High Availability in Analog PID controllers

Abhijeet Malik Ninad Pradhan Rohit Pradhan
Rahul Saxena

Conducted for

Larsen & Toubro Ltd., Mumbai.

University of Mumbai

Fr. Conceicao Rodrigues College of Engineering,

Bandra, Mumbai.

2002–2003

CERTIFICATE

This is to Certify that the report titled

High Availability in Analog PID Controllers

submitted by

Abhijeet Malik Ninad Pradhan Rohit Pradhan
Rahul Saxena

Embodies the bonafide work done for

Larsen & Toubro Ltd., Mumbai.

This is to certify that Abhijeet Malik, Ninad Pradhan, Rohit Pradhan, Rahul Saxena have successfully completed the requirement of the TERMWORK OF PROJECT II as prescribed by the University of Mumbai.

Date	Signature of the guide
Official Seal	Name, Designation
H.O.D	Principal
External Examiner	Internal Examiner

Acknowledgements

It gives us great pleasure to submit our project report on ’High availability in analog PID controllers‘, for the completion of the final year course in Electronics engineering. This project was carried out for the Electronics Department, Heavy Engineering Division, Larsen and Toubro, Powai. Working under the guidance of Mr. M.G Ramakumar has been a steady journey up the steep learning curve which awaits final year students on their first project experience. We would like to express our appreciation for him for having taken time from his schedule to accommodate the numerous discussions we had along the way. He has been patient with our mistakes and tried to maximise the result given our limitations. We would also like to thank our Principal, Dr. R Sesha Iyer and Head of Department, Prof. B.R Prabhu for having provided us with the facilities necessary for the successful completion of this project. We are grateful to Mr. D.V Bhoir, our internal guide, for providing us with theoretical and practical insights into analog design and controllers. He also referenced to us the appropriate material when required. A word of thanks is also due to Mr. Dilip for his technical help at many junctures.

Contents

Acknowledgements **ii**

Abstract **viii**

1 Introduction **1**

1.1 A Typical Application 1

1.2 An Overview 3

2 Current Methods of Implementation **7**

2.1 Microcontroller Based Method 7

2.2 Analog Method 10

2.3 A Combination Of Methods 10

3 The PID Card **12**

3.1 Control Algorithms 13

3.1.1 On–off 13

3.1.2 P or proportional 13

3.1.3 PI or proportional–integral 14

3.1.4 PD or proportional–derivative 15

3.1.5 PID or proportional–integral–derivative 16

3.2 Circuit Of The Cards 18

3.2.1 Choosing The Right OP–AMP 19

4 High Level Control 23

4.1 A Feedforward Controller 23

4.2 A Cascaded Controller 25

5 Driver Motor and Transducer 29

5.1 Motor 29

5.2 Driver 31

5.3 Transducer 32

5.4 Why Use Non–Standard Components 33

6 Monitoring Card 34

6.1 Setpoint Of The Primary Loop 35

6.2 Feedback From Transducer 37

6.3 System Shutdown 38

6.4 Setpoint Of The Secondary Loop 38

7 DC Motor Modelling And Tuning 40

7.1 Parameters Of DC Motor 41

7.2 Measurement Of DC Motor Parameters 42

7.3 Limitations Of Parameter Measurement For Non Standard Motor 44

7.4 Tuning Of The Controller 45

8 A User Interface 47

8.1 Keypad Matrix 50

8.1.1 Interfacing the matrix with the 89C51 52

8.2 LCD Display 53

8.3 The DAC0808 DAC 54
8.4 The ICL8038 Waveform Generation IC 55
8.5 Software 57

9 Further Developments 60
9.1 A Scalable Model For Greater Redundancy 60
9.2 A Bus–Based System 61
9.3 Provisions For A Standard Driver 62
9.4 A User Interface With Greater Functionality 62

10 Conclusion 64

A User Interface Program 67

B Data Sheets 81

List of Figures

1.1 Block Representation of The Proposed System 6

3.1 The Basic Feedback Loop 12
3.2 P Controller Response 14
3.3 PI Controller Response 15
3.4 PD Controller Response 16
3.5 PID Controller Response 17
3.6 The Inner PID circuit 20
3.7 The Outer PID circuit 21

4.1 A FeedForward Controller 24
4.2 A Cascaded Controller 25

5.1 DC Motor Voltage Speed Characteristics 30

6.1 Window Detector for Monitoring Circuit 36
6.2 The Basic Monitoring Circuit 36

7.1 Eqivalents Circuit of a DC motor 41

8.1 The UI Layout 48
8.2 The Keypad Layout 51
8.3 Sample LCD Display 52

8.4 Wave generation using 8038 56
8.5 Wave generation using 8038 56

10.1 A Flowchart representation of the System 66

Abstract

The primary aim of our system is to control the position of a DC motor using analog circuit components. The motor is driven by a high current op-amp driver. The second goal is to make a redundant system which will have a higher probability of remaining in operation for a sustained period of time. Redundancy is achieved by using two PID controller cards with the active card being continuously monitored by analog circuits. Appropriate action is taken when an error is detected. A user interface card provides a human serviceable diagnostic tool.

Chapter 1 gives the system block diagram and explains its potential application.

Chapter 2 covers the two currently available methods for implementing a control system.

Chapters 3, 5, 6 and 8 give details of the cards and components which the system is composed of; namely the PID cards, driver motor and transducer, monitoring card and user interface card respectively.

Chapter 4 discusses the advantages of high level control, and compares and contrasts the feedforward and cascade schemes of this type of control.

Chapter 7 talks about tuning of cards, and modelling of a process which is

the first step for standard tuning methods. The process in question is a bidirectional DC motor.

