A Novel Principle based on Modified relay feedback Approach for auto tuning of robust PID Controller

Major Project Report

Submitted in partial fulfillment of the requirements

For the degree of

Master of Technology

in

Electronics & Communication Engineering (Embedded System)

By

Patel Ankur A.

(12MECE12)

Electronics & Communication Engineering Branch

Electrical Engineering Department

Institute of Technology

Nirma University

Ahmedabad - 382481

May 2014

A Novel Principle based on Modified relay feedback Approach for auto tuning of robust PID Controller

Major Project Report

Submitted in partial fulfilment of the requirements

for the degree of

Master of Technology

In

Electronics & Communication Engineering(Embedded System)

By

Patel Ankur A.

(12MECE12)

Under the Guidance of

External Project Guide:

Mr. Vijay Patel
Executive Engineer(R&D Depart.),
Masibus Automation & Instrumentation Pvt. Ltd.,
Gandhinagar.

Internal Project Guide:

Prof. Vijay Savani
Assistant Professor(EC Depart.),
Institute of Technology,
Nirma University, Ahmedabad.

Electronics & Communication Engineering Branch

Electrical Engineering Department

Institute of Technology

Nirma University

Ahmedabad - 382481

May 2014

Declaration

This is to certify that

a. The thesis comprises of my original work towards the degree of Master of Technology in Electronics & Communication(Embedded System) at Nirma University and has not been submitted elsewhere for a degree.

b. Due acknowledgement has been made in the text to all other material used.

- Patel Ankur A.

NIRMA UNIVERSITY
INSTITUTE OF TECHNOLOGY

Certificate

This is to certify that the Major Report entitled "**A Novel Principle based on Modified relay feedback Approach for Auto Tuning of Robust PID Controller**" submitted by **Mr. Ankur A. Patel (12MECE12)**, towards the partial fulfilment of the requirements for the degree of **Master of Technology in Electronics & Communication(Embedded System)** of **Nirma University of Science and Technology; Ahmedabad** is the record of work carried out by him under our supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for the examination. The results embodied in this major project, to the best of our knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Date: Place: Ahmedabad

Prof. Vijay Savani
Internal Guide

Mr.Vijay Patel
External Guide

Dr. N.P. Gajjar
Program Coordinator

Dr. P.N.Tekwani
Head of EE Dept.

Dr. K. Kotecha
Director, IT

Blank Page for Company Certificate

Acknowledgement

I would have never succeeded in completing my Thesis without the cooperation, Encouragement and help provided to me by various people.

Firstly, my sincere thanks to the Masibus R&D team, for their help during this training. Their wisdom, clarity of thought and support motivated me to bring this project to its present state.

I am highly indebted to R&D team HOD, **Mr.Suhant Raval**, Executive Engineer & External Guide, **Mr.Vijay Patel**. My immediate supervisors **Mr.Ritesh Panchal** & **Mrs.Khyati M. Jasani**. for providing necessary information regarding the project and also for their constant guidance, supervision, kind co-operation, and invaluable support in all aspects. My thanks and appreciations also go to my colleagues and team members in developing the project and for providing me with a lively and energetic work environment.

I would like to express my sincere gratitude to **Dr.Ketan Kotecha** (Director, Nirma University, Ahmedabad) for his continuous guidance, support and enthusiasm. I would take this opportunity to thank **Dr.(Prof.)P.N.Tekwani** (Head of Department, Electrical Engineering), **Dr.N.P.Gajjar** (Professor and Program Coordinator,M.Tech-EC(Embedded System)), Internal Guide **Prof.Vijay Savani** (Assist. Prof.,M.Tech-EC(Embedded System)) and all the faculties at **Nirma University (Embedded System)**, for their vision and relentless effort, support, and encouragement to provide me with this excellent opportunity to carry out my project work in such a highly renowned and esteemed organization.I am equally thankful to **Masibus Automation & Instrumentation Pvt. Ltd.** for providing me the invaluable exposure to the industry and the current market trends.

Finally, I would like to express my heartfelt thanks to my parents and colleagues for their blessings and for their constant love and support.

- Patel Ankur A.

(12MECE12)

Abstract

This project concerned with the designing a low cost PID temperature controller. Because of their simplicity, robustness and successful practical application, PID controllers are the most popular and widely-used controllers in industry. Many PID design methods have been proposed, each has its advantages and limitations. However finding appropriate parameters for the PID controller is still not easy task. The objective is to achieving an important design compromise, acceptable stability, linearity and medium fastness of response. In this project, Atmega128 has been used as a brain to the prototype. PID controller has been selected as a controller because it has faster response than the conventional on-off controller. The experimental result reveals that the PID controller can be implemented into the low cost prototype and able to control the temperature according to standard regulation.

Contents

List of Tables

List of Figures

Symbols and Abbreviations

PID .... Proportional-Integral-Derivative

SP .... Set-Point

PV .... Process Variable

°C .... Degree Celsius

K_p .... Proportional Gain

K_i .... Integral Gain

K_d .... Derivative Gain

T_i .... Integral Time

T_d .... Derivative Time

PB .... Proportional Band

CO .... Control Output

P_u .... Ultimate Period

K_u .... Ultimate Gain

PLC .... Programmable Logic Controller

DCS .... Distributed Control System

SCADA .... supervisory control and data acquisition

RTU .... Remote Terminal Unit

SMPS .... Switching Mode Power Supply

RTD .... Resistance Temperature Detectors

TC .... Thermocouple

IEC .... International Electro-technical Commission

NMRR .... Normal-Mode Rejection Ratio

CMRR .... Common-Mode Rejection Ratio

ASCII .... American Standard Code for Information Interchange

Chapter 1

Introduction

1.1 Background

Most of heavy industries in the manufacturing field have their own manufacturing target. To achieve the target, most of the machines are run continuously for 24 hours and stop only for maintenances. The most important part in manufacturing is maintenance which is divided into three parts, preventive maintenance, predictive maintenance and corrective maintenance.[15] Theoretically, preventive maintenance is the most important part in protecting equipments from being damaged and expensive corrective maintenance can be avoided.

Linearity and stability of the Temperature is the main problem that has been faced by heavy industries in an attempt to achieve a production target.

1.2 Problem definition

Why PID controller must be invented? Actually, the main problem is most of the industry did not monitor the operating temperature of their equipment / plants. So, it is more important to monitor and control the excessive temperature. Excessive heat can cause failure of the equipment / plant and will affect the operating budget to replace all the damaged equipments. Once the prototype is ready, the operator of the equipment does not have to worry about excessive heat produce because the temperature controller will make sure the operating temperature will follow the standard requirement of the IEC.

1.3 Objective of Project

There two objectives that need to be achieved in this project, and there are:

- To overcome linearity problem in the industry by designing a new and low cost prototype using the proposed PID controller tuning methods.
- To develop a system prototype to evaluate the performance of proposed controller with good linearity, stability and robust performance in critical Condition.

1.4 Scope of Project

The scope of the project includes identifying on how PID controller can be implemented in the low cost system. The other scopes are providing good linearity and stability with maintaining the safe operating temperature and conduct an experiment at Process control Lab.

1.5 Work Breakdown

In order to achieve the objective in this project, there are several tasks that need to be done as shown in below Figure.1.1. The tasks are divided into three categories and they are studying, design and implementation.

The related items that need to be a study in detail are the principles of PID controller and its Tuning methods, micro-controller architecture, programming for micro-controller etc.

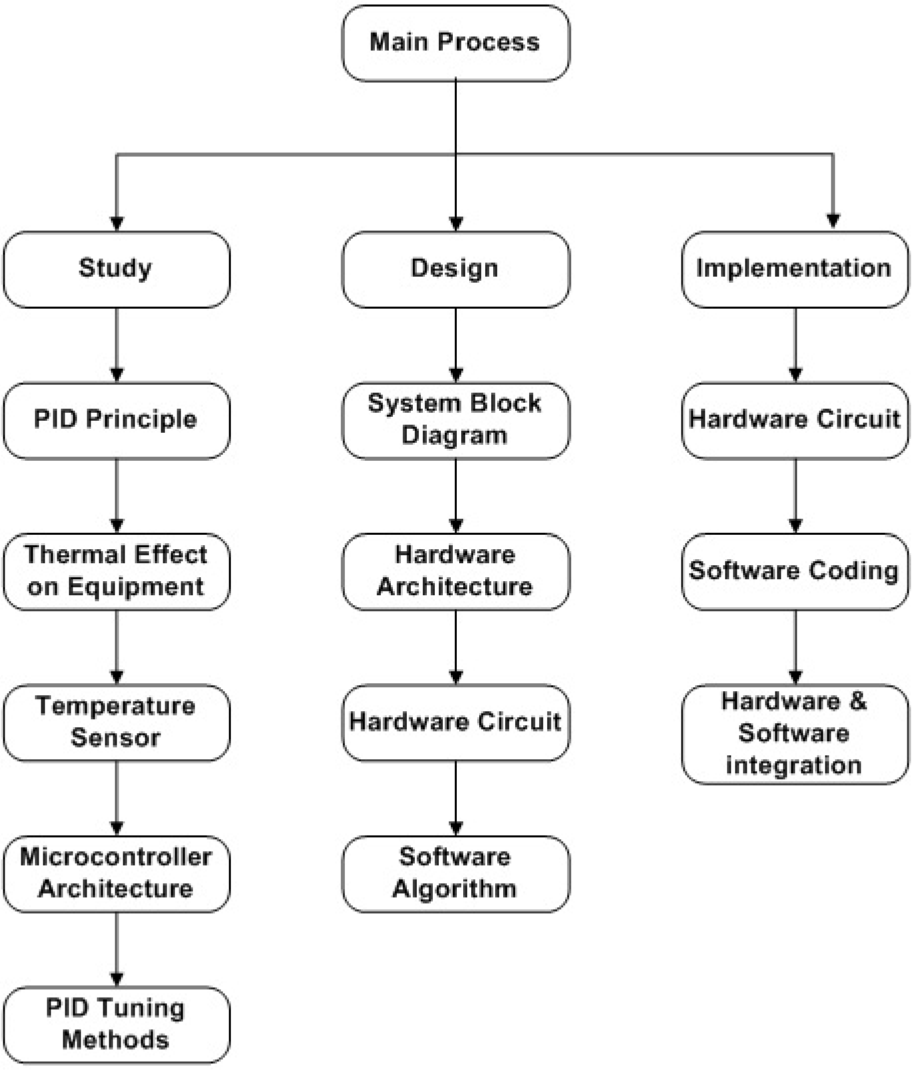


Figure 1.1: Work breakdown

1.6 Project Work Flow

The Figure.1.2 shows the work flow of this project and has been simplified in block diagram as below. These working flows are continuously done in the time given by the Company within 2 semesters.

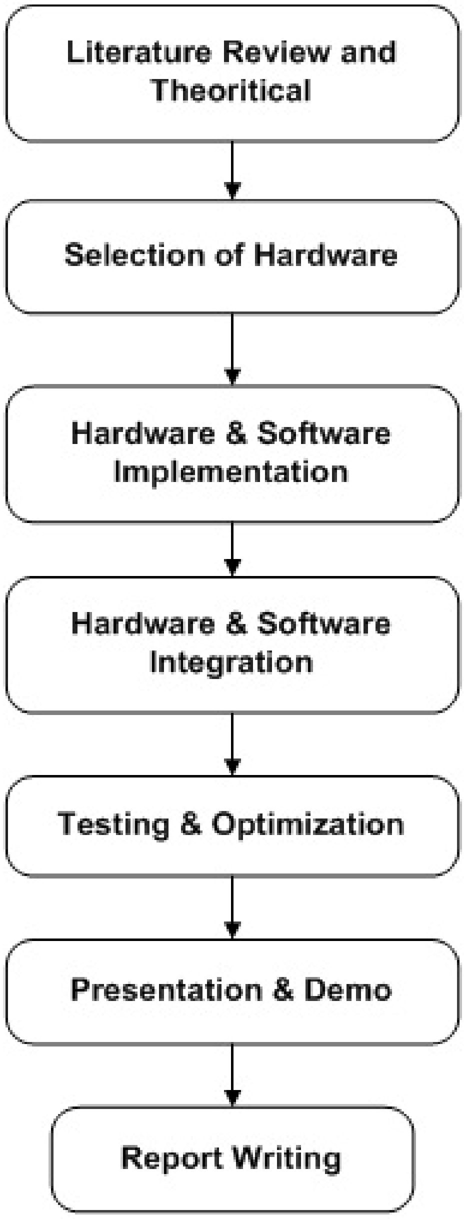


Figure 1.2: Project Work flow

1.7 Gantt Chart

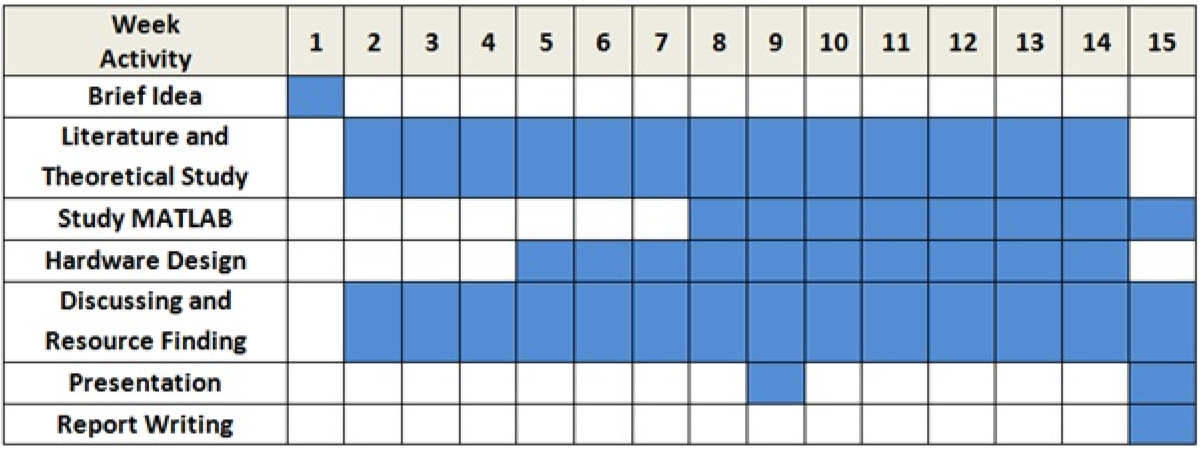


Figure 1.3: Gantt chart of the project timetable for semester 3

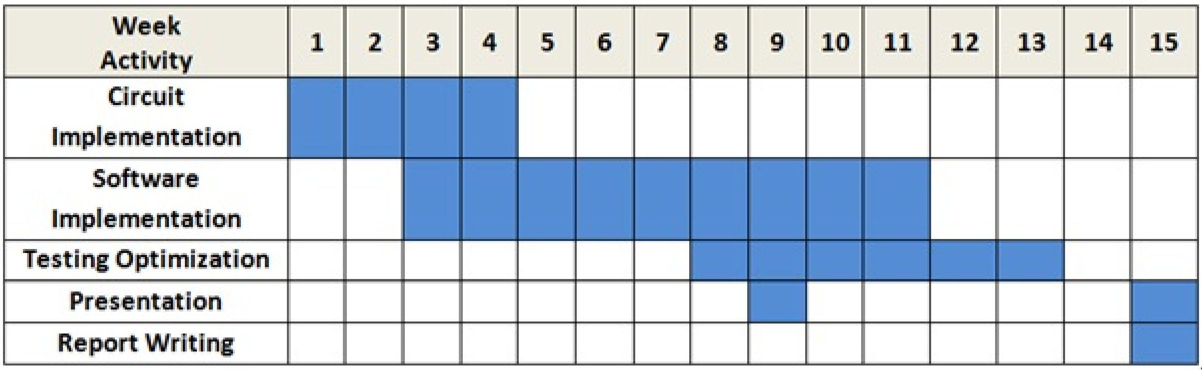


Figure 1.4: Gantt chart of the project timetable for semester 4

1.8 Thesis Organization

These reports consist of five chapters in total. The framework of the project will be described as follows:

Chapter 1 provides a brief introduction about the Industrial PID Controller, objective and scope of project, project work breakdown and work flow of project duration.

Chapter 2 describes the literature review of the project. It contains the literature which we studied before starting the project. It also describes introduction to basic PID parameters effects on closed loop system. Also describe the basic theory of Modbus Protocol for communication.

