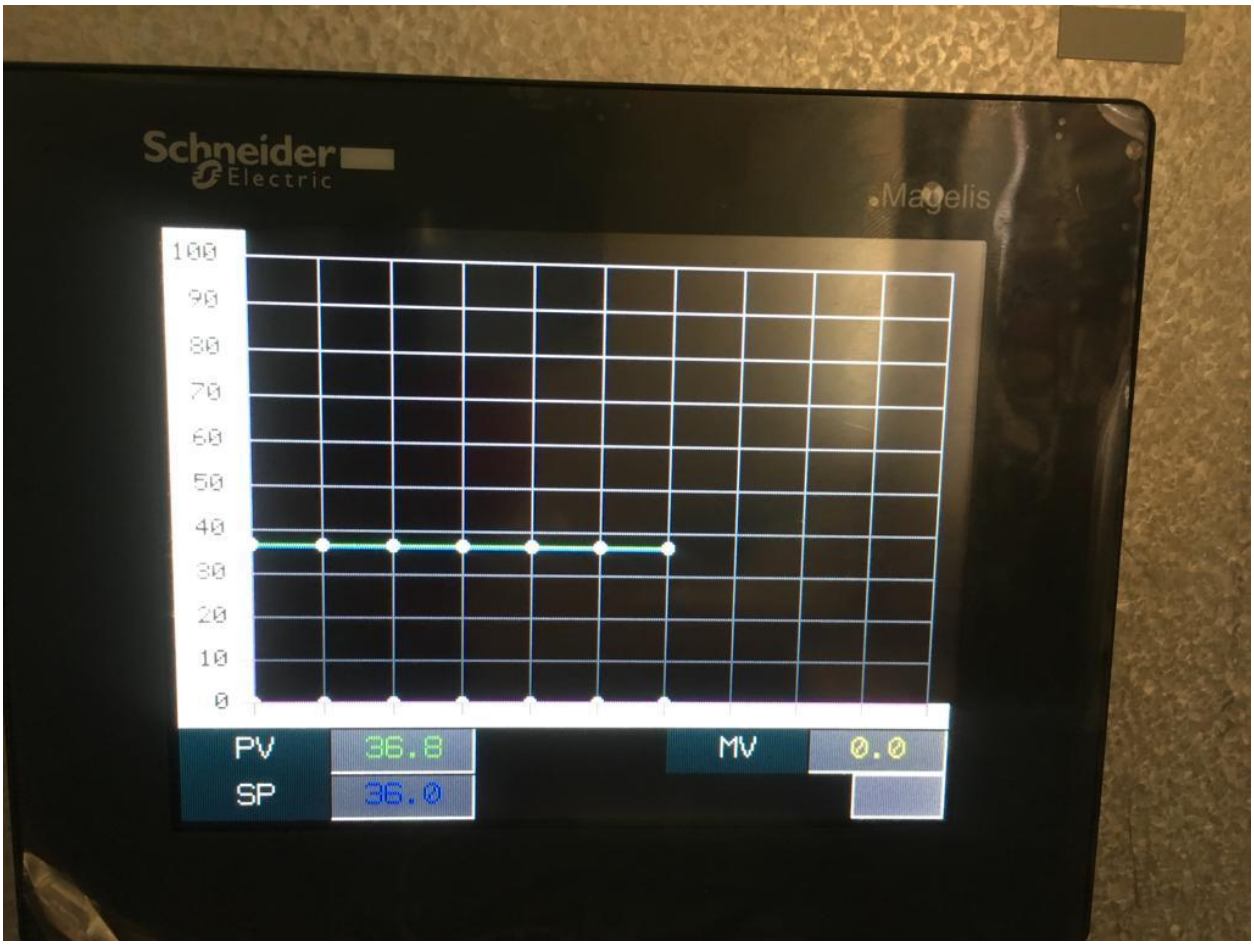


Fig 4.2 Graphical representation of PV, SP and MV

4.3 Summary of Results

It can be noted from the study above that (PV)process variable will always follow the set point (SP)when PID control is inputted on auto, the manipulated variable will act according in reverse or forward action to make sure that the process variable matches the set point.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 CONCLUSION

The performance of the handmade PID control blocks, input and output signal conditioners were excellent in the control of temperature. The results obtained were as equally good as those that can be yielded using inherent PID control blocks and signal conditioners from hybrid PLCs.in other words, effective and timeous temperature control was achieved using an ordinary PLC and handmade PID blocks and signal conditioners.