Chapter 9 suggests further improvements in our implementation.

Appendix A is a listing of the microcontroller code which defines the functioning of the user interface.

Appendix B is a collection of data sheets of the components we used in our circuit designs.

Chapter 1

Introduction

1.1 A Typical Application

The project by itself provides for three distinct features:

1. control of angular position
2. redundancy in design
3. diagnostic capability

These features are best utilised by an application for position control which is located in an area with low accessibility and which requires consistent and uninterrupted service from the system. Such systems would be found in rudder flight surface control systems.

- To control the direction of the vessel, the rudder moves in either a clockwise or anticlockwise sense. The angle of the rudder with respect to the body of the vessel determines the direction of the fluid flow around the body, and ultimately the direction of the boat or ship. This calls for the use of a system to control angular position, the first feature of our project.

- Given its position of importance in the movement of the boat, the position control of the rudder has to be failsafe. This is where the concept of multiple actuators connected in parallel comes into use. If one of these actuators or the control system driving it fails, another one working with the same input conditions and providing the same output can take its place without a loss in time or causing a system stoppage. This calls for the use of redundancy in design, the second feature of our project.

- A system malfunction is a dangerous condition for the rudder. If for any reason such a malfunction takes place, and switching to a redundant controller does not help solve the problem, the operator should be informed about this functioning error. Our system does this with the use of simple LEDs, but if the location is remote, error indication can be carried out by a communication device. Once a problem is detected in one or more cards, the vessel can be navigated while the faulty card is still being replaced. The faulty card can be tested with a dummy motor and transducer using the user interface. This allows us to test for basic functionality before putting it in command of ship steering. An application for a critical area needs a test facility before putting it to use. This calls for diagnostic capability, the third feature of our system.

Hence our application is well suited to the purpose of rudder position control. Our section on further developments highlights the fact that for more specific applications, the system can be standardized and components used can have more utility in real life applications than the motor and potentiometer pair we have used as a prototype.

1.2 An Overview

Our project is a motor position control system. The position of the motor is represented electrically as a voltage feedback from the potentiometer coupled with its shaft. Now every voltage value returned by the potentiometer during the rotation of the motor shaft is uniquely associated with the angular position of the shaft in degrees.

We make use of the above property to provide the voltage equivalent of a desired angle value, as the set point to our control system. The manner in which a set point is reached by the system is decided by the controller algorithm to be followed. We have used a PID algorithm in the controller. The PID controller provides a voltage to the driver of the bidirectional motor, and the driver is a high current output circuit which provides adequate current to the motor for its rotation. The separation between the desired position and the current position determines the speed at which the motor rotates during a specific instant of its motion. Since the PID system takes care of these variables, that are the speed of motor and its ultimate position, we get an automatic control system for motor position control. The utility of our design is in applications which need a system to be running without interruptions for long periods of time. Also, in case of failure of the process, it is of vital importance that the failure is indicated and the input to the driver is cut off, causing it to halt. A simple position control system is not equipped to deal with a possibility of card failure.

The solution to the above problem is twofold:

- Introduction of redundancy in the control cards.
- Monitoring of different points in the circuit to check for improper functioning.

Redundancy in design is when two or more cards in parallel receive the same input and feedback and, when working correctly, give the same output to the next stage. In a one card system, failure of the card would cause a shutdown, which is highly undesirable in a critical application. Let us say we have a case where two such cards are connected in parallel. Now, failure of one card would simply initiate a card switching operation, where the second card now becomes responsible for passing its output to the next stage, which in our case happens to be the motor driver.

Redundancy clearly helps us buy time by keeping the system running despite a card failure. The faulty card can now be replaced. But it is still not clear how a card failure is detected. For this purpose, we have a third card called a monitoring card which tells us whether the card supplying an input voltage to the driver is in good health. It monitors a few parameters of the card currently in control and, if it finds some error in its functioning, it switches cards so that the other card now gets control.

The system may be used as one of the final stages for a bigger position control system, which may be governed by a microprocessor. It may also be used all by itself, or in a stand alone configuration. In any case, we may need to know how the cards are performing before putting them directly in use in an application.

Hence, we have a fifth card called the user interface card which is essentially a diagnostic feature of our system. Through the interface, we can give a set point in terms of the angular position. It converts this value to corresponding voltage and gives that to the set point of the PID cards. If standard waveforms are essential for testing, then a waveform generator is provided for sine, square and triangular wave outputs. Our project is a generalised implementation of the position control system with card redundancy and diagnostic features. This gives flexibility in using the concept for more specific tasks. More features can be added according to the needs of the application it is intended for. The user interface can have a more intricate diagnostic toolkit, the driver circuit can be replaced by a standard driver card, a motor of higher rating may be selected, among other possible additions. Our goal was to come up with a framework from which a system can be built which confirms to more specifications.