Chapter 3 discusses the Prototype design of product, proposed modified relay based auto tune technique. It consists of hardware block diagram and software flow chart which is used in this project.

Chapter 4 shows the implemented project's outputs result and some discussion on some testing procedure and test results for RTD input type Temperature sensors.

Chapter 5 concludes the project and suggest some more research work in future scope for precious industrial application.

In this chapter discuss about project definition that why PID Controller are very useful now a day in industries then objective and scope of this project and also talk about work distribution during a project and flow of work in terms of flow chart and Gantt chart.

Chapter 2

Literature Review

2.1 Basics of PID Controller

The PID control algorithm is one of the most popular approaches for industrial field to apply in the process control systems. The main reason why PID is the main popular method of control in industry is because the simple structure of the PID itself, it also conceptually easy to understand and makes the manual tuning is possible.[15] PID is proportional integral and derivative controller which is a linear combination of the input, the derivative of input and integral of input. Theoretically there are several types of tuning methods that can be used in tuning PID.[16][17] There are Ziegler-Nichols rules, symmetric optimum rule, Ziegler-Nichols complementary rule, some overshoot rule, no overshoot rule and integral of absolute error rule.

The PID controller calculation (algorithm) involves three separate constant parameters, and is accordingly sometimes called three-term control, the proportional, the integral and derivative values, denoted P, I, and D.[18] Heuristically, these values can be interpreted in terms of time: P depends on the present error, I on the accumulation of past errors, and D is a prediction of future errors, based on current rate of change.[15] The weighted sum of these three actions is used to adjust the process via a control element such as the position of a control valve or the power supply of a heating element.[16][17]

In the absence of knowledge of the underlying process, a PID controller is the best controller. By tuning the three parameters in the PID controller algorithm, the controller can provide control action designed for specific process requirements.[1] The response

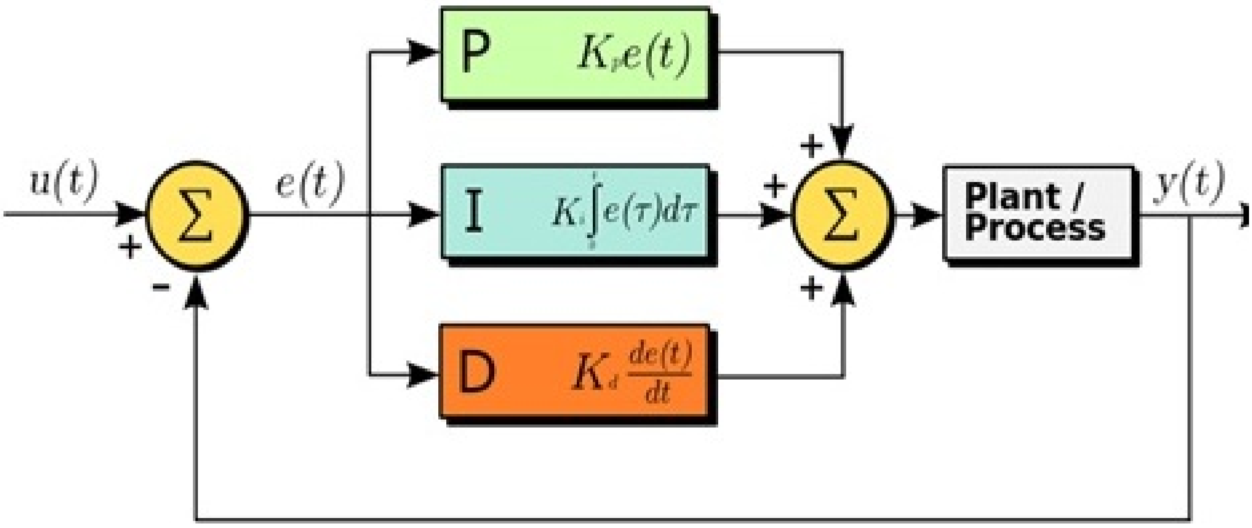


Figure 2.1: PID control for general system[1]

of the controller can be described in terms of the responsiveness of the controller to an error, the degree to which the controller overshoots the set point and the degree of system oscillation.[19] Note that the use of the PID algorithm for control does not guarantee optimal control of the system or system stability.[15]

There are also some disadvantages that exist in PID method which lead to not enough tuning and cause oscillation in practical. These are the factors which lead to the unsatisfactory in tuning PID controller.[19]

- Non-linearity exists in the system such as directional dependent actuator or plant dynamics.

- Various uncertainties such as modelling error and noise which involved in the controller systems.

- Continuous tuning by time may be necessary due to ageing and general wear of the system applied.

- The load which often variable and affect the dynamic performance of the system.

2.2 Control Modes

PID controller has proportional, integral and derivative control modes. These modes each react differently to the error, and also, the degree of control action is adjustable for each mode.

2.2.1 Proportional Controller

The proportional control mode changes the controller output in proportion to the error. The adjustable setting here is called the Controller Gain (Kc), sometimes also referred to as a PID controller's P-setting or its proportional setting.[6]

The control action is proportional to both the controller gain and the error. A higher controller gain will increase the amount of output action and so will a larger error.

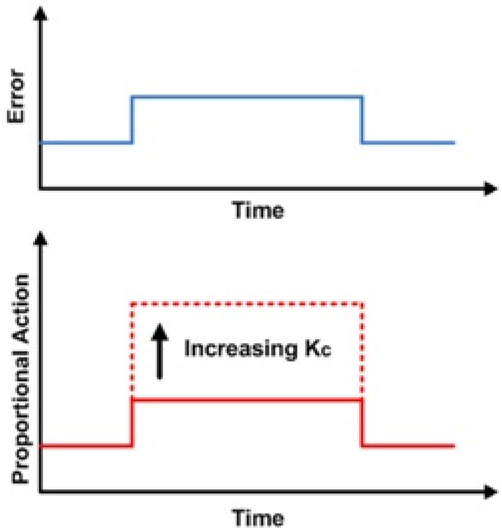


Figure 2.2: Proportional controller action[2]

Although most controllers use controller gain (Kp) as the proportional setting, some controllers use Proportional Band (PB), which is expressed in percent. Below Table.I shows the relationship between Kp and PB.

$$PB = \frac{100\%}{K_P} \quad \ldots\ \ldots\ \ldots.[2.1]$$

Table I: Relationship between K_p a PB[3]
nd

Controller Gain (K_p)	Proportional Band (PB)
0.1	1000
0.2	500
0.5	200
1	100
2	50
5	20
10	10

2.2.2 Integral Controller

The use of proportional control alone has a large drawback like Offset. Offset is a sustained error that cannot be eliminated by proportional control alone. Integral action appears as a ramp of which the slope is determined by the size of the error, the controller gain and the Integral Time (Ti), also called the I-setting of the controller.[2]

$$\text{I Action} = \frac{K_C}{T_i} * \int e(\tau)d(\tau) \qquad [2.2]$$

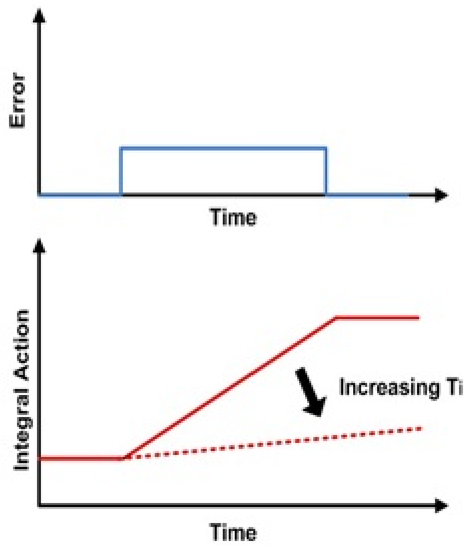


Figure 2.3: Integration controller action[1]

Most controllers use integral time in minutes as the unit for integral control, but some others use integral time in seconds, Integral Gain in Repeats / Minute or Repeats / Second. Table.II compares the different integral units.[18][19]

Table II: Integral Control Units[3]

Integral Time		Integral Gain	
Minutes	Seconds	Rep/Minutes	Rep/Seconds
0.05	3	20	0.333
0.1	6	10	0.167
0.2	12	5	0.0833
0.5	30	2	0.0333
1	60	1	0.0167
2	120	05	0.00833
5	300	02	0.000333
10	600	01	0.000167
20	1200	0.05	0.000833

2.2.3 Derivative Controller

The third controller action in a PID controller is derivative. Derivative control is rarely used in controllers. It is very sensitive to measurement noise and it makes tuning very difficult if trial and error methods are applied. Nevertheless, derivative control can make a control loop respond faster and with less overshoot.[15]

The derivative control mode produces an output based on the rate of change of the error. Derivative action is sometimes called Rate. Its action is dependent on the rate of change (or slope) of the error. It has an adjustable setting called Derivative Time (Td), which is the D-setting of the controller.[2]

$$\text{D Action} = K_C * T_d * \frac{de(t)}{dt} \; \dots \dots \dots \; [2.3]$$

Two units are used for the derivative setting of a controller are minutes and seconds.

Derivative control appears to have predictive or anticipative capabilities. Technically this is not true, but PID control does provide more control action sooner than possible

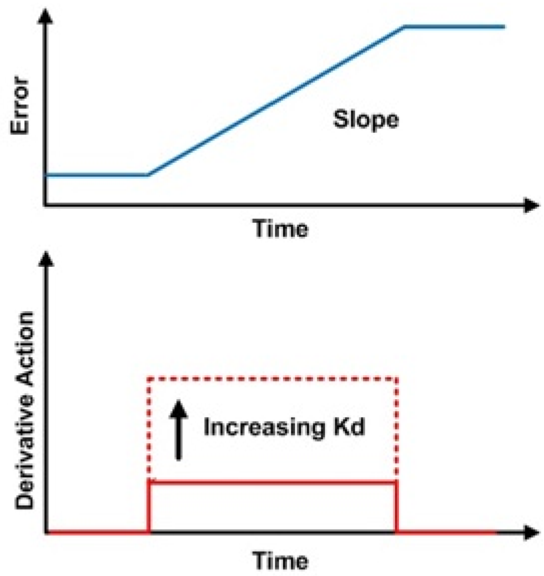


Figure 2.4: Derivative controller action[1]

with P or PI control. Note that how derivative control reduces the time it takes for the level to return to its set point. Also note that with derivative control the controller output appears noisier. This is due to the derivative control mode's sensitivity to measurement noise.[6]

2.2.4 PI Controller

PI controller will eliminate forced oscillations and steady state error resulting in operation of on-off controller and P controller respectively. However, introducing integral mode has a negative effect on speed of the response and overall stability of the system. Thus, PI controller will not increase the speed of response. It can be expected since PI controller does not have means to predict what will happen with the error in near future.[16] This problem can be solved by introducing derivative mode which has ability to predict what will happen with the error in near future and thus to decrease a reaction time of the controller.[6]

PI controllers are very often used in industry, especially when speed of the response is not an issue. Control without D mode is used when:[4]

- Fast response of the system is not required
- Large disturbances and noise are present during operation of the process

- There is only one energy storage in process (capacitive or inductive)

- There are large transport delays in the system

2.2.5 PID Controller

PID controller has all the necessary dynamics like a fast reaction on change of the controller input (D mode), increase in control signal to lead error towards zero (I mode) and suitable action inside control error area to eliminate oscillations (P mode).[4]

Derivative mode improves stability of the system and enables increase in gain Kp and decrease in integral time constant Ti, which increases speed of the controller response.

PID controller is used when dealing with higher order capacitive processes (processes with more than one energy storage) when their dynamic is not similar to the dynamics of an integrator (like in many thermal processes).[17] PID controller is often used in industry, but also in the control of mobile objects (course and trajectory following included) when stability and precise reference following are required. Conventional autopilot is for the most part PID type controllers.[16]

Table III: Effect of each parameter on output response[15]

Action	Rise Time	Overshoot	Settling Time	S-S Error
Increase K_p	Decrease	Increase	Small Change	Decrease
Increase K_i	Decrease	Increase	Increase	Eliminate
Increase K_d	Small Change	Decrease	Decrease	Small Change

2.3 Types of Controller Structures

Controller manufacturers integrate the P, I and D-modes into three different arrangements or controller structures.[6] These are called Series, Ideal and Parallel controller structures. Some controller manufacturers allow you to choose between different controller structures as a configuration option in the controller software.[12]

2.3.1 Series controller structure

This very popular controller structure is also called the Classical, Real or Interacting structure. The original pneumatic and electronic controllers had this structure and we still find it in most PLCs and DCSs today. Most of the controllers tuning rules are based on this controller structure.[12]

$$CO = K_c\left[e(t) + \frac{1}{T_i} * \int e(\tau)d(\tau)\right] X \left[1 + T_d * \frac{de(t)}{dt}\right] \ldots\ldots\ldots [2.4]$$

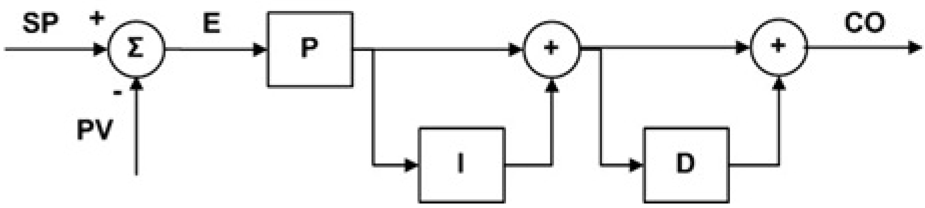


Figure 2.5: Series controller structure[4]

2.3.2 Ideal controller structure

Also called the Non-Interacting, Standard or ISA structure, this controller structure was popularized with digital control systems. If no derivative is used (i.e. Td = 0), the series and ideal controller structures become identical.

$$CO = K_c\left[e(t) + \frac{1}{T_i} * \int e(\tau)d(\tau) + T_d * \frac{de(t)}{dt}\right] \ldots\ldots\ldots [2.5]$$

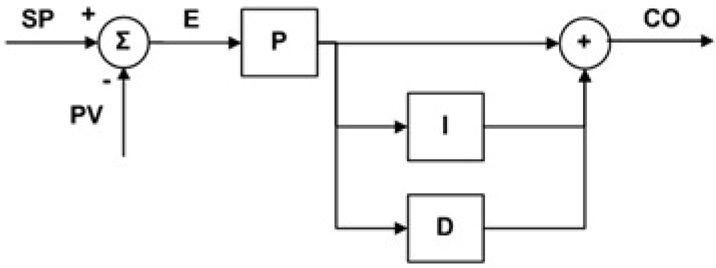


Figure 2.6: Ideal controller structures[4]

2.3.3 Parallel controller structure

Academic-type textbooks generally use the parallel form of PID controller, but it is also used in some DCSs and PLCs. This structure is simple to understand, but really difficult to tune. The reason is that it has no controller gain, but has a proportional gain instead. Tuning should be done by adjusting all the settings simultaneously.[6]

$$CO = K_p * e(t) + K_i * \int e(\tau)d(\tau) + K_d * \frac{de(t)}{dt} \ldots \ldots \ldots [2.6]$$

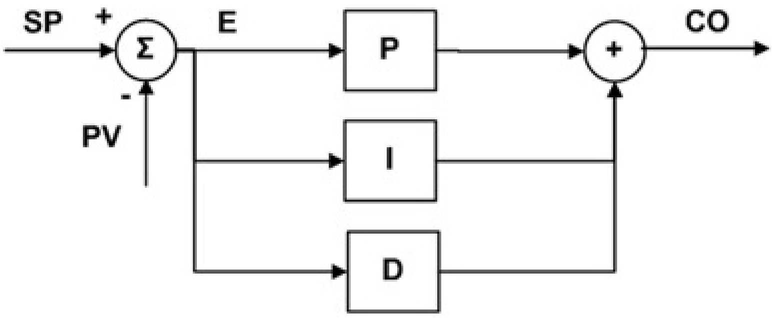


Figure 2.7: Parallel controller structures[4]

2.4 PID Tuning Method: Ziegler Nichols Tuning

Basically, Ziegler Nichols tuning method is one of the several types of tuning method exist in control systems. This type of tuning method is proposed by Ziegler and Nichols in the 1940s and this type of tuning method largely based on certain assumed model.[1][2]