Conclusively, it be said that cheaper handmade signal conditioners and PID controllers can effectively and accurately control complex temperature control loops without difficulties.

5.2 RECOMMENDATIONS OF THE STUDY

The focus of the study was mainly on the cost of hybrid controllers and the cost impact they have on growing and already developed companies.it can therefore be seen that hybrid controllers can be very costs to the tune of hundreds of thousands and some companies might struggle to acquire these for effective process control. The author recommends the following for further study:

1. To have PID control and signal conditioning be applied to all varying processes that require sensitive and accurate control.

2. To also have other simple PLC brands besides modicon M340 be used to prove the effectiveness of the same PID control and signal processing concept.

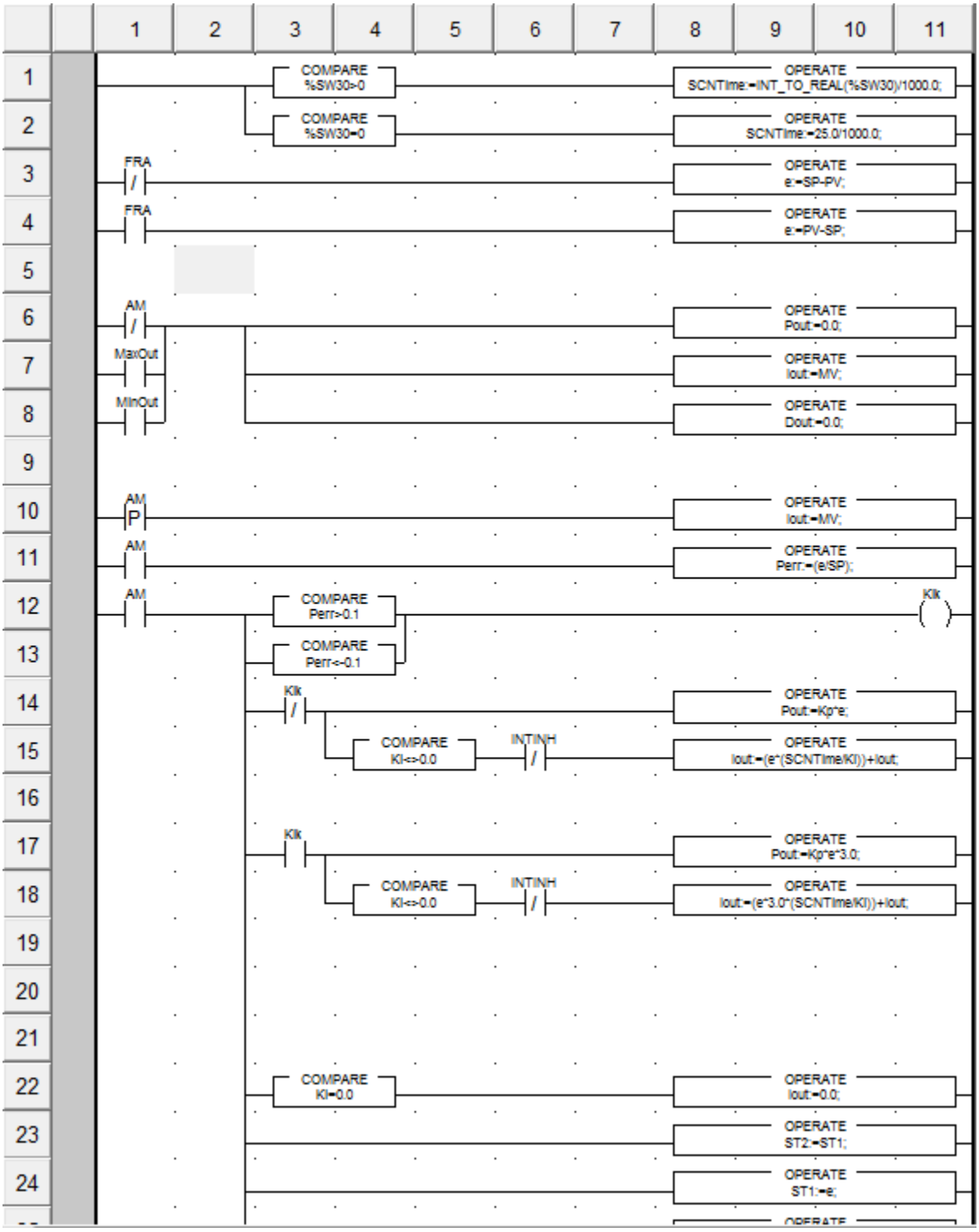
Appendix 1: CODE for Input Signal Conditioning