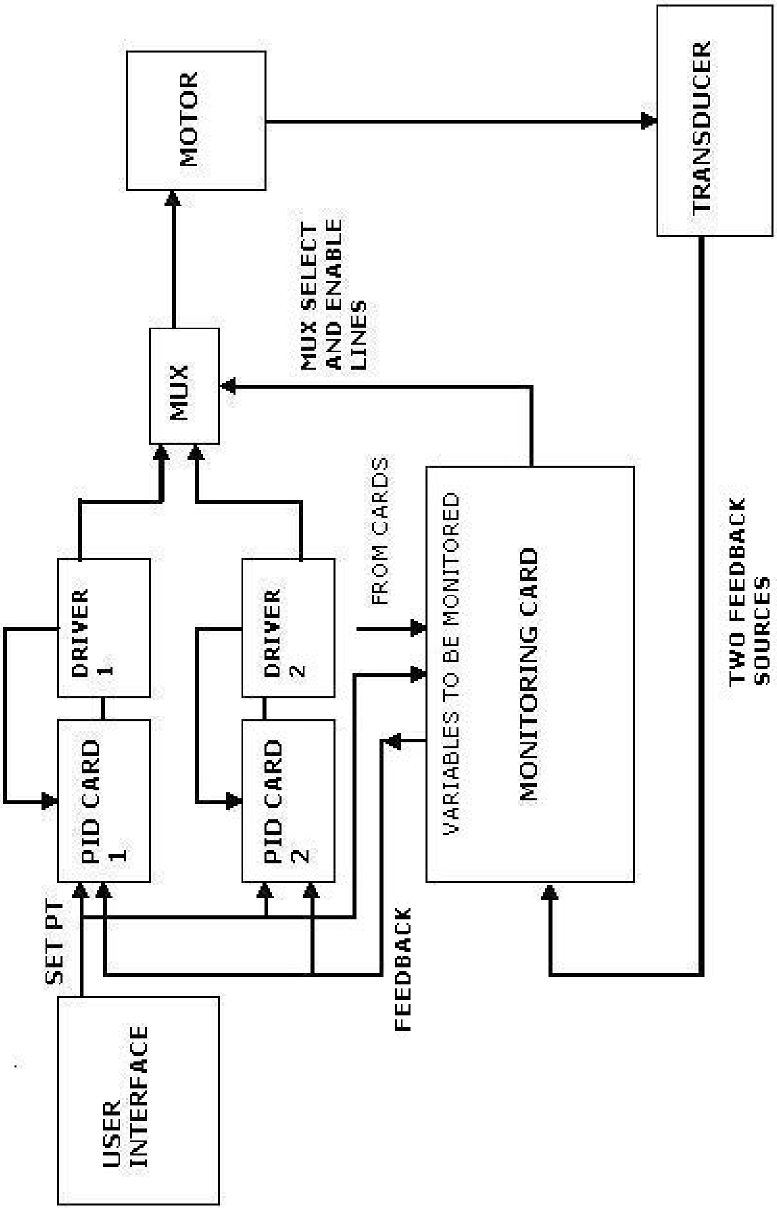


Figure 1.1: Block Representation of The Proposed System

Chapter 2

Current Methods of Implementation

The control system algorithm can easily be represented by a mathematical model; hence the system on the whole gives a well defined response depending on the values of the variables at a given point of time. Therefore a computer is often used to implement a PID controller. The hardware requirements for a microcontroller versus that of an analog system are obviously different, but the considerations which led us to choose an analog implementation were the result of other major differences between the analog and digital methods of PID implementation.

2.1 Microcontroller Based Method

One of the main advantages of using microcontrollers in control system application is the flexibility by the controller. In the microcontroller, the whole controller is emulated by means of software routines. This gives us the ability to handle multiple feedback variables. The interaction between all these separate loops can be accounted for by using software. Microcontrollers also benefit from the ability to store variable data and hence log various parameters of the system. Ancillary advantages such as susceptibility to noise, linearisation and error correcting in data also result from such an implementation.

Another feature, one which is inherent of any finite state machine, is the ability to iterate. This is beneficial when tuning of the controller is required. However the advantage of such a feature is in question considering the amount of effort required in developing routines, for such a processes. The first disadvantage stems from the fact that the computer is a serial device in terms of the manner in which it carries out its operations. It only performs one operation at a time. The data read by the computer has strictly to be in a digital form. This necessitates the use of an analog to digital converter prior to the input stage to the computer. ADCs have the inherent limitation of speed, and are known to be slow devices.

Now the digital system has to carry out the following operations in a simple single loop PID controller which doesn't even consider the monitoring aspect of the system:

- Accept analog setpoint
- Digitize and send to computer
- Accept analog feedback
- Digitize and send to computer
- Carry out P I and D calculations sequentially
- Add the output to get PID output digitized value

- Convert to analog and send to system

When we impose additional conditions which require cascaded control and system monitoring to be incorporated into the digital system, it requires a more complex flowchart. The first penalty that a digital system bears is that of time.

The data which is sent to the computer for analysis has a bit resolution fixed by the ADC that is working on the data at the previous stage. Quantization errors are imminent in such a scheme. Reduction in quantization error will result from using more levels to represent the data digitally, but that will mean a greater number of bits per sample, and more time spent in the conversion and possibly more hardware required. The second penalty that the digital system bears is that of resolution.

The nature of our system is such that the highest priority has to be given to its performance not being hampered by the malfunction of the PID card. This was the reason for the concept of redundancy being used in realizing the circuit. The failure of one card simply means that the other card/s take over control of operation. The faulty card can be replaced by the operator while the other card is still providing uninterrupted service. The replacement can be done while the system on the whole is still online, and time is saved. It is quite cost effective to replicate an analog card and replace it when it stops working. To replace the digital card in a similar redundant digital system comes at a cost. The digital circuits needs ADC, DAC, buffers and the microcontroller. Hence a modular replacement of the PID card will bear a bigger financial burden on the user. The third penalty that the digital system bears is that of cost of replacement.

2.2 Analog Method

The emphasis under analog systems is on electronic techniques, using op amps as active elements. The main reason for this is their widespread use and proven reliability, in terms of robustness and linearity, as compared to pneumatic systems. Also implementation of electronics techniques allows for easier integration with existing electronics systems.

For analog data, op amp implementations are ideal because they don't require any scaling or quantisation of data. Hence the response time is only limited by the time lag caused by the internal propagation delay of the components in the signal path. This delay is of the order of nanoseconds.