Tuning a control loop is the adjustment of its control parameters (gain/proportional band, integral gain/reset, derivative gain/rate) to the optimum values for the desired control response. Stability (bounded oscillation) is a basic requirement, but beyond that, different systems have different behaviour, different applications have different requirements, and requirements may conflict with one another.[3]

PID tuning is a difficult problem, even though there are only three parameters and in principle is simple to describe, because it must satisfy complex criteria within the limitations of PID control. There are accordingly various methods for loop tuning and more sophisticated techniques are the subject of patents.[4]

$$C(s) = K_p + \frac{K_i}{s} + K_d\, s \ldots\ldots\ldots [2.7]$$

$$C(s) = K_p \left[1 + \frac{1}{T_i\, s} + T_d\, s\right] \ldots\ldots\ldots [2.8]$$

where,

$K_p = proportionalgain$

$K_i = integralgain = K_p/T_i$

$K_d = derivativegain = K_p *_d$

$T_i = resettime$

$T_d = ratetimeorderivativetime$

The proportional term used in the controller generally helps in establishing system stability and improving the transient response while the derivative term is often used to improve the closed response speed when it is necessary to use. Theoretically, the effect of the derivative term is to feed information on the rate of change of the measured variable into the controller action. The most important term in the controller is the integrator term that introduced in the forward loop process.[1][3]

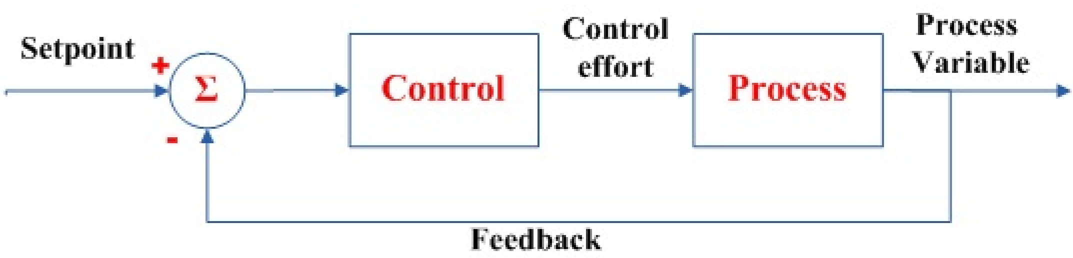


Figure 2.8: PID controller block diagram[5]

Based on the Figure2.8, there is some equation below that may explain the PID controller function.[2][4]

$$u(t) = K_P \left[e(t) + \frac{1}{T_i} * \int e(\tau) d(\tau) + T_d * \frac{de(t)}{dt} \right] \dots \dots \dots [2.9]$$

$$e(t) = \text{Set Point} - \text{Process Variable} \dots \dots \dots [2.10]$$

Difference between Set point and Process Variable, and the control input relation is as shown below Table.IV

2.4.1 Ziegler Nichols First Method

Table.V is the first method of Ziegler and Nichols used to tune PID controller. This type of response is typical of first order system with some transportation delay such as fluid flow and heat. The response is characteristic on two parameters, L the delay time and T constant time. These are usually found by drawing a tangent to the step response at its point of inflection and noting its intersection with time axis and steady state value. The model of the plant is:[1][3]

Based on the Figure2.9 above clearly it shows the over damped second order system response curve which consist L delay time and T for time constant. So based on this Ziegler-Nichols has produced the recipe for tuning method. [1][2][3][4]

Table IV: PID controller variation[3]

Control Type	K_p	K_i	K_d	C(s)
P (Proportional)	±0	Zero	Zero	K_p
I (Integral)	Zero	±0	Zero	K_i/s
D (Derivative)	Zero	Zero	±0	K_d s
PI (proportional+ integral)	±0	±0	Zero	K_p+K_i/s
PD (proportional + derivative)	±0	Zero	±0	K_p + K_d s
PID (proportional+ integral +derivative)	±0	±0	±0	K_p+ K_i/s + K_d s

$$G(s) = \frac{K\,e^{-SL}}{Ts+1} \ldots\ldots\ldots [2.11]$$

2.4.2 Ziegler Nichols Second Method

To render under proportional control, the second method can be used. These techniques specially develop to result in a closed loop system with 25% of overshoot. Some step must be taken for tuning PID controller via the second method.[6]

- In the Second method, First set Ti = and Td = 0. Using the proportional control action only. Increase Kp from 0 to a critical value Ku where the output first exhibits sustained oscillation. (If the output does not exhibit sustain oscillation for whatever value of Kp may take, this method does not apply).[2][12]

- Using the ultimate gain Ku and the corresponding period Pu, Ziegler and Nichols suggested setting the values of the parameters Kp, Ti and Td according to the formula shown in Table.VI.

2.5 Modbus Protocol

2.5.1 Introduction

This is the first of a two-part series on Modbus. The first issue addresses the protocol itself. What do ARCNET, Ethernet and Modbus have in common? They were all developed in the 1970s and are still widely used today. Of course they have evolved

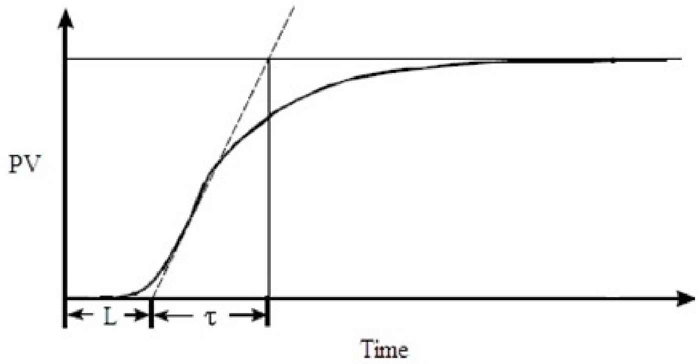


Figure 2.9: response Curve for Ziegler Nichols first method

Table V: Ziegler-Nichols First Method[1]

Controller Type	K_p	T_i	T_d
P	T/L	∞	-
PI	0.9 T/L	3.33 L	-
PID	1.2 T/L	2 L	0.5 L

over time, but their basic operation remains intact.[11]

There is one basic difference in the three technologies. Both ARCNET and Ethernet are data link and physical layer standards without a protocol while Modbus is a protocol that can operate over several data links and physical layers. Originally intended as a point-to-point interface between proprietary Modicon products, the protocol has found use in multi-drop and peer-to-peer networks like TCP/IP. It is no longer restricted to just Modicon equipment.

2.5.2 Implementation

It is interesting to note that Modicon did not use Modbus in a multi-drop network but instead used point-to-point connections with EIA-232C interfaces installed on their PLCs. The Modbus protocol is a master-slave protocol and the terms "master" and "slave" continue to be used today. Modbus allows only one master and up to 247 slaves. A slave is typically a Modicon PLC with an EIA-232C interface. Masters are typically programming panels or host computers. Therefore, if one host computer needed to communicate to four PLCs, four serial ports would be required on the host

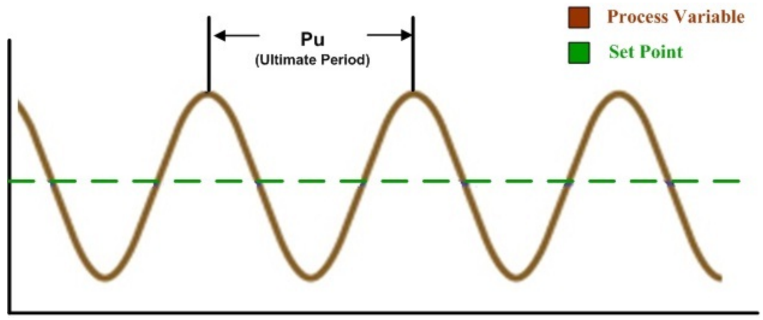


Figure 2.10: Sustained oscillation with Period Pu[5]

Table VI: Ziegler Nichols tuning rules based on Ultimate gain K_u and Period P_u[5]

Controller Type	K_p	T_i	T_d
P	$0.5K_u$	∞	0
PI	$0.45K_u$	$0.833P_u$	0
PID	$0.6K_u$	$0.5P_u$	$0.125P_u$

computer. This results in a star topology. EIA-232C cable lengths are short, so if longer distances are required modems can be used. It was not until later that 2-wire and 4-wire EIA-485 multi-drop networks appeared. With the Modbus protocol, only the master can initiate a message. Slaves cannot. So if a slave notices "that the cooling water pumps to the nuclear reactor have stopped," the slave cannot inform the master until the master happens to send a query to the slave with the effective message how are things going? The master has no address, but the slaves are numbered from 1 to 247.

Address "0" is reserved as a broadcast address to all slaves. All slaves will receive the broadcast message but will not respond.

QueryResponse messaging the command issued by the master is called a Query and the response from the slave is simply called the Response. The format in Figure.2.11 shows the simplified structure of the messages that can serve as either a query or a response.

The master has no address so the device address is always the intended slave. If it is a query, the query is directed to the slave with the assigned device address. If

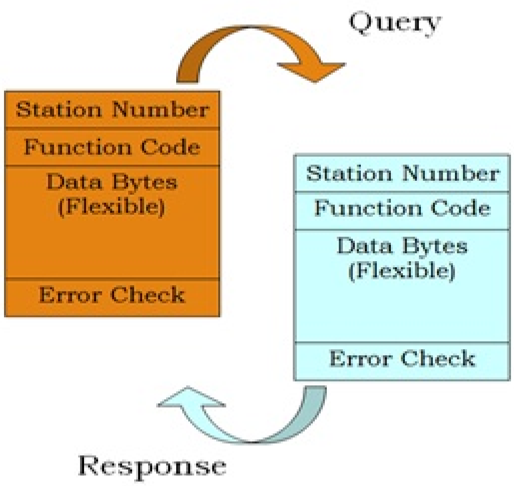


Figure 2.11: Simplified Modbus message format[11]

the message is a response, the response came from the slave with the indicated device address. Commands are issued by function codes such as 03Read Holding Registers. In this case the master must indicate the range of registers to be queried. The slave responds with the requested data based upon the indicated range. The message format is similar for all function codes, but of course the data changes based upon the code itself. After each message there is an error check appended by the originating station so that the receiving station can check the integrity of the received message.

The above scenario assumes a successful interchange of a query and a response. If the slave wants to communicate an error condition or an exception case, the function code is modified by the slave by setting the most-significant-bit (MSB) of the function code to a 1. The data field will then contain information specific to the exception. The master can still extract the original function code that it sent. It should be noted that the query-response cycle is completed before the master sends out the next query to either the same slave or another slave. This is unlike protocols such as DeviceNet that can send out one multicast command and then wait for several devices to response in no particular order to this one command.[11]

Modbus has no multicast capability so time is lost as each master query requires the directed slave to not only receive the message but act upon it and respond before the master moves on to other communications activity.

RTU Message Framing

When operating in RTU mode the timing is much more critical. There is no specific Start of Frame character. Instead, the message frame begins with four character times of marking. After this interval, the device address is sent followed by the function code and data. There are other differences from that of the ASCII message frame as noted in Figure.2.5.2. Instead of a Longitudinal Redundancy Check (LRC) check in the ASCII frame, a more robust Cyclic Redundancy Check (CRC) check of the data is applied in the RTU frame. The End of Frame indication is strictly based upon four character times of marking.

RTU messages must be sent as a continuous stream and any significant gaps between characters could result in a dropped message. Unlike ASCII, the RTU messages are not human readable. However, the messages are quite compact and more efficient to send. The RTU mode remains the more popular format.

Table 2.7 : RTU framing of a Modbus message[11]

Start of Frame	Device Address	Function Code	Data	CRC Check	End of Frame
4 Character times	8 bit	8 bit	n x 8 bits	16 bits	4 Character times

Modbus Register Map

Before we discuss function codes, we should study the Modbus register map in Table.VII since certain function codes imply specific register ranges. Early PLCs were mostly concerned with discrete inputs and discrete outputs each represented by one-bit in a register map. For Modicon PLCs, discrete outputs begin at location 00001 and discrete inputs begin at location 10001. Each requires one-bit of storage. Inputs, called contacts, can only be read and outputs, called coils, can be read or written. Since early PLCs were considered relay panel replacements, the terms coils and contacts were retained to assist electricians trying to understand these new electronic controllers.[11]

As the complexity of PLCs increased, the ability to handle analog input/output (I/O), and to execute math calculations was added. The I/O range of 16-bit register references begins at 30001 for read-only analogy inputs or thumb wheel switches, and 40001 for general purpose read/write registers that can also serve as analog outputs.

There are really no restrictions above 40001. Depending upon the vendor of the equipment, they can be internal registers, analogy inputs, analogy outputs, and even discrete inputs and outputs. However, not all function codes reference this rangebut enough do.

Table VII: Typical Modbus Register Map[11]

I/O Range	Description
00001 - 10000	Read / Write discrete output or coils
10001 - 20000	Read discrete inputs
30001 40000	Read input register 16 bit register such as analog input
40001 - 50000	Read/Write holding register 16 bit storage or I/O

Function Codes

The Modbus function codes are defined in both the Modicon Modbus Protocol Reference Guide and the Modbus Application Protocol Specification. Because there are differences in the function names and the number of function codes the latter document is recommended. Although the function code range spans from 1 to 127, only about 20 are defined public function codes. User defined function codes are allowed in specific locations within this range. However, many Modbus devices only support a small subset of the available codes. We will only examine those function codes that involve single-bit and 16-bit data access to get a flavour of how I/O is handled. A list is provided in Table.VIII.

Table VIII: Data access function codes[11]

Code	1/16-bit	Description	I/O Range
01	1-bit	Read coils	00001 - 10000
02	1-bit	Read contacts	10001 20000
05	1-bit	Write a Single coil	00001 10000
15	1-bit	Write multiple coil	00001 10000
03	16-bit	Read holding register	40001 50000
04	16-bit	Read input register	30001 40000
06	16-bit	Write single register	40001 50000
16	16-bit	Write multiple register	40001 50000
22	16-bit	Mask write register	40001 50000
23	16-bit	Read/write multiple register	40001 50000
24	16-bit	Read FIFO queue	40001 50000

You will notice from above Table.VIII that 1-bit function codes relate to discrete devices such as contacts and coils. The 16-bit function codes relate to input registers and holding registers. Input registers can only be read while holding registers can be either read or written. Also notice there is an implied I/O range depending upon the function code. For example, function code 06Write single register, only addresses the relevant range of 4000150000 and no other range. Therefore it is only necessary to reference the offset from the base range when structuring the message.

Instead of indicating register location 40001 we simply say 0000. This is a good time to explain one of the more confusing aspects of Modbus and that is referencing I/O points in Modbus messages. Modicon elected to number physical points within a range beginning with the number 1 instead of 0. Coil 1 is referenced in a message as location 0000 and not 00001. Likewise, Contact 1 is referenced as 0000 instead of 10001. The same applies to holding register 40001. It is also referenced as 0000. The function code always points to the proper I/O range and only the offset from base address of that range is needed to uniquely identify the point. The offset is a 16-bit word and is displayed in hexadecimal when examining the actual Modbus message. All references in the Modbus Register Map are decimal references. Register 40016 is referenced as 0x000F which is hexadecimal for 4001640001. Although this is confusing at first, it is only an issue for those writing Modbus drivers.

In this chapter, discuss about the basic theory of PID Controller and its parameter's effect on closed loop system. Also see that different control mode of controller and types of different structure. Discuss about some PID auto tuning methods and see that how to find out constant parameters. For communication between PC and PID controller, study a modbus protocol and its frame format.

Chapter 3

Methodology

3.1 Introduction

This Chapter will describe the design and implementation for the PID controller system for industrial use. This system function as one of the safety features in avoiding overheating. This project consists of the implementation of hardware and software. The system prototype will be tuned and design to achieve standard temperature for optimal electronic and electric operation within 40°C to 70°C.