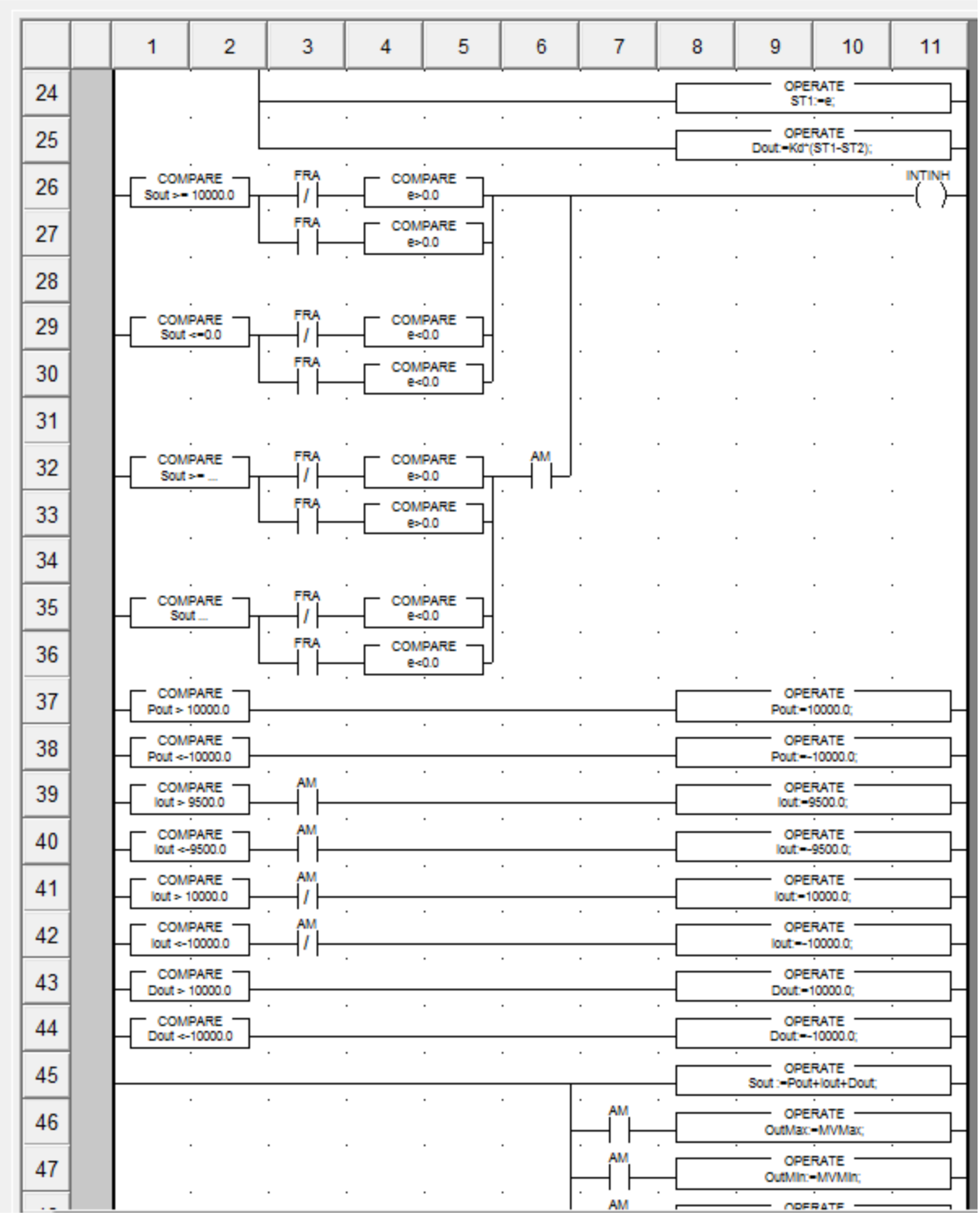
```
(* Loading the Xsignal ALgorithm into the Functional Block *)

(* Calculating Mimic and Display Value *)


T:=0;

IF SMP < 1 THEN SMPL:=1; ELSE SMPL:=SMP; END_IF;

IF SMP >= 200 THEN SMPL:=200; END_IF;


N:=SMPL;

STOT:=0.0;


FOR T:=0 TO SMPL DO

        SDAT[(N-T)]:=SDAT[((N-1)-(T))];

        STOT:=SDAT[(N-T)]+STOT;

END_FOR;


SDAT[0]:=INT_TO_REAL(RawPV);

SAVG:=STOT/(INT_TO_REAL(SMP)+1.0);


Dpoint:=SAVG;

Dgrad:=((E_max-E_min)/(D_max-D_min));

DCmin:=(E_min-(((E_max-E_min)/(D_min-D_max))* D_min));
```