The second advantage is the cost of implementation of the analog control system; it can almost entirely be constructed using resistors, capacitors and op amps. These components are widely available and have proven reliability. The cost of construction is also low. This allows us to implement redundancy in the control system.

2.3 A Combination Of Methods

The analog system is not without its drawbacks; it is less flexible to change in parameters and every change in design related to the controller algorithm requires a sweeping replacement of the analog circuitry. Tuning demands a careful study of the system response, unlike the digital method where an iterative tuning procedure takes care of much of the manual efforts at repeated tuning, if that is necessary. Simulation of the system becomes easy and a dedicated user interface may not be the norm for a reliable system. The front end is much more user friendly and the operator need not have deep insights into control systems to keep a tab on its quality of output.

For a major system such as vehicle steering, a fast, reliable and robust method as provided by the analog method is desirable, coupled with the advantages of the digital system. These systems use both of the methods to get the best of both. The front end or the ultimate controller is a microcontroller, while all the controllers in cascade to it at different stages are analog.

We have used only analog components for the actual control circuit since it was sufficient for effective functioning of the motor position control system.

Chapter 3

The PID Card

The PID card, being responsible for controlling action, is at the heart of our system. The basic feedback control loop is shown below:

A control system regulates a process. The regulation of this process is achieved by the measurement of its current value, which, on comparison with the desired value, tells the controller how much energy to give to the process. The desired value, or set point, is where we want the output to reach at the end of control action.

The manner in which a controller reacts to a difference in the set point and feedback is called the controller algorithm. There are various criteria

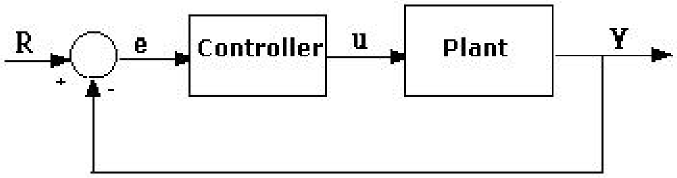


Figure 3.1: The Basic Feedback Loop

which provide references from which we can narrow down on the suitable control algorithm. The two most important to our implementation were the overshoot and the steady state error.

3.1 Control Algorithms

Control algorithms available to us were on–off, P, PD, PI and PID. These are explained in the following section.

3.1.1 On–off

This is the most basic method of achieving control. The valve is turned on when the output is below the dead–band, which is a range of acceptable output values, and it is turned off when the output is above it. On–off control suffers from the shortcoming that the controller output is always hunting for the set point, that is, it oscillates about it. This oscillation or ringing is an undesirable characteristic, and the on–off controller is not suited to our application.

$$u = \begin{cases} u_{max} & \text{if } e > 0 \\ u_{min} & \text{if } e < 0 \end{cases}$$

3.1.2 P or proportional

The ‘proportional’ term in the nomenclature tells us that the output of controller is made proportional to some value. This value is the error.

$$u(t) = K_p\, e(t) + u_b$$

This indicates that, depending on the value of K_p, more the error, higher is the output. Proportional band is defined as the input range required for the

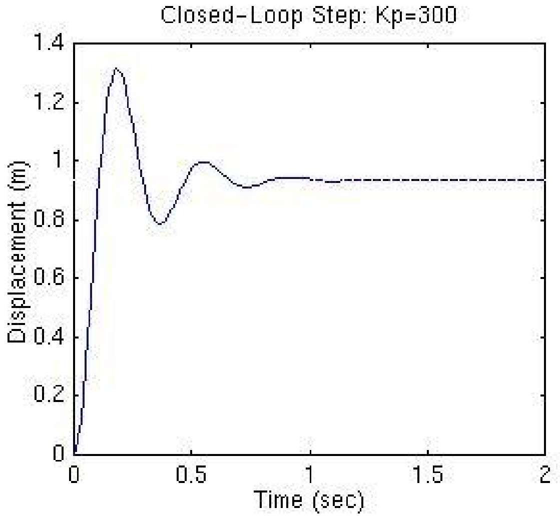


Figure 3.2: P Controller Response

output of the controller to saturate. A smaller proportional band indicates that the gain of the controller is higher, and it takes less error to get maximum output. This increases speed of response but makes the control loop more vulnerable to disturbances, and hence unstable. Another disadvantage of the P controller is that it has an inherent offset which implies that steady state error is not the minimum possible value in this algorithm.

3.1.3 PI or proportional–integral

This is an improvement over the previous P mode. The improvement is to introduce a correction to the output which will keep adding or subtracting a small amount to the output until the motor reaches the setpoint, at which point no further changes are made. We need to define how often such corrections are to be made. The answer to the 'how often' question is called the reset time, or K_i. It supplies energy to process depending on the accumulated

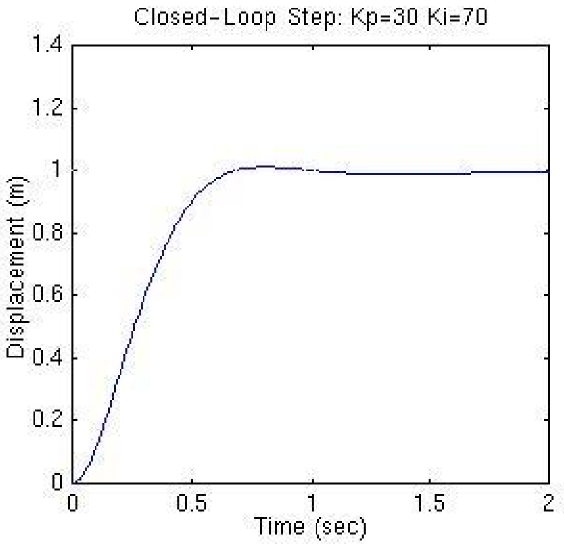


Figure 3.3: PI Controller Response

error over the reset time period.