3.1.1 PID Controller Block Diagram

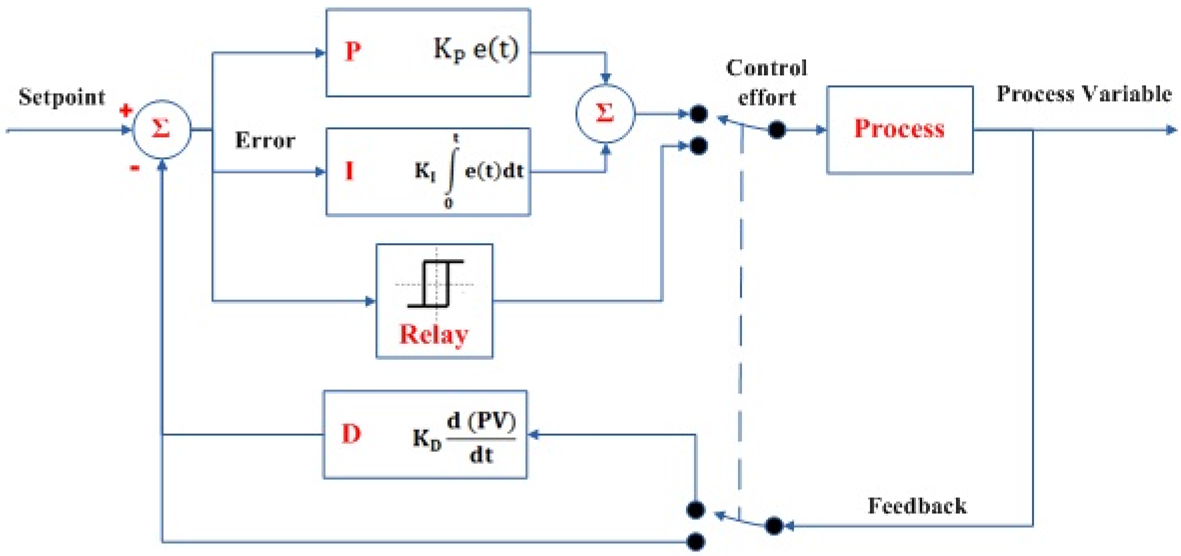


Figure 3.1: Modified PID Controller block diagram with derivative on process value instead of error

The Figure.3.1 shows the PID controller block diagram which used in the systems. In this block diagram shows the switching circuit. At the first time when PID Controller start the auto tune at that time relay will be on picture and working as an on-off controller with unity feedback closed loop system.

After Auto Tune Complete PID Controller Algorithm Find the Parameters PB (Proportional Band), Ti (Integral Time Constant), Td (Derivative Time Constant) according to heater characterises. In this block diagram shows some modification in PID structure which is not regularly used in Industries. Basically here derivative done on process value instead of error. This is the main research work to develop a system prototype. The reason behind this is a derivative kick which gives you in detail after this.[7][8]

3.1.2 Reason for derivative on PV

The proportional term considers how far PV is from SP at any instant in time. Its contribution to the CO is based on the size of e(t) only at time t. As e(t) grows or shrinks, the influence of the proportional term grows or shrinks immediately and proportionately.[9]

The integral term addresses how long and how far PV has been away from SP. The integral term is continually summing e(t). Thus, even a small error, if it persists, will have a sum total that grows over time and the influence of the integral term will similarly grow.[10]

A derivative describes how steep a curve is. More properly, a derivative describes the slope or the rate of change of a signal trace at a particular point in time. Accordingly, the derivative term in the PID equation above considers how fast, or the rate at which, error (or PV as we discuss next) is changing at the current moment.

While the proportional and integral terms of the PID equation are driven by the controller error, e(t), the derivative computation in many commercial implementations should be based on the value of PV itself.[9][10]

The derivative of e(t) is mathematically identical to the negative of the derivative of PV everywhere except when set point changes. And when set point changes, derivative on error results in an undesirable control action called derivative kick.

The mathematical defence that derivative of e (t) equals the negative derivative of PV when SP is constant considers that, since e(t) = SP PV, the equation below follows. That is, derivative of error equals derivative of set point minus process variable.

The derivative of a constant is zero, so when SP is constant, mathematically, the derivative (or slope or rate of change) of the controller error equals the derivative (or slope or rate of change) of the measured process variable, PV, except the sign is opposite.[10]

$$\frac{de(t)}{dt} = \frac{d(\overset{0}{\cancel{SP}} - PV)}{dt} = \frac{-dPV}{dt} \dots \dots \dots [3.1]$$

The Figures.3.1.2 below provide a visual appreciation that the derivative of e(t) is the negative of the derivative of PV.

The top plot shows the measured PV trace after a set point step. The bottom plot shows the e(t) = SP PV trace for the same event.[9]

If compare the two plots after the SP step at time t = 10, we see that the PV trace in the upper plot is an exact reflection of the e(t) trace in the lower plot. The PV trace ascends peaks and then settles, while in a reflected pattern, the e(t) trace descends, dips and then settles.

Mathematically, this of trace shapes means that the derivatives (or slopes or rates of change) are the same everywhere after the SP step, except they are opposite in sign.

While the shape of e(t) and PV are opposite but equal everywhere after the set point step, there is an important difference at the moment the SP changes. The lower plot

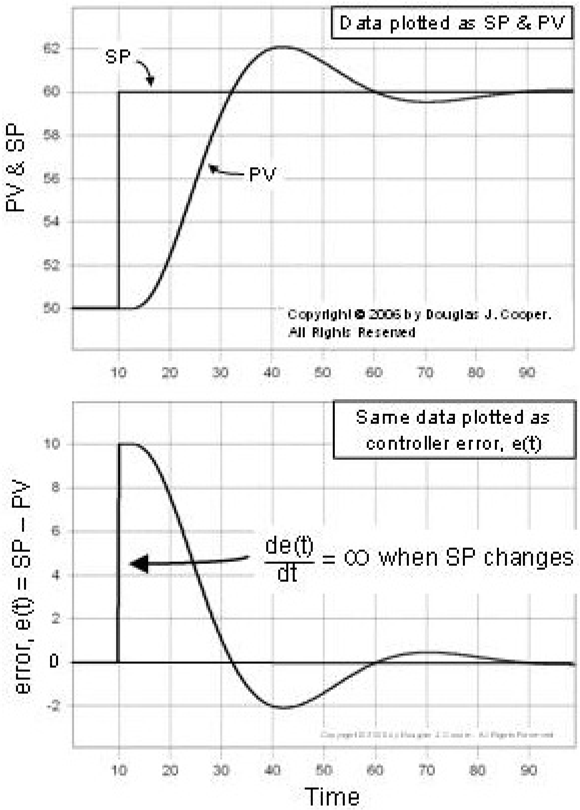


Figure 3.2: Plot for derivative on error[13]

shows a vertical spike in e(t) at this moment. There is no corresponding spike in the PV plot.

The derivative (or slope) of a vertical spike in the theoretical world approaches infinity. In the real world it is at least a very big number. If Td is large enough to provide any meaningful weight to the derivative term, this huge derivative value will cause a large and sudden manipulation in CO. This large manipulation in CO, referred to as derivative kick, is almost always undesirable. As long as loop sample time T, is properly specified, the PV trace will follow a gradual and continuous response, avoiding the dramatic vertical spike evident in the e(t) trace.[10]

Because derivative on e(t) is identical to derivative on PV at all times except when the SP changes, and when the set point does change, derivative on error provides information we do not want our controller to use, we substitute the equation in the above to obtain the PID with derivative on measurement controller.[9]

$$u(t) = K_p * e(t) + K_i * \int e(\tau)d(\tau) + K_d * (-\frac{dPV}{dt}) \ldots \ldots \ldots [3.2]$$

A rapidly changing PV has a steep slope and this yields a large derivative. This is true regardless of whether a dynamic event has just begun or if it has been underway for some time. In the plot below, the derivative dPV/dt describes the slope or "steepness" of PV during a process response.

Early in the response, the slope is large and positive when the PV trace is increasing rapidly. When PV is decreasing, the derivative (slope) is negative. And when the PV goes through a peak or a trough, there is a moment in time when the derivative is zero.

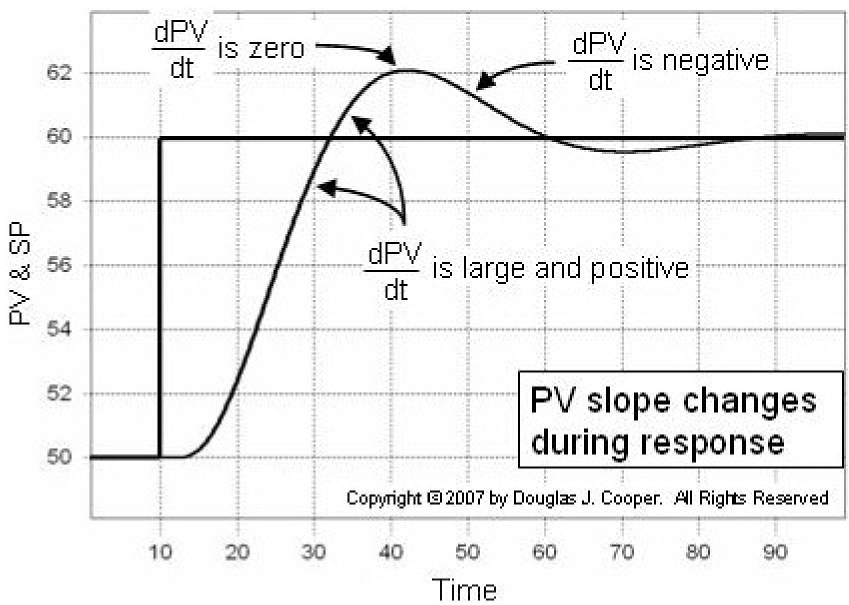


Figure 3.3: Plot for Process Variable on derivative[13]

To understand the impact of this changing derivative, let's assume for discussion that:[10]

- Controller gain, Kc, is positive.
- Derivative time, Td (always positive) is large enough to provide meaningful weight to the derivative term. After all, if Td is very small, the derivative term has little influence, regardless of the slope of the PV.

The negative sign in front of the derivative term of the PID with derivative on measurement controller (and given the above assumptions) means that the impact on CO from the derivative term will be opposite to the sign of the slope:

$$u(t) = K_p * e(t) + K_i * \int e(\tau)d(\tau) + K_d * (-\frac{dPV}{dt}) \ldots \ldots \ldots [3.3]$$

Thus, when dPV/dt is large and positive, the derivative term has a large influence and seeks to decrease CO. Conversely, when dPV/dt is negative, the derivative term seeks to increase CO. It is interesting to note that the derivative term does not consider whether PV is heading toward or away from the set point (whether e(t) is positive or negative). The only consideration is whether PV is heading up or down and how quickly.

The result is that derivative action seeks to inhibit rapid movements in the PV. This could be an especially useful characteristic when seeking to dampen the oscillations in PV that integral action tends to magnify. Unfortunately, as we will discuss, the potential benefit comes with a price.

3.2 Proposed Modified Relay based Auto Tuning Method

3.2.1 Basic Principle of Relay feedback method

A Proportional-integral-derivative (PID) controller operating in a feed-back can be very effective at driving a measured process variable towards a desired set point, but only if the controller is adjusted or tuned to accommodate the behaviour of the controlled process. For a PID controller using the ISA standard form of the PID formula, tuning is a matter of selecting appropriate values for the tuning parameter P, Ti, and Td.[7][8]

Tuning the loop-or more correctly, tuning the controller requires adjusting those weights so that the controller can eliminate error quickly without making matters worse. If the controlled process happens to be relatively sluggish, the controller can

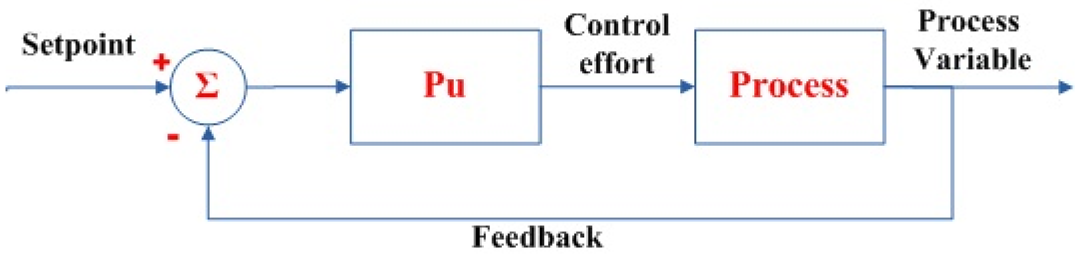


Figure 3.4: Closed loop system with only proportional controller[5]

be tuned to make immediate and dramatic action when-ever a random disturbance changes the process variable or an operator changes the set point. A relatively large value for Kp and a small value for Ti would be appropriate in this case.[14]

Conversely, if the process happens to be particularly sensitive to the controller's efforts, the loop must be tuned to apply more conservative corrective efforts over a longer interval. The essence of loop tuning is identifying just how dramatically the process reacts to the controller's efforts and how aggressive the controller can afford to be as it tries to eliminate errors.[8]

Unfortunately, Loop tuning is not as consulting a table to find a value of Kp that produces a desired degree of overshoot or a specific settling time. The best choice for each of the tuning parameters Kp, Ti and Td depends on the values of the other two as well as the behaviour of the controlled process. Furthermore, modifying the tuning of my one term affects the performance of the other two since the modified controller affects the process and the process in turn affects the controller.

3.2.2 Auto-Tuning based on Ziegler-Nichols Method

John G. Ziegler and Nathaniel B. Nichols of Taylor Instruments (now part of ABB) addressed these problems in 1942 when they proposed a two-steps method for tuning a loop. They devised tests for qualifying the behaviour of a process in terms of how fast and how much the process variable changes when the control effort changes. They also developed a set of empirical formulas for translating the results of those tests into appropriate settings for the controller's tuning parameters.[7]

The Ziegler-Nichols closed-loop tuning test is conducted with the controller oper-

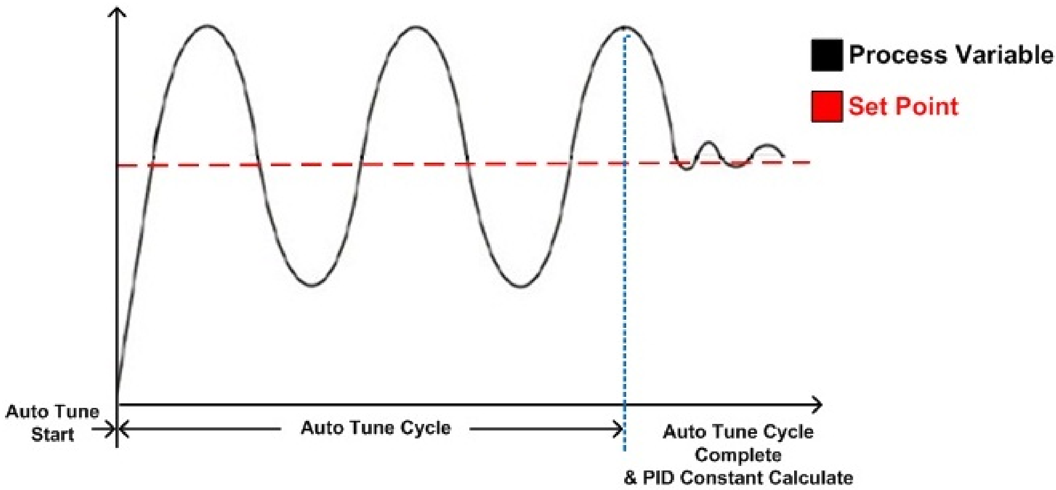


Figure 3.5: Auto tune cycle of PID Controller

ating in automatic mode but with the integral and derivative terms temporarily shut off. The controller gain is increased until even the slightest error causes a sustained sinusoidal oscillation in the process variable as shown in below Figure3.6.

The smallest controller gain that can cause such an oscillation is called the ultimate gain Pu. The period of those sinusoidal oscillations is called the ultimate period Tu. The appropriate tuning parameters are computed from these two values according to the Ziegler-Nichols tuning rules as shown below Table.I.

Table I: Ziegler Nichols tuning rules based on Ultimate gain K_u and Period P_u[14]

Controller Type	K_p	T_i	T_d
P	$0.5K_u$	∞	0
PI	$0.45K_u$	$0.833P_u$	0
PID	$0.6K_u$	$0.5P_u$	$0.125P_u$

3.2.3 Tuning process for PID Controller

Then in 1984, Karl Astrom and Tore Hagglund of the Lund (Sweden) Institute of Technology proposed a less risky alternative to the Ziegler-Nichols open-Loop test. Their relay method generates a sustained oscillation restricted to a safe range.