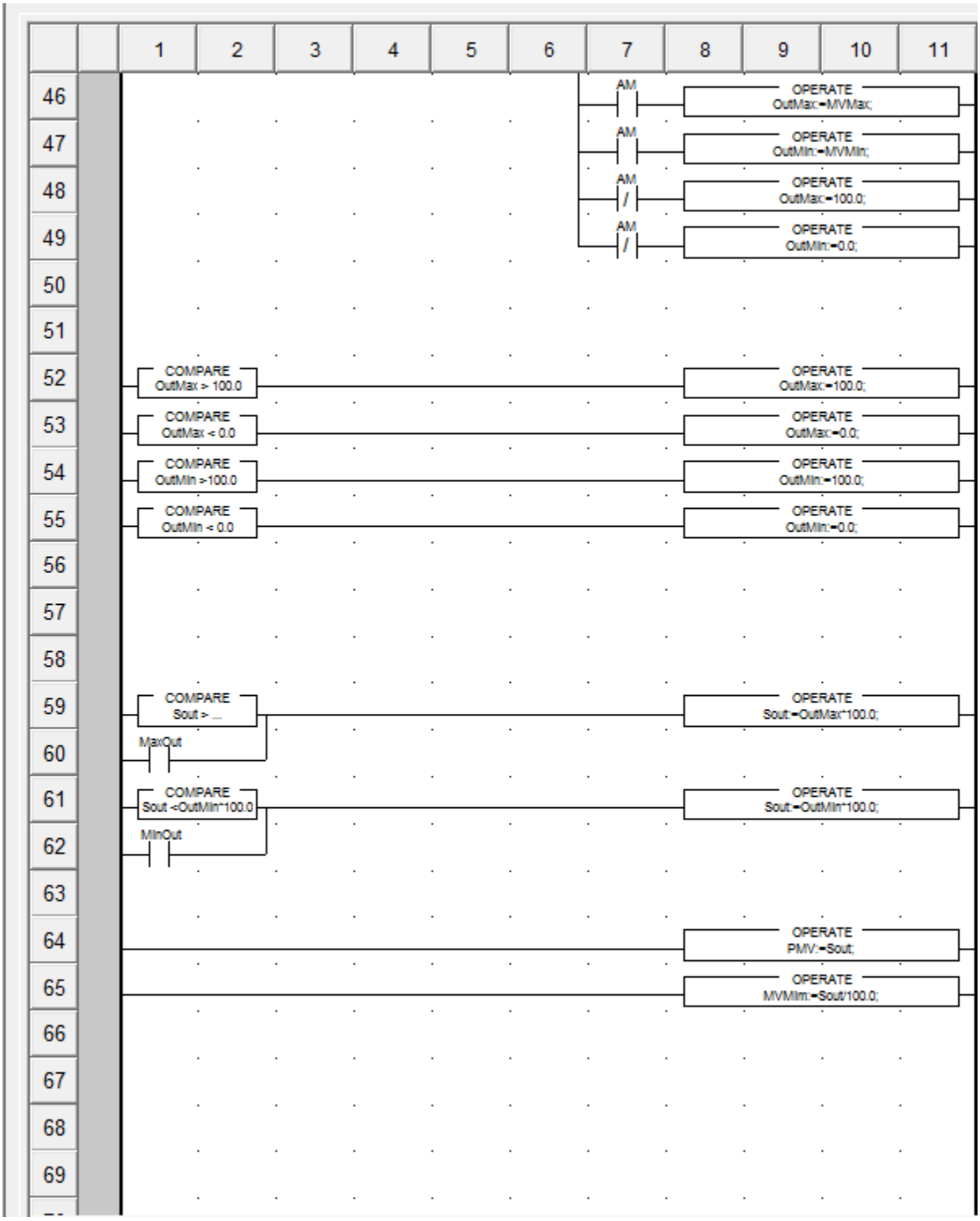
```
DCmax:=(E_max-(((E_max-E_min)/(D_min-D_max))* D_max));

DCres:=(E_min-(((E_max-E_min)/(D_min-D_max))* D_min));


MimPV:=dgrad*dpoint+DCmin;


(* Calculating PV Signal to Controller Value range 0 to 10000 *)


PV_PLC := -((10000.0)/(D_max-D_min))*D_min +((10000.0)/(D_max-
D_min))*int_to_real(RawPV);


(* Calculating SP Signal to Controller Value range 0 to 10000 *)


SV_PLC:=-(((10000.0)/(E_max-E_min))*E_min)+((10000.0)/(E_max-E_min))*MimSP;


MV_PLC:=((10000.0)/(100.0-0.0))*MimMV;



(* Clipping Outputs into respective ranges*)


If MimPV > E_max then MimPV:=E_max; End_if;

If MimPV < E_min then MimPV:=E_min; End_if;


If SV_PLC>10000.0 then SV_PLC:=10000.0; End_if;
```

```
If SV_PLC<0.0 then SV_PLC:=0.0;End_if;


If PV_PLC>10000.0 then PV_PLC:=10000.0; End_if;
If PV_PLC<0.0 then PV_PLC:=0.0;End_if;


If MV_PLC>10000.0 then MV_PLC:=10000.0; End_if;
If MV_PLC<0.0 then MV_PLC:=0.0;End_if;



End_If;
```