This method implies the use of an integrator.

$$u(t) = K_p \left(e(t) \; + \; \frac{1}{T_i} \int_0^t e(\tau) \, \mathrm{d}\tau \right)$$

Integral action keeps happening as long as there is an error present in process, and hence it eliminates the offset inevitable in a P only controller. Integral tends to introduce oscillations in the output, and a careful balance of the P and I gains is required for satisfactory system performance.

3.1.4 PD or proportional–derivative

Just as the integral term responds to an accumulation of error over time, the derivative responds to rapid changes in error with respect to time.

$$u(t) = K_p \left(e(t) \; + \; T_d \frac{\mathrm{d}e(t)}{\mathrm{d}t} \right)$$

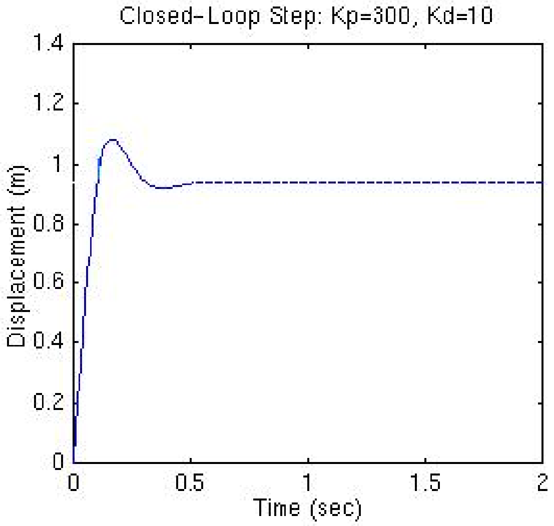


Figure 3.4: PD Controller Response

Since it is dependent on changes in error rather than the error itself, the derivative action in rather ineffective for a steady state error. It is highly effective in responding to sudden changes in error and helps stabilise the closed loop response. Its disadvantage is that it causes the control system to become more sensitive to noise. It reduces overshoot in the system.

3.1.5 PID or proportional–integral–derivative

The PID system clearly is an integration of the three control algorithms listed above. It offers the combined advantages of the P, PI and PD methods and eliminated their individual shortcomings.

$$u(t) = K_p \left(e(t) \; + \; \frac{1}{T_i} \int_0^t e(\tau)\,\mathrm{d}\tau \; + \; T_d \frac{\mathrm{d}e(t)}{\mathrm{d}t} \right)$$

A well tuned PID controller works to keep overshoot down, or in fact completely avoid it, and keeps the steady state error down, to zero in an

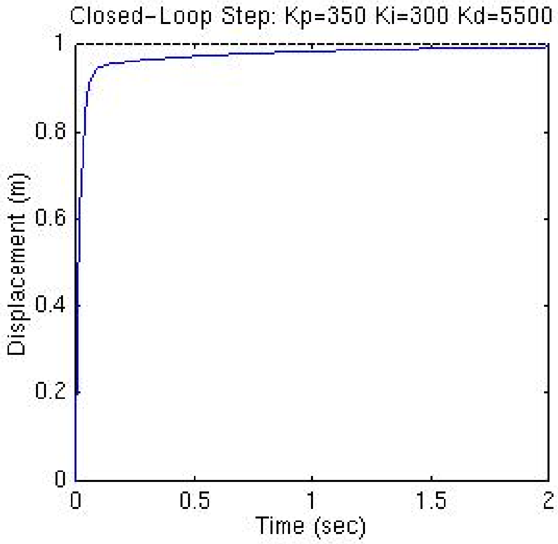


Figure 3.5: PID Controller Response

ideal case. Why are these two parameters important for selection of a suitable control mode for our application?

Overshoot is a measure of how much excess current is provided to the motor as the controller output rises initially. The steady state error should not be more than the minimum requirement to keep the motor moving. The objective, hence, is to keep both the steady state error and overshoot to an absolute minimum. A PID system satisfies this requirement.

In any case, all the three amplifiers are provided for. In the case that certain motors do not require all of the facilities offered by a PID, one or both of the two amplifiers, I or D, may be taken out as desired. The block diagram of the PID control card is given. We notice that there are two PID loops cascaded on the same card. There are certain advantages in the use of this cascaded control scheme which are enumerated in the next section.

3.2 Circuit Of The Cards

The op–amp level circuit of the PID cards is almost a direct realisation of the PID equation. According to the equation, the P I and D gains operate on the error. Thus we require a difference amplifier of unity gain to provide us with the error which serves as input to the next circuit. The next stage is an arrangement of the gain amplifier, integrator and differentiator in parallel.

Though we have separate terms for P I and D in the equation, the proportional gain is a common factor in all the terms. This implies that the integrator and differentiator each have a resistor tuned to the proportional gain of the system. This means that there are two gains to be considered when fixing values for the I and D resistor components: the proportional gain and the integral or derivative time gain. For this reason we have a potentiometer instead of a fixed value resistor in the suitable place for the I and D amplifiers.

Once the error goes to the PID configuration, it is processed as per the parameters in the mathematical form of the controller. The output from each of these amplifiers now arrives at a single point where it is summed. Now the output of the first PID loop provides a set point to the second or the inner PID loop. The output of the inner PID loop passes on to the driver circuit after it has been suitably adjusted for the voltage range of the driver.

The two PID loops have the following set point–feedback–output combinations:

Outer loop: system set point–potentiometric feedback–output to inner PID.

Inner loop: outer PID output as set point–motor current converted to voltage as feedback–output to driver.

The output of inner PID going to driver tells us intuitively that the driver needs to be known before an input is given to it directly. This is because the input range acceptable to the driver and the range the inner PID is capable of delivering may be different. This is considered at a later stage.