The Astrom-Hugglund method works by forcing the process variable into a limit cycles shown in the ”relay test” graphic. With all three PID terms temporarily disabled, the

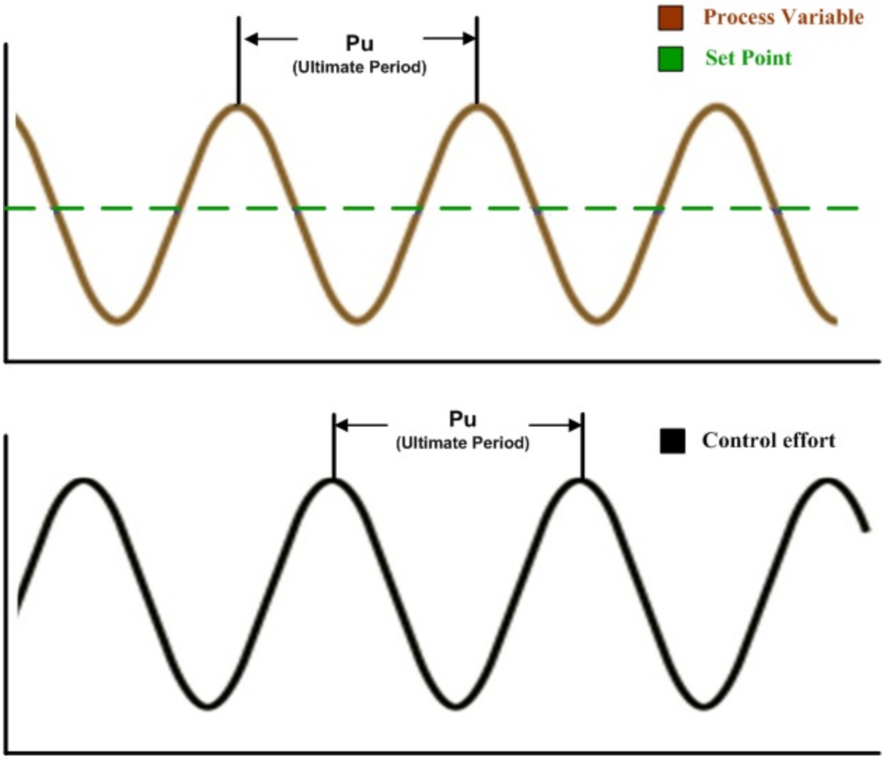


Figure 3.6: Sustained oscillation of PV and control effort with ultimate period Pu[5]

controller uses an on/off relay to apply a step-like control effort to the process. It then holds the control effort contestant and waits for the process variable to exceed the set point.

At that point, it applies a negative step and waits for the process variable to drop back below the set point. Repeating this procedure each time the process variables passes the set point in either direction forces the process variable to oscillate out of sync with the control effort but at the same frequency.[10][17]

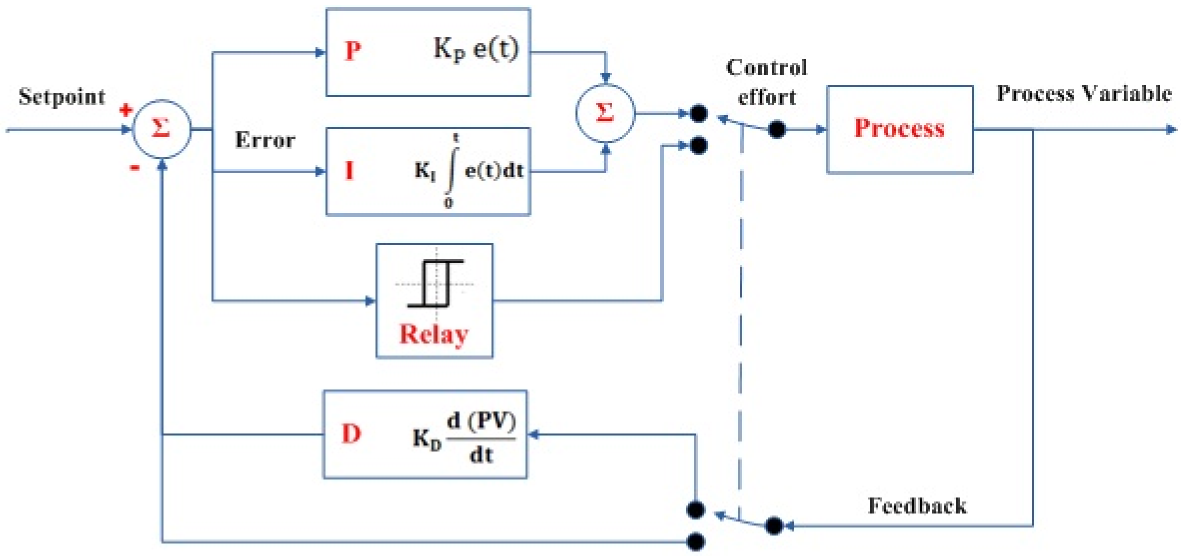


Figure 3.7: PID Controller block diagram with modified relay feedback approach

Although the process variables oscillations are not strictly sinusoidal, their period turns out to be a close approximation of the ultimate period that Ziegler and Nichols used for their tuning rules. And the amplitude of the process variable's oscillations relative to the amplitude of the control effort's oscillations approximates the process's ultimate gain when multiplied by $4/\prod$. So once the ultimate period and ultimate gain have been determined based on below formulas:[5][13]

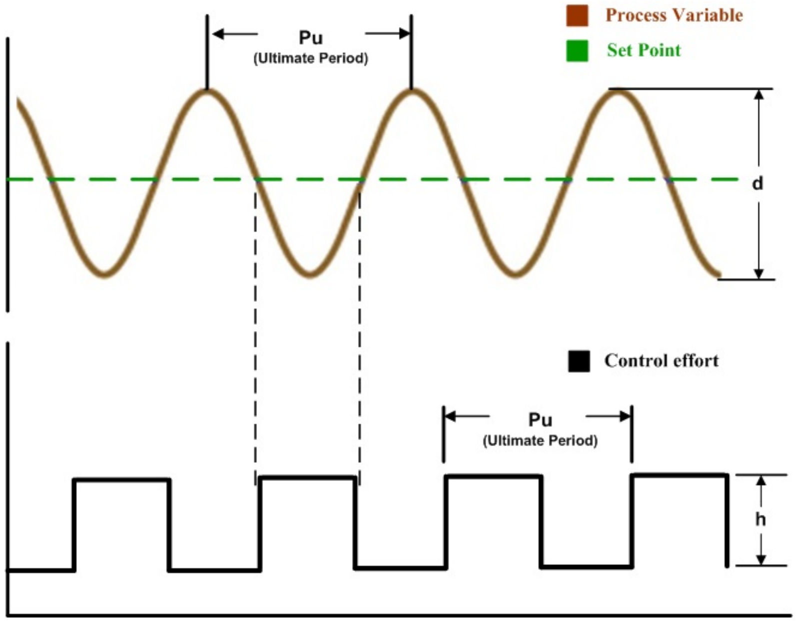


Figure 3.8: from sustained oscillation graph finding parameters for ultimate gain[5]

$$\text{First Order model equation } G(s) = \frac{K_c\, e^{-\theta s}}{\tau s+1} \dots\dots\dots [3.4]$$

$$\text{Ultimate Gain } K_U = \frac{4h}{\pi d} \dots\dots\dots [3.5]$$

$$\text{Ultimate Frequency } \omega_U = \frac{2\pi}{P_U} \dots\dots\dots [3.6]$$

And finally put all calculated above value in below indicate PID equation in the form of Laplace domain.[8][10]

$$u_k = u_{k-1} + K_p[PV - SP + \left(\frac{PB}{2}\right)] + K_i\, e_k + K_d[PV - 2e_{k-1} + e_{k-2}] \dots\dots\dots [3.7]$$

Where,

$u_k = CurrentPIDOutput$

$u_{k-1} = PreviousPIDOutput$

$SP = SetPoint$

$PV = ProcessValue$

Table II: Modified PID Parameter calculation based on Ultimate gain and ultimate period

Specification	K_p	T_i	T_d
Original	$0.6K_u$	$0.5P_u$	$0.125P_u$
Modified	$0.25K_u$	$0.833P_u$	$0.208P_u$

$PB = ProportionalBand$

$K_p = Proportionalgain$

$K_i = Integralgain$

$K_d = Derivativegain$

$e_k = currenterror$

$e_{k-1} = previouserror$

$e_{k-2} = lastpreviouserror$

3.2.4 Advantage

But unlike the original Ziegler-Nichols closed-loop test, the relay test can be configured to limit the amplitude of the process variable's oscillation by fixing the amplitude of the control effort's oscillation at a user-defined value. This allows the controller to force the process variable to oscillate just enough to distinguish the process's behaviour from measurement noise. The process variable needs not swing so wildly as to endanger the process.[7]

Better still, the controller can be configured to conduct the relay test and tune the loop without operator intervention. Theoretically, even an operator unfamiliar with the fundamentals of tuning theory can press a button and let the controller conduct its own relay test and select its own tuning parameters accordingly. If the resulting closed-loop behaviour proves unacceptable, the operator can simply push the tune button again.[9]

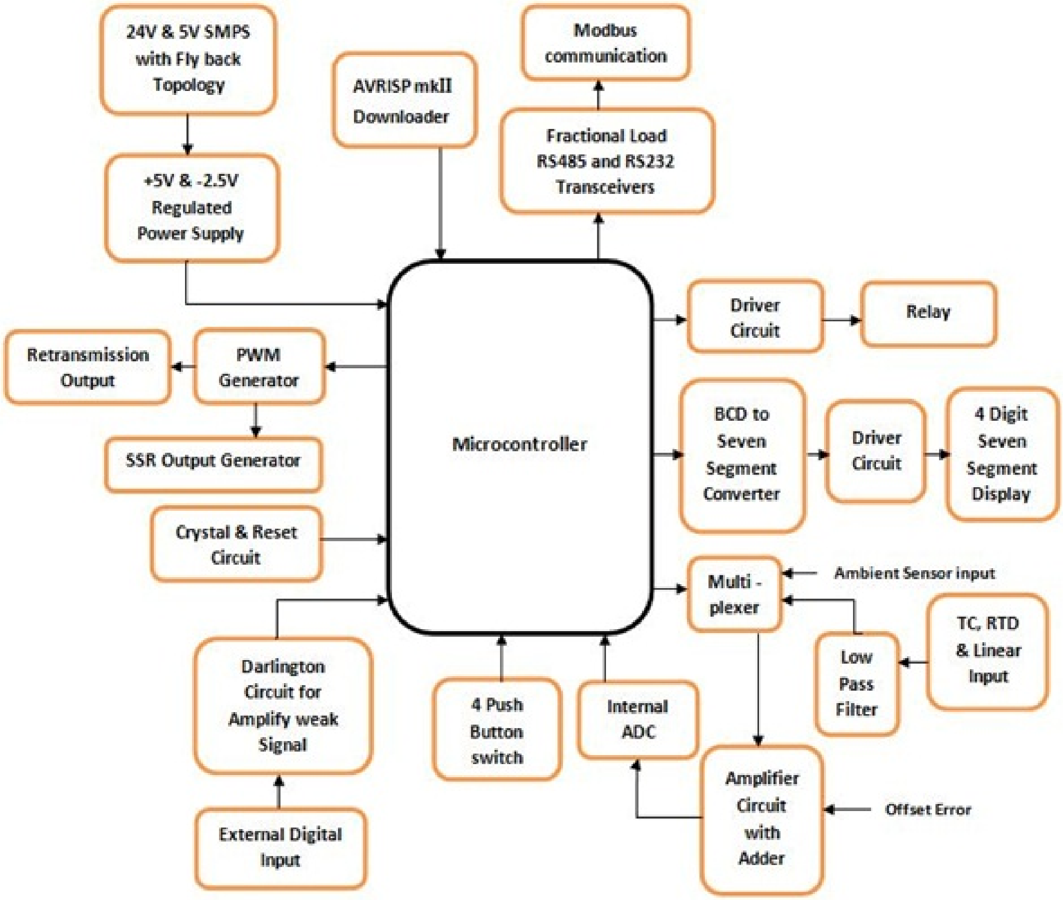


Figure 3.9: schematic diagram of the PID Controller

3.3 Hardware Schematic Block Diagram

3.3.1 SMPS Section

A switched-mode power supply (switching-mode power supply, SMPS, or switcher) is an electronic power supply that incorporates a switching regulator to convert electrical power efficiently. Like other power supplies, an SMPS transfers power from a source, like mains power, to a load, such as a personal computer, while converting voltage and current characteristics. An SMPS is usually employed to efficiently provide a regulated output voltage, typically at a level different from the input voltage.

Unlike a linear power supply, the pass transistor of a switching-mode supply continually switches between low-dissipation, full-on and full-off states, and spends very little time in the high dissipation transitions (which minimizes wasted energy).

Ideally, a switched-mode power supply dissipates no power. Voltage regulation is achieved by varying the ratio of on-to-off time. In contrast, a linear power supply regulates the output voltage by continually dissipating power in the pass transistor. This higher power conversion efficiency is an important advantage of a switched-mode power supply. Switched-mode power supplies may also be substantially smaller and lighter than a linear supply due to the smaller transformer size and weight. Switching regulators are used as replacements for the linear regulators when higher efficiency, smaller size or lighter weight is required. They are, however, more complicated; their switching currents can cause electrical noise problems if not carefully suppressed, and simple designs may have a poor power factor.

In an SMPS, the output current flow depends on the input power signal, the storage elements and circuit topologies used, and also on the pattern used (e.g., pulse-width modulation with an adjustable duty cycle) to drive the switching elements. Typically, the spectral density of these switching waveforms has energy concentrated at relatively high frequencies. As such, switching transients, like ripple, introduced onto the output waveforms can be filtered with small LC filters. In SMPS of PID Controller the line voltage is first pass through the surge protector. The surge protector removes the surge or any kind of noise which is coming from the mains line. After that it passes from the bridge rectifier for converting it into AC to DC. Next stage will have the PI filter and the capacitor filter to make it to constant DC voltage. The reference diode is kept after the capacitor for regulated output. The TNY-275 is a PWM Controller for switching the DC power supply. It is given to the switching transformer and the outputs of the switching transformer are 4 regulated DC output. 5V, 12V, 24V and +5V isolated are taken as a output. The outputs are given to the all other cards via connectors.

3.3.2 CPU & Display Section

CPU unit section contains micro controller AVR ATmega128 and other drive circuitry for it. The 16 MHz crystal is used for giving clock pulse to the ATmega128. The inputs from the signal conditioning are given to the ATmega128 ADC pin. The ambient temperature generation input is also given to the micro controller. The output for transmission to PC and input from PC is received in the micro controller through UART pins.

Display section contains 7 Segment displays of 0.56 inch and four digits. It also contains four keys and its driver circuitry including 3 to 8 decoder and driver transistors. It also includes 6 LEDs for different indications. I.e. Relay-1 Relay-2, TX, Rx, SSR, AT. Four switches are used for different functions. I.e. INC, DEC, FUN, SET. These switches are used as operating part of the device.

3.3.3 Signal Conditioning Section

The inputs from the back plates are given to the signal conditioning section. In this signal conditioning the 8X1 MUX is used for selecting the inputs given to it. The selection line inputs are given from the micro controller pin. The power supply is given to the signal conditioning card is from SMPS card. After that the voltage buffer OPAMP is used for voltage buffering. And it is followed by the gain OPAMP. The final output is given to the micro controller for further procedure. The ambient voltage is also generated in the signal conditioning card. -2.5 volt is also generated in this section. +2.5 volt is also generated in this section. RTD and thermocouple inputs are given to the multiplexer after processing with the gain and other controlling circuitry. It is accurate input section for micro controller.

3.3.4 Communication Section

The communication section is very useful section for transmission to PC. The UART pins are connected to the DC-DC converter of the communication card. +5V Isolated is given from the PS card. The DC-DC converter output is given then to the RS-485 Transceiver IC. After that the drive circuits are used with OPAMP and other components. The output is given then to the back plate. This output is driven then to the PC. The RS-485 to RS-232 converter is used for communicating with the PC. MODBUS Tester software is used for data read and write to and from the device.

3.4 Software Flow Chart

In this section explain that how PID controller software is working step by step. It have so many feature include like alarm, retransmission output etc...