Appendix 2: code for PID control algorithm




1 2 3 4 5 6 7 8 9 10 11
24 OPERATE ST1:=e;
25 OPERATE Dout:=Kd*(ST1-ST2);
26 COMPARE Sout >= 10000.0 FRA / COMPARE e>0.0 INTINH
27 FRA COMPARE e>0.0
29 COMPARE Sout <=0.0 FRA / COMPARE e<0.0
30 FRA COMPARE e<0.0
32 COMPARE Sout >= ... FRA / COMPARE e>0.0 AM
33 FRA COMPARE e>0.0
35 COMPARE Sout ... FRA / COMPARE e<0.0
36 FRA COMPARE e<0.0
37 COMPARE Pout > 10000.0 OPERATE Pout:=10000.0;
38 COMPARE Pout <-10000.0 OPERATE Pout:=-10000.0;
39 COMPARE Iout > 9500.0 AM OPERATE Iout:=9500.0;
40 COMPARE Iout <-9500.0 AM OPERATE Iout:=-9500.0;
41 COMPARE Iout > 10000.0 AM / OPERATE Iout:=10000.0;
42 COMPARE Iout <-10000.0 AM / OPERATE Iout:=-10000.0;
43 COMPARE Dout > 10000.0 OPERATE Dout:=10000.0;
44 COMPARE Dout <-10000.0 OPERATE Dout:=-10000.0;
45 OPERATE Sout :=Pout+Iout+Dout;
46 AM OPERATE OutMax:=MVMax;
47 AM OPERATE OutMin:=MVMin;
AM


1 2 3 4 5 6 7 8 9 10 11
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
AM
OPERATE
OutMax:=MVMax;
AM
OPERATE
OutMin:=MVMin;
AM /
OPERATE
OutMax:=100.0;
AM /
OPERATE
OutMin:=0.0;
COMPARE
OutMax > 100.0
OPERATE
OutMax:=100.0;
COMPARE
OutMax < 0.0
OPERATE
OutMax:=0.0;
COMPARE
OutMin >100.0
OPERATE
OutMin:=100.0;
COMPARE
OutMin < 0.0
OPERATE
OutMin:=0.0;
COMPARE
Sout > ...
OPERATE
Sout:=OutMax*100.0;
MaxOut
COMPARE
Sout <OutMin*100.0
OPERATE
Sout:=OutMin*100.0;
MInOut
OPERATE
PMV:=Sout;
OPERATE
MVMin:=Sout/100.0;

Appendix 3: Code for output signal conditioning