3.2.1 Choosing The Right OP–AMP

The PID circuit we have designed is an analog circuit which exclusively uses op–amps for the control action. It is binding that this component is chosen with care so that the circuit functioning is adequate. Considerations while choosing an op-amp IC were as follows:

The PID circuit is potentially of use in cases where fast changes in the desired set point may be taking place. For example, in case of a standard wave like the sinusoid being used to test the controller. The op–amp needs to be able to handle fast changes; in effect it needs to have a good slew rate. Offset values need to be low.

The error generated by the error amplifier may be either positive of negative. Hence the op–amp ICs need to be of dual supply type. Also the error magnitude may be as high as around 7–8 volts and the output range of the op–amp IC needs to be sufficiently high to deal with such a large voltage. The number of op–amps in one IC should be more so that the chip count of the system can be reduced.

Keeping these factors in mind, we found the op–amp IC LF347N suited to the application. It is a JFET input quad op–amp IC with a high slew rate of 13 V/s. It can operate at differential voltages as high as ±30V. The connection of the IC and interfacing it with the rest of the components in the final circuit layout of the PID cards is given below.

It has to be kept in mind that two such cards are being employed in our system. Also the values of the components used at corresponding locations in

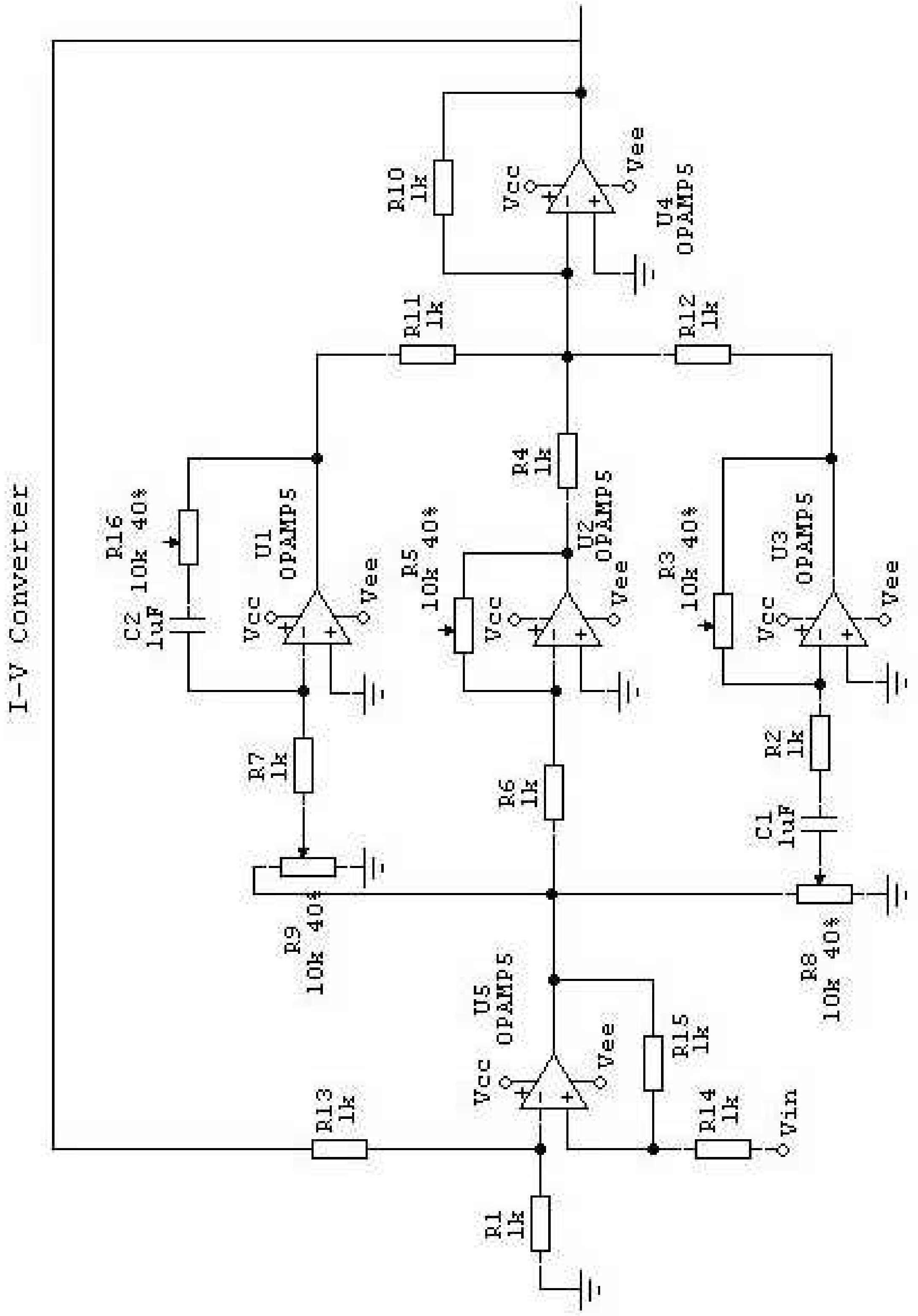


Figure 3.6: The Inner PID circuit

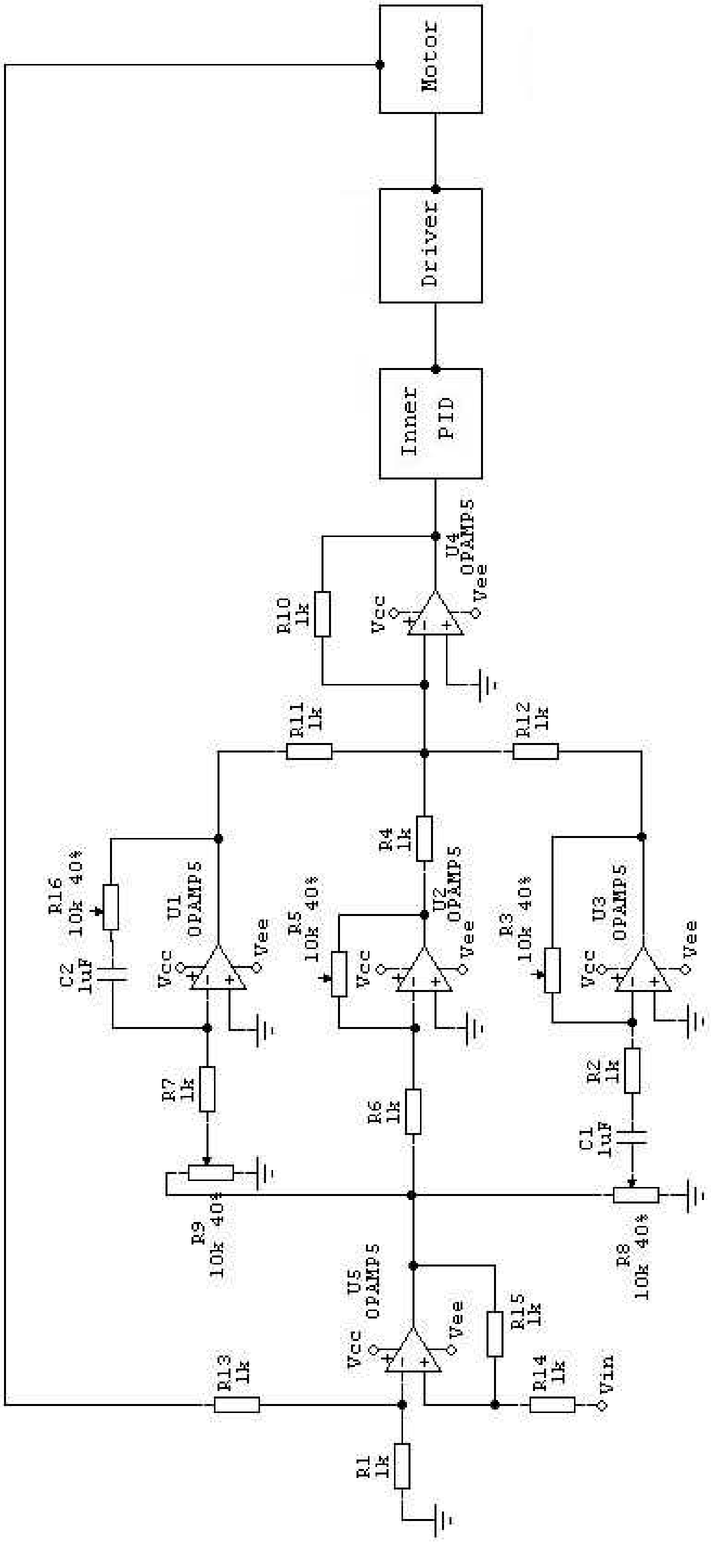


Figure 3.7: The Outer PID circuit

the implementation have to be exactly the same for both cards. This is what is meant by redundancy. How this redundant system is used fully through continuous monitoring of variables is a function of the monitoring card.

Chapter 4

High Level Control