The software explanation is in the form of flow chart. First part shows the main functionality menu of the PID Controller.

Second part shows the internal ADC functionality, ambient temperature calculation, power on display and watch dog timer initialisation.

Third part shows the functionality for alarm, retransmission output 1 & 2 and check routine for debauching & detection key.

Forth part include the routine for PID controller & On-Off controller. Software flow chart is explain as given below:

3.5 Modbus Communication Tools

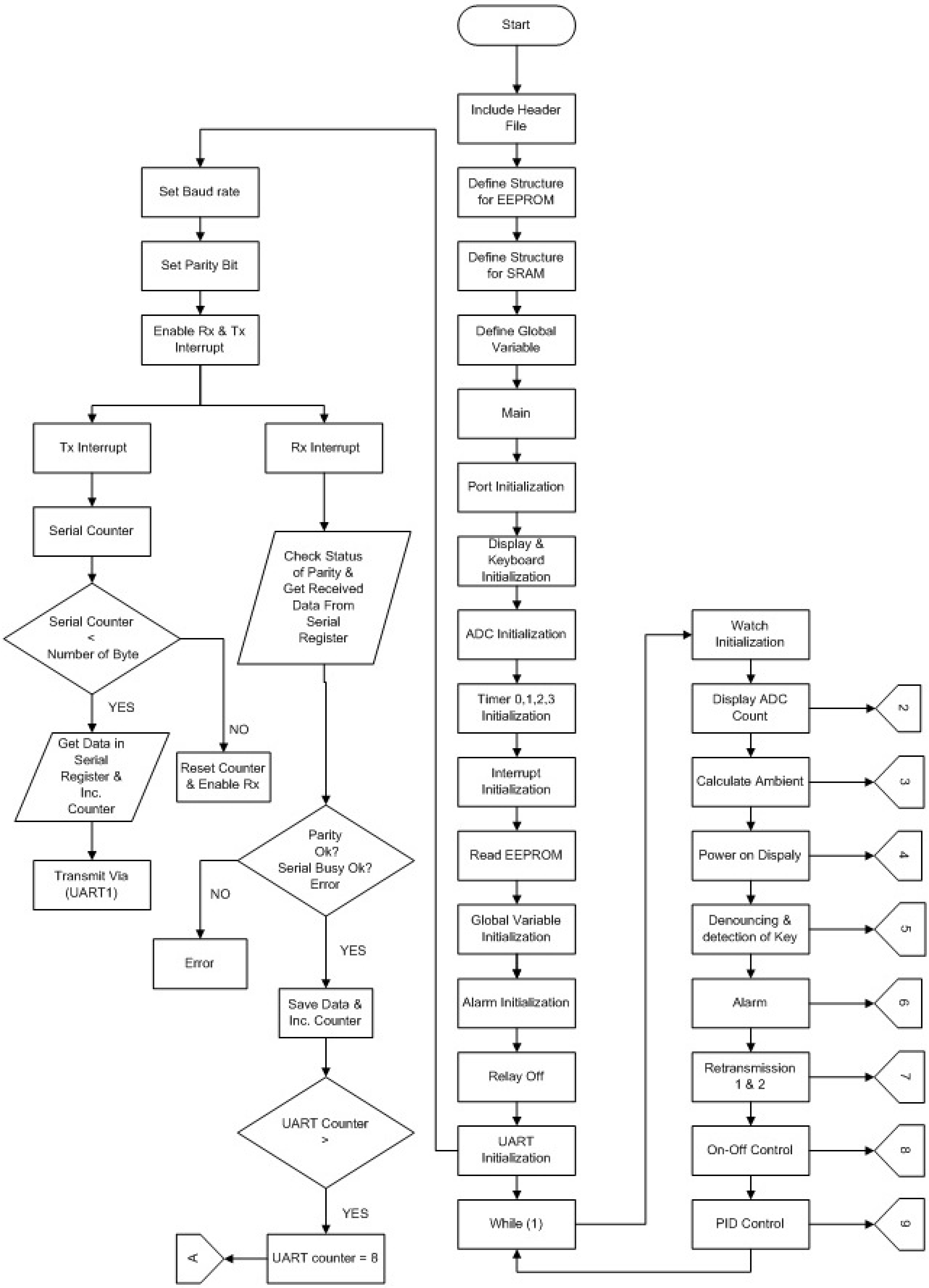
Modbus Protocol is used for communication between PID Controller and PC. For that mostly Modbus Tester Software is used in industries and it is free ware software so any one can use without paying any cost. In PC, Modbus Tester Software behaves as a Master and it can communicate other 256 slave device simultaneously.[11]

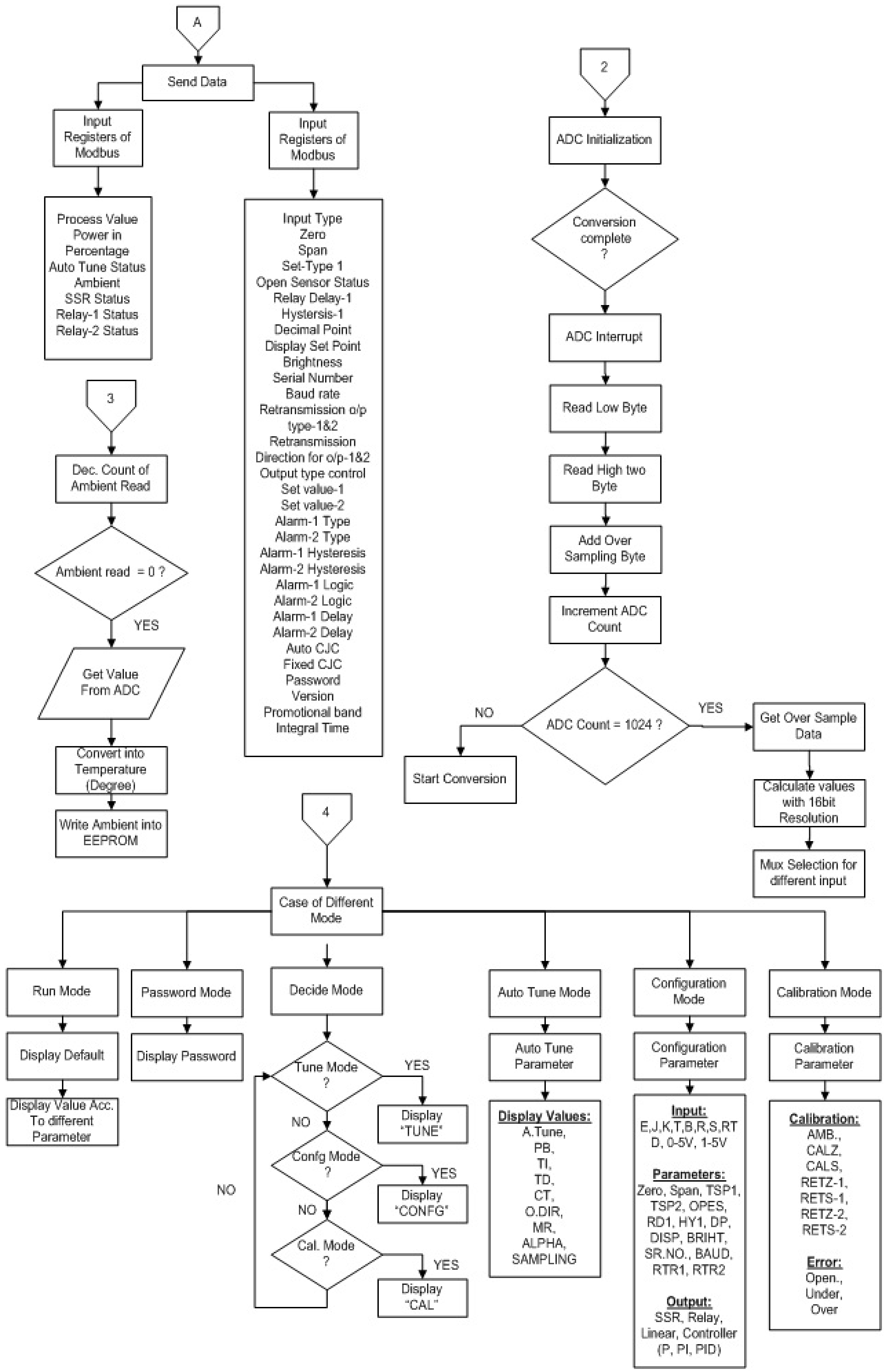
In this chapter, shows both screen shot of the modbus tester software dialog box. In capture, it shows COM Port settings, baud rate, parity, stop bits settings and other screen shot shows how to select device address, data address, starting address and length of the number of parameters and data format. It also shows the current status of the communication.[11]

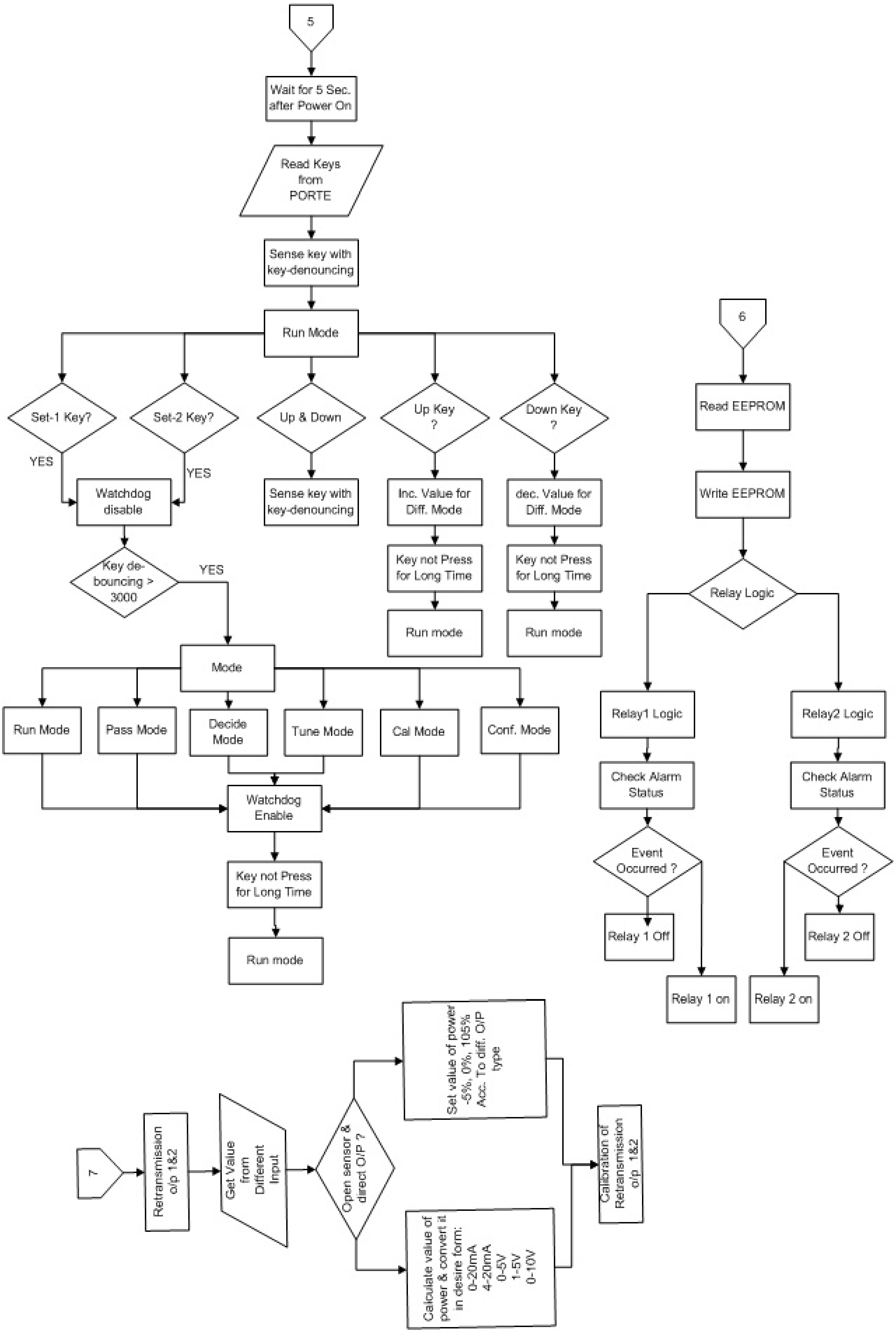
3.6 Experimental Setup

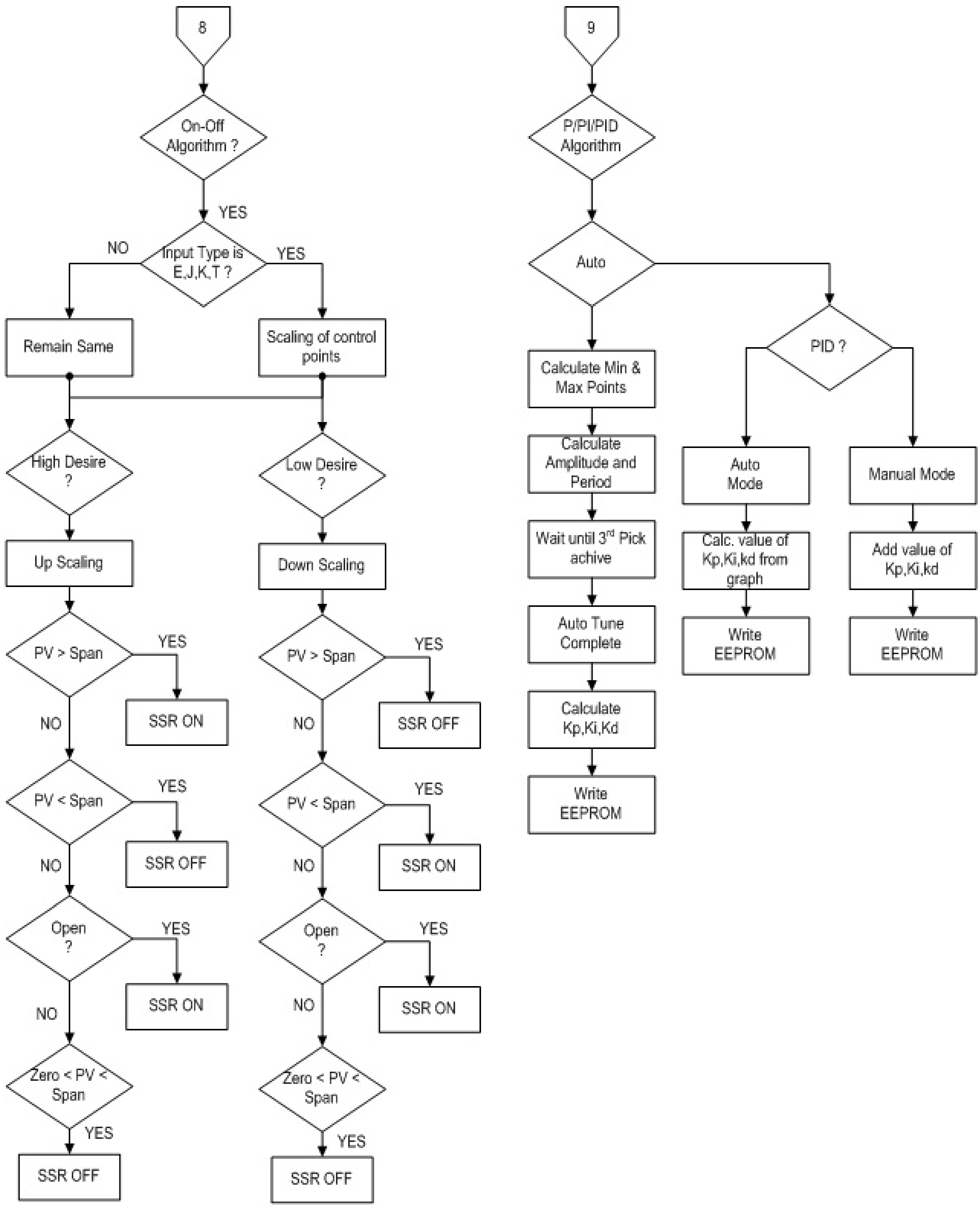
Here shows some snapshots about experimental setup of modbus communication between PID controller and PID Controller, contactor testing zig which consists of heater and final snap shot of whole experimental setup.