Any reliable control system cannot operate on a simple logical relationship of the input, measurement and output. For an effective implementation, we need to consider the disturbances which act on the system and try and eliminate them. There are two models for this purpose which we have considered: the cascaded and the feedforward model. The difference between these two lies in when the disturbance is actually corrected. The cascaded system acts on the disturbance after the has occurred by measuring it and taking corrective action, whereas the feedforward system uses a highly precise model of the system to correct a disturbance before it shows up in the system working itself. The requirement of a model and the sensitivity of the feedforward system to such a model made us choose the cascaded implementation.

4.1 A Feedforward Controller

Before deciding on the use of a cascaded controller, we considered using a feed–forward type of controller. Feedforward control is a strategy used to compensate for disturbances in a system before they affect the controlled

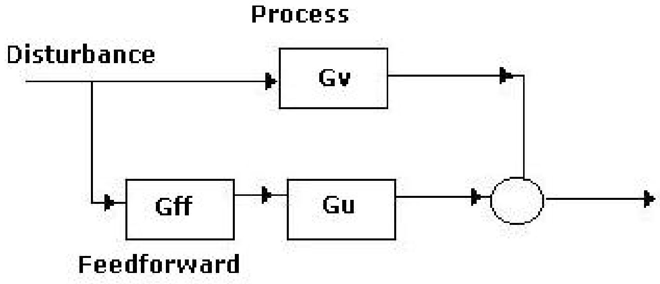


Figure 4.1: A FeedForward Controller

variable.

A feedforward control system measures a disturbance variable, predicts its effect on the process, and applies corrective action. Given an exact model of the process, the feedforward controller will adjust the manipulated variable so that the controlled variable is unaffected by the disturbance. In fact, the controlled variable has no influence over the control; corrective action is totally in response to the disturbance. It would be unusual to find that the input and output relationship for feedforward controller had to be one to one and linear to compensate for the disturbance. A gain and bias adjustment is always required to match the manipulated variable to the uncontrolled and disturbance variables. Additionally, it may require lead–lag elements, linearisers, non–linearisers, and a summer. Thus, feedforward by itself is an insufficient control. However, combined with conventional feedback, it can be a powerful control tool.