In this chapter summarised about the implementation of proposed modified relay based auto tuning techniques. Discuss about modification of derivative on PV instead of error and its reason also. Proposed method discuss in detail with equation and graphs. Discuss about Hardware block diagram of PID Controller and software flow chart also. Here shows some snap shots of modbus tester software dialog box and also experimental setup.


Start
Include Header File
Define Structure for EEPROM
Define Structure for SRAM
Define Global Variable
Main
Port Initialization
Display & Keyboard Initialization
ADC Initialization
Timer 0,1,2,3 Initialization
Interrupt Initialization
Read EEPROM
Global Variable Initialization
Alarm Initialization
Relay Off
UART Initialization
While (1)
Set Baud rate
Set Parity Bit
Enable Rx & Tx Interrupt
Tx Interrupt
Rx Interrupt
Serial Counter
Serial Counter < Number of Byte
YES
NO
Get Data in Serial Register & Inc. Counter
Reset Counter & Enable Rx
Transmit Via (UART1)
Check Status of Parity & Get Received Data From Serial Register
Parity Ok? Serial Busy Ok? Error
NO
Error
YES
Save Data & Inc. Counter
UART Counter >
YES
UART counter = 8
A
Watch Initialization
Display ADC Count
Calculate Ambient
Power on Dispaly
Denouncing & detection of Key
Alarm
Retransmission 1 & 2
On-Off Control
PID Control
2
3
4
5
6
7
8
9


A
Send Data
Input Registers of Modbus
Input Registers of Modbus
Process Value
Power in Percentage
Auto Tune Status
Ambient
SSR Status
Relay-1 Status
Relay-2 Status
Input Type
Zero
Span
Set-Type 1
Open Sensor Status
Relay Delay-1
Hystersis-1
Decimal Point
Display Set Point
Brightness
Serial Number
Baud rate
Retransmission o/p type-1&2
Retransmission Direction for o/p-1&2
Output type control
Set value-1
Set value-2
Alarm-1 Type
Alarm-2 Type
Alarm-1 Hysteresis
Alarm-2 Hysteresis
Alarm-1 Logic
Alarm-2 Logic
Alarm-1 Delay
Alarm-2 Delay
Auto CJC
Fixed CJC
Password
Version
Promotional band
Integral Time
2
ADC Initialization
Conversion complete ?
ADC Interrupt
Read Low Byte
Read High two Byte
Add Over Sampling Byte
Increment ADC Count
ADC Count = 1024 ?
NO
YES
Start Conversion
Get Over Sample Data
Calculate values with 16bit Resolution
Mux Selection for different input
3
Dec. Count of Ambient Read
Ambient read = 0 ?
YES
Get Value From ADC
Convert into Temperature (Degree)
Write Ambient into EEPROM
4
Case of Different Mode
Run Mode
Display Default
Display Value Acc. To different Parameter
Password Mode
Display Password
Decide Mode
Tune Mode ?
YES
Display "TUNE"
NO
Confg Mode ?
YES
Display "CONFG"
NO
Cal. Mode ?
YES
Display "CAL"
NO
Auto Tune Mode
Auto Tune Parameter
Display Values:
A.Tune, PB, TI, TD, CT, O.DIR, MR, ALPHA, SAMPLING
Configuration Mode
Configuration Parameter
Input:
E,J,K,T,B,R,S,RTD, 0-5V, 1-5V
Parameters:
Zero, Span, TSP1, TSP2, OPES, RD1, HY1, DP, DISP, BRIHT, SR.NO., BAUD, RTR1, RTR2
Output:
SSR, Relay, Linear, Controller (P, PI, PID)
Calibration Mode
Calibration Parameter
Calibration:
AMB., CALZ, CALS, RETZ-1, RETS-1, RETZ-2, RETS-2
Error:
Open., Under, Over


5
Wait for 5 Sec. after Power On
Read Keys from PORTE
Sense key with key-denouncing
Run Mode
Set-1 Key?
Set-2 Key?
Up & Down
Up Key ?
Down Key ?
YES
YES
Watchdog disable
Sense key with key-denouncing
Inc. Value for Diff. Mode
dec. Value for Diff. Mode
Key not Press for Long Time
Key not Press for Long Time
Run mode
Run mode
Key de-bouncing > 3000
YES
Mode
Run Mode
Pass Mode
Decide Mode
Tune Mode
Cal Mode
Conf. Mode
Watchdog Enable
Key not Press for Long Time
Run mode
6
Read EEPROM
Write EEPROM
Relay Logic
Relay1 Logic
Relay2 Logic
Check Alarm Status
Check Alarm Status
Event Occurred ?
Event Occurred ?
Relay 1 Off
Relay 2 Off
Relay 1 on
Relay 2 on
7
Retransmission o/p 1&2
Get Value from Different Input
Open sensor & direct O/P ?
Set value of power -5%, 0%, 105% Acc. To diff. O/P type
Calculate value of power & convert it in desire form: 0-20mA 4-20mA 0-5V 1-5V 0-10V
Calibration of Retransmission o/p 1&2


8
On-Off Algorithm ?
YES
Input Type is E,J,K,T ?
NO
YES
Remain Same
Scaling of control points
High Desire ?
Low Desire ?
Up Scaling
Down Scaling
PV > Span
YES
SSR ON
NO
PV < Span
YES
SSR OFF
NO
Open ?
YES
SSR ON
NO
Zero < PV < Span
YES
SSR OFF
PV > Span
YES
SSR OFF
NO
PV < Span
YES
SSR ON
NO
Open ?
YES
SSR ON
NO
Zero < PV < Span
YES
SSR OFF
9
P/PI/PID Algorithm
Auto
PID ?
Calculate Min & Max Points
Calculate Amplitude and Period
Wait until 3rd Pick achive
Auto Tune Complete
Calculate Kp,Ki,Kd
Write EEPROM
Auto Mode
Calc. value of Kp,Ki,kd from graph
Write EEPROM
Manual Mode
Add value of Kp,Ki,kd
Write EEPROM

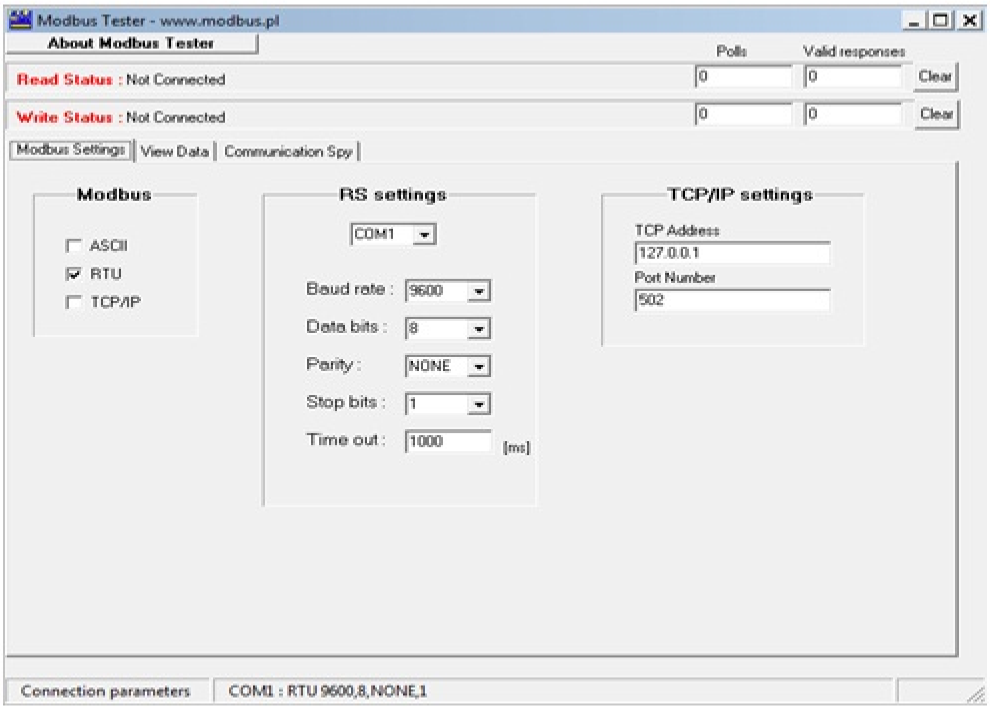


Figure 3.10: Modbus settings dialog box of Modbus Tester Software

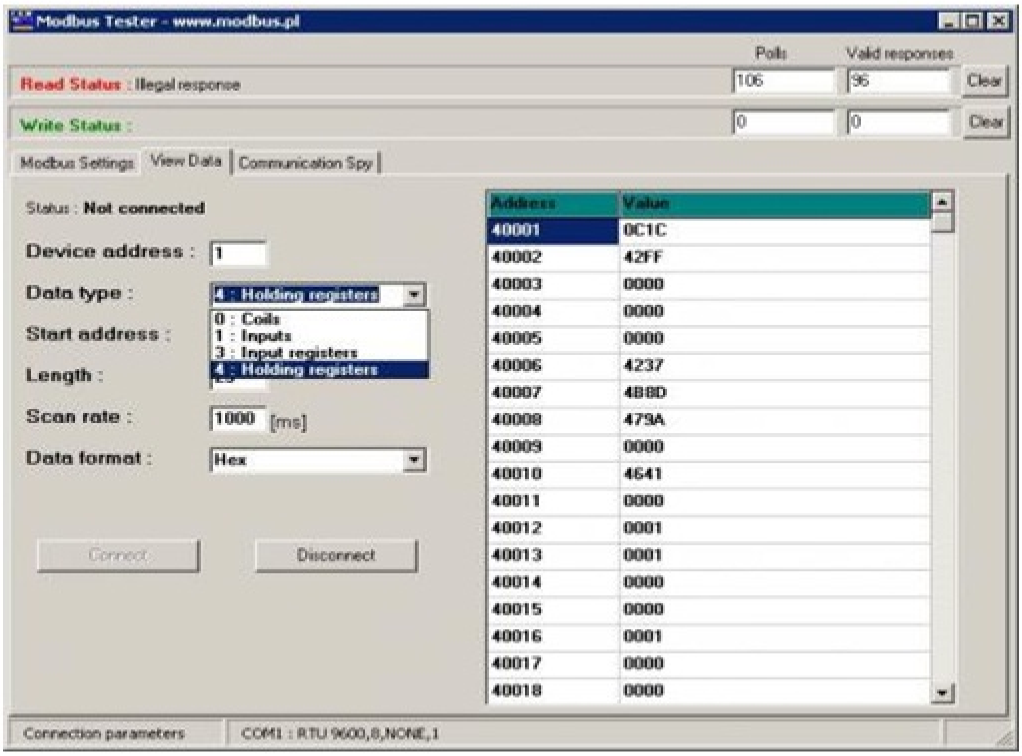


Figure 3.11: Modbus Parameter dialog box of Modbus Tester Software

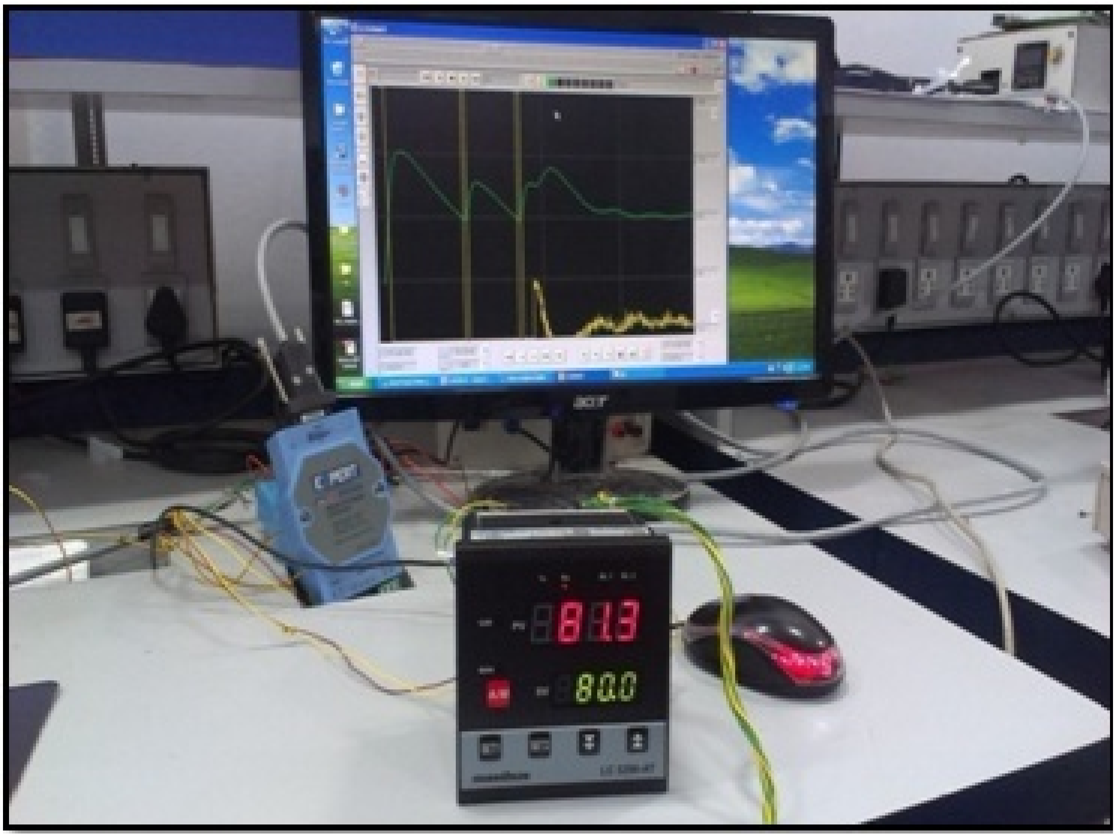


Figure 3.12: Snapshot for Modbus communication with PID controller

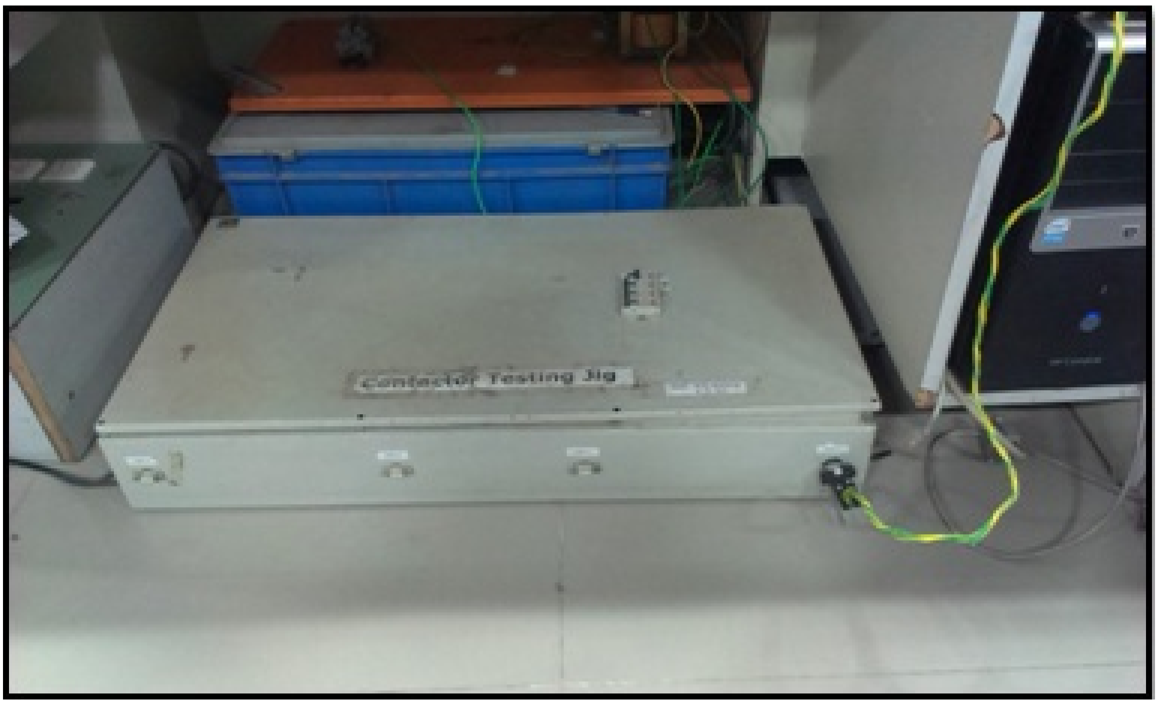


Figure 3.13: Snapshot of contector testing zig consist of heater for Industrial Process test

Figure 3.14: Snapshot of whole hardware experiment setup

Chapter 4

Results and Discussion

4.1 Experimental Results

The prototype of PID temperature controller was able to detect, monitor and control the temperature and able to maintain safe operating temperature point of equipments. There are many types of temperature sensor will used by this PID Controller but here only for experiment, discuss about RTD sensor and taken reading and testing based on that discuss some points.

For monitoring a reading here used Citec SCADA software v6.1 (licence) and create a project for taking a reading of output response for PV and %power of PID Controller. Below some snapshots are shown in different condition.

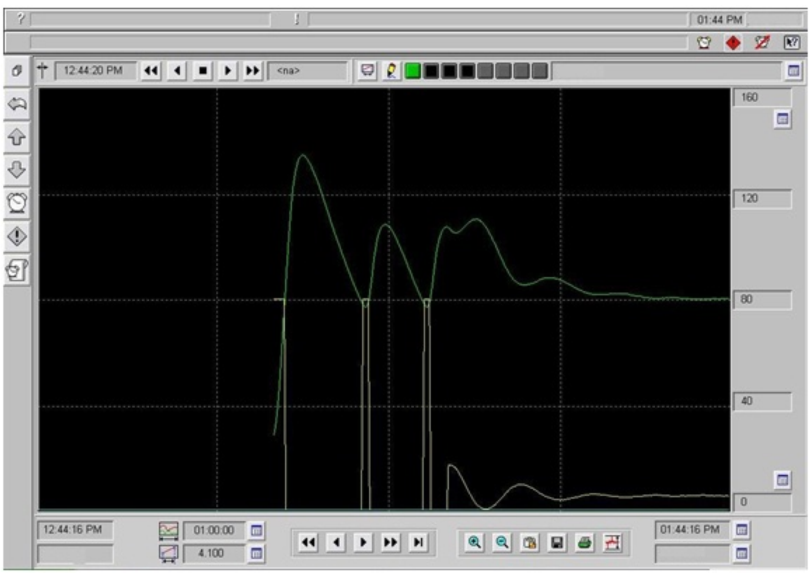


Figure 4.1: Auto tune Cycle with Modified Relay Feedback Method

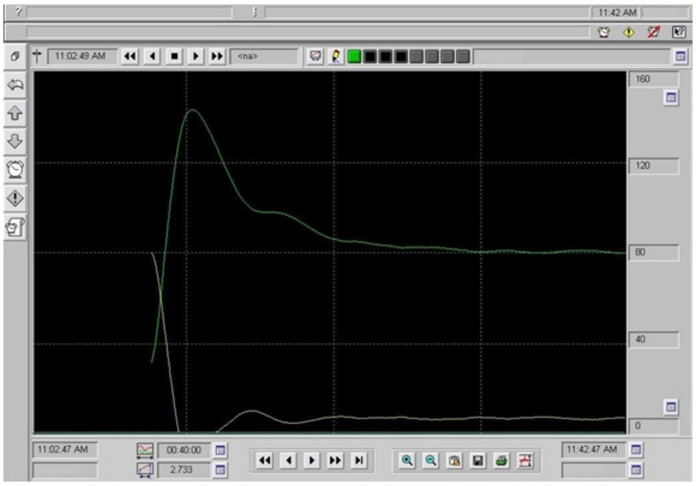


Figure 4.2: After Auto tuning PID Controller output response of PV (green) and %power (yellow) with a set point 80°C

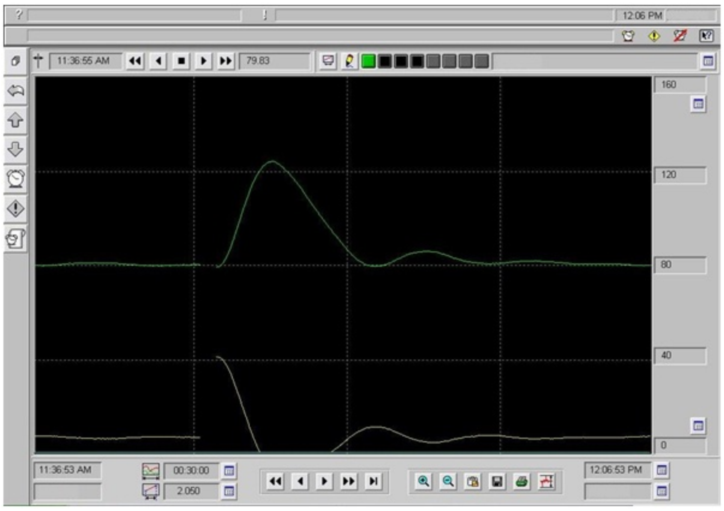


Figure 4.3: Output Response of PV(green) and %power(yellow) after a restart the PID Controller at set point 80°C

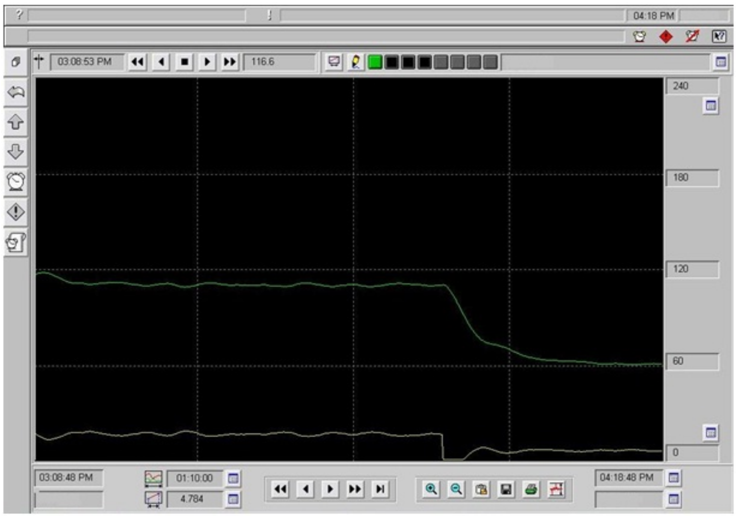


Figure 4.4: Output response of PV (green) and %power (yellow) for change the set point in runtime from 100°C to 60°C

4.2 Testing Process

There are several testing used in this project like Linearity test, Stability test, Isolation test, Ignition test, Temp-Co test, Vibration test, Radiation test etc. for checking a quality of product. Here some testing results are given below at some specification.

4.2.1 Stability test

Here take a stability test of PID Controller with RTD Input type and set point is around 80°C. Testing result table 4.1 is as shown below.

Table 4.1: Stability test readings for RTD I/P Type

Sr. No.	Time	Actual Display Reading (°C)	Expected Display Reading (°C)	% Error
1	12:45 pm	79.97	80.00	0.0003
2	01:00 pm	79.97	80.00	0.0003
3	01:15 pm	79.95	80.00	0.0005
4	01:30 pm	79.97	80.00	0.0003
5	01:45 pm	79.97	80.00	0.0003
6	02:00 pm	79.98	80.00	0.0002
7	02:15 pm	79.97	80.00	0.0003
8	02:30 pm	79.98	80.00	0.0002
9	02:45 pm	79.96	80.00	0.0004
10	03:00 pm	79.98	80.00	0.0002
11	03:15 pm	79.95	80.00	0.0005
12	03:30 pm	79.96	80.00	0.0004
13	03:45 pm	79.96	80.00	0.0004
14	04:00 pm	79.96	80.00	0.0004
15	04:15 pm	79.96	80.00	0.0004
16	04:30 pm	79.96	80.00	0.0004
17	04:45 pm	79.96	80.00	0.0004
18	05:00 pm	79.95	80.00	0.0005
19	05:15 pm	79.96	80.00	0.0004
20	05:30 pm	79.95	80.00	0.0005
21	05:45 pm	79.95	80.00	0.0005
22	06:00 pm	79.95	80.00	0.0005
23	06:15 pm	79.96	80.00	0.0004
24	06:30 pm	79.96	80.00	0.0004
25	06:45 pm	79.96	80.00	0.0004

4.2.2 Linearity test

For testing a output is linearly vary according to input with RTD input time for short and Long cable test reading are as below in table with full span range.

Table 4.2: Linearity test readings for RTD Short wire

Input RTD (Range -200 to 850 Deg. C)		
Exp. Disp. Rdg. In deg. C	**Actual UUT Disp. Rdg. In deg. C**	**% Error**
-200	open	open
-100	-100	0.00
-50	-50	0.00
0	0	0.00
50	50	0.00
100	100	0.00
150	150	0.00
200	200	0.00
300	300	0.00
400	400	0.00
500	500	0.00
600	600	0.00
800	800	0.00
850	open	Open
Above 850	open	Open
I/P Open	open	Open
Max % Error positive		**0.00**
Max % Error Negative		**0.00**

Table 4.3: Linearity test readings for Long Wire

Input RTD (Range -200 to 850 Deg. C) with Long Wire Test		
Exp. Disp. Rdg. In deg. C	Actual UUT Disp. Rdg. In deg. C	% Error
-200	open	open
-100	-100	0.00
-50	-49	-0.10
0	0	0.00
50	50	0.00
100	100	0.00
150	150	0.00
200	201	-0.10
300	300	0.00
400	400	0.00
500	500	0.00
600	600	0.00
800	799	0.10
850	849	Open
Above 850	open	Open
I/P Open	open	open
Max % Error positive		0.10
Max % Error Negative		-0.10

Result: Reading is within accuracy (0.25% of FS + 1 degree), hence it is acceptable

4.2.3 NMRR Test

Table 4.4: NMRR test readings for RTD I/P type

NMRR Test	I/P Type:-	RTD.1°C		Total	v / count	V
Range:-	-199.9	TO	850.0	1049.9	9.9E-05	0.130088
Resistance:-	18.53		390.38			
Voltage source:-	2500		2500			
mV:-	94.825	TO	1134.324	1.03950		
NMRR test taken at 800.0 °C						
			NMRR =	-20 log (v/V)		
Apply VRMS:-	0.046	V		62.3712213		dB

Result: # NMRR is >40dB in RTD PT 100. Hence it is acceptable

4.3 Discussion

The first stage of this project is to analyze all the PID Auto tune techniques to get the Temperature sensor characteristic using proto board and multi meter. After the project proceeds to the next stage.

The next stage is software development. Based on the analysis, the project programming can be designed. After designed the project programming, the programming was tested by using TASKING software. This method use d to test either the program is successfully built or not. If there are any errors in the programming, the ICC AVR V6.2 software will locate the errors, and then the error can be fixed either directly or by checking the problem using help library.

After successfully build and compile the project programming, the project proceeds to the next stage which the programming was tested on the micro controller ATMEGA 128. The hex file was created by ICC AVR and downloaded into the ATMEGA 128 using AVR Studio V4.0 software. The programmer used as interface between ATMEGA128 with the software AVR Studio.

The AMEGA128 that already downloaded with the hex file was tested using proto board and multi meter. The micro controller will interface with temperature sensor and other hardware to test either the project programming practically successful or not.

After the circuit on the proto board successfully process the data from the sensor circuit and can control the output, the circuits are ready to be soldered on the PCB board. After the circuit was soldered, once again all the circuit will be tested.

If the circuit has a problem, the circuit will be troubleshooting by using multi meter. The project is successfully done when all the sensor work properly based on characteristic.

For the result of tuning and experiment there was clearly that the PID controller was built successfully and fully functioning without a problem.

Based on the IEC regulation most of the electric equipments are required to operate in safe temperature range and the system was proved of being able to work in the IEC temperature regulation requirement.

Chapter 5

Conclusion and Future Scope

5.1 Conclusion

The objective of this project is to develop a system prototype to evaluate the performance of proposed PID Controller auto tune method with Low Rise time, small overshoot, fast settling time and low steady state error. This project has almost successfully accomplished the objective. The complete prototypes successfully control the temperature with good linearity and stability condition.

The project also successfully reached the second objective which to overcome linearity problem in the industry by design a new and low cost prototype using the proposed PID controller tuning method. It also can be seen as a small investment to protect expensive equipments that widely used in industry field.

This project is quite tough for me because most of the work done is new for me. Most of the time spent to read, understand and make a research about the different PID Controller Tuning Algorithms. Nearly one semester was taken to design the project structure and make an analysis about the required system. After understand the project system, the next adventure is to implement the hardware development and testing with Industrial Environment Condition.

5.2 Future Scope

Even though the PID Controller was successfully built, many improvements can be done to maximize the capability of the PID temperature controller.

The controller can be implemented with Fuzzy logic or Genetic Algorithm which is widely used now a day in industries for precious application and where, the rise time and pick over shoot is a major issue. PID Controller was seen as a method that can minimize the power consumption rather using typical on-off controller which used more power energy and can save the running cost of a industry.

References

[1] Astrom, K. J. & Hagglund, T., Revisiting the Ziegler-Nichols step response method or PID control. Journal of Process Control, Vol. 14, pp. 635-650, 2004

[2] J. G. Ziegler and N. B. Nichols, ”Optimum settings for automatic controllers,” Trans. ASME, vol. 64, pp. 759768, 1942.

[3] Majhi, S. and Atherton, D.P., A Novel Relay Feedback Identification Method. Proc. of 5th European Control Conference, Karlsruhe, Germany, 1999.

[4] C. C. Hang and Kok Kee Sin, ”On-Line Auto Tuning of PID Controllers Based on the Cross-Correlation Technique” IEEE Transactions on industrial electronics, Vol. 38.No.6, December 1991.

[5] Vance VanDoren, Relay method automates PID Loop tuning, control engineering, pp.52-54, September 2009

[6] Yu-chang cheng, cheng-ching Yu,” Relay feedback identification for actuator with hysteresis”, Industrial & enineering chemical research,vol.39,no.11,pp.4239-4249,2000.

[7] Majhi, S. and Atherton, D.P., An Auto tuning Method for PID Controllers. Proc. of Intl. Conf. on CCSP-2000, IISc. Bangalore, India, July 2000, 427-431.

[8] Mehta, U. and Majhi, S.,On-line Relay Test for Automatic Tuning of PI Controllers for Stable Processes, Transactions of the Institute of Measurement and Control, Vol. 34(7), pp. 903 - 913, 2012.

[9] YangQuan Chen, ChuanHua Hu and Kevin L. Moore, Center, ”Relay Feedback Tuning of Robust PID Controllers with Iso-Damping Property” Proceedings of the 4th IEEE Conference on Decision and Control,USA,2003

[10] Huang C.C., A.P. Loh and V.U. Vasnani, Relay feedback auto-tuning of cascade controllers, IEEE Trans. Control System Tech., vol.2, pp.42-45, 1994

[11] Theory of Modbus Protocol: http://en.wikipedia.org/wiki/Modbus

[12] Ang, K.H. and Chong, G.C.Y. and Li, Y., PID control system analysis, design, and technology. IEEE Transactions on Control Systems Technology 13(4):pp. 559-576, 2005

[13] Sung, S.W., J.H. Park and I. Lee, Modified relay feedback method, Ind. Eng. Chem. Res., 34, 4133-4135, 1995.

[14] Majhi, S. and Atherton, D.P. : Relay Feedback Identification of Nonminimum Phase Systems. Proc. of 2nd Intl. Conf. on Control of Industrial Processes (ICCIP), NewCastleupon Tyne, UK, 245-248.

[15] Astrom, K., T. Hagglund, .PID Controllers; Theory, Design and Tuning., Instrument Society of America, Research Triangle Park, 1995.

[16] David I. Wilson "Relay-based PID Tuning", Department of Electrical & Electronic Engineering, Feb/March, 2005.

[17] Yu, C. C. Autotuning of PID Controllers: Relay Feedback Approach; Springer-Verlag, London, 1999.

[18] Jhon A. Shaw, "The PID Control Algorithm", process control solution, 2nd edition, December 2003

[19] Yu, cheng-ching, Auto tuning of PID Controller-a relay feedback approach, 2nd edition, springer,2006

List of Publications

[1] Ankur Patel, Vijay Savani, ”Performance Analysis of PID Controller and Its significance for Closed Loop System” Published in International Journal of Engineering Research & Technology (IJERT) March Vol.3 Issue-3, ISSN: 2278-0181, pp.1843-1847,2014