Along with these, the feedforward controller has a couple more drawbacks: All instruments in the loop had to be perfectly calibrated Disturbances other than feedforward variable could not be controlled Thus, a cascaded system controls both measured variables, with the master determining the set–point

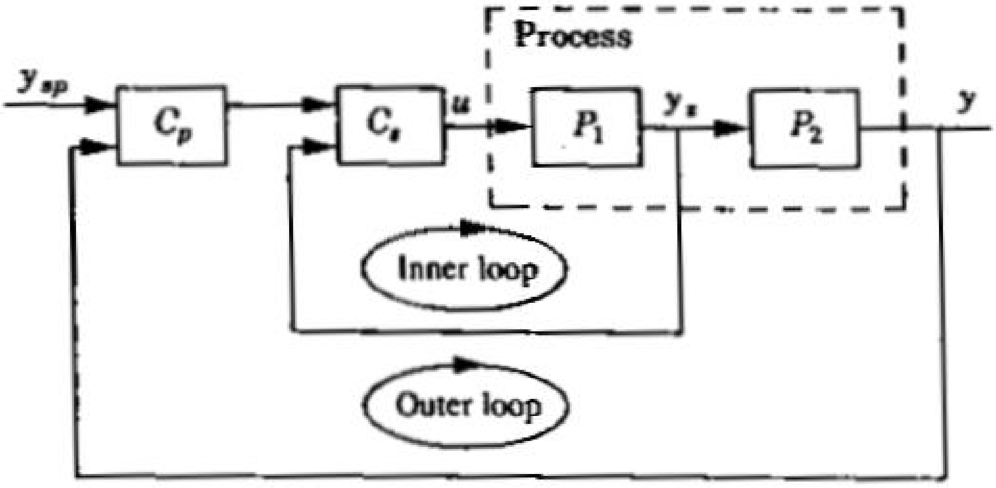


Figure 4.2: A Cascaded Controller

of the slave. In contrast, feedforward and feedback corrections independently adjust the control valve, and there is no control applied to the feedforward variable.

4.2 A Cascaded Controller

A cascaded control system is used in situations where there are many variables to be monitored and one control variable. It is a multiple–loop system where the primary variable is controlled by adjusting the setpoint of a related secondary variable controller. The secondary variable then affects the primary variable through the process.

The primary objective in cascade control is to divide an otherwise difficult to control process into two portions, whereby a secondary control loop is formed around a major disturbances thus leaving only minor disturbances to be controlled by the primary controller. It is used when there are significant dynamics between the primary and secondary loop. An intermediate measured variable measures the dynamics and takes appropriate action on the signal. It thus responds faster to the control system and achieves tighter control.

More than one loop is used in a cascade connection to completely account for the multiple variables. The set of all measured variable values which completely define the system at any given point of time is called the state of the system. Sometimes cascade loops use state feedback.

Our system in particular uses two feedback variables:

- Motor position feedback as provided by the potentiometer.
- Motor current feedback

The motor current feedback is of particular interest since it explains the necessity of a cascaded loop in our system. The motor can be expressed as a process when we mathematically model the motor as will be seen in the section on the driver, motor and transducer. Fluctuations in these parameters which define a DC motor can change the current flowing to the motor. The motor, as we know, is a current controlled device. These stray fluctuations in current to the motor can change the motor speed as it approaches the setpoint, changing the feedback to the primary error amplifier and hence there are fluctuations in the error signal. Hence a disturbance in one part of the motor is allowed to propagate through the entire system.

The second argument in favour of a secondary loop is the load changes which the motor shaft experiences. The final output of our system is mechanical, involving a pulley and belt drive attached to the motor, in addition to the gear system which is integrated with the motor itself. A motor requires more energy to move a greater load, or more current is required from the driver to make it work with the same speed as before a load increase. A momentary load change may be effected by minor mechanical shortcomings of the gear system, for example. This should not result in unusually high currents being drawn from the driver.

These disturbances are ideally controlled by a local loop which senses the actuator output, which is the current flowing to the motor from the driver. Corrective action is taken at this stage itself.

Some basic guidelines are followed for cascaded algorithm:

1. There should be a well defined relationship between the primary and secondary loops.
2. Essential disturbances should act in the inner loop.
3. Inner loop should have smaller average residence time than the outer loop.

In summary, the advantages of cascade control are all somewhat interrelated. They include:

- Better control of the primary variable
- Primary variable less affected by disturbances
- Faster recovery from disturbances
- Increase the natural frequency of the system
- Reduce the effective magnitude of a time–lag
- Improve dynamic performance
- Provide limits on the secondary variable

Cascade control is most advantageous on applications where the secondary closed loop can include the major disturbance and second order lag and the major lag is included in only the primary loop. The secondary loop should be established in an area where the major disturbance occurs. It is also important that the secondary variable respond to the disturbance. If the slave loop is controlling flow and the disturbance is in the heat content of the fluid, obviously the flow controller will not correct for this disturbance.

In our PID circuit we wanted to take into account the redundancy factor. Also we had been asked to design a circuit for a very high degree of accuracy, like a radar position control system or a precision motor control system.

All these factors helped us decide on the use of cascaded PID controllers for our project.

Chapter 5

Driver Motor and Transducer

The next stage in the system is the driver–motor–transducer combination which ultimately is the test for proper component selection and performance of the controller. This is because the motor position is the process which is to be regulated.

5.1 Motor

The motor we have used is for the purpose of demonstration. Hence the only criteria for selection of motor were that:

- It should be bidirectional to respond to the driver command, which is a reflection of the PID controller output.
- It should be operational within the voltage range at which the system is expected to work.
- It should be easily replaceable in case of motor failure.

For these reasons, we have used a Honeywell make bidirectional geared DC motor. This motor operates within a 0.5–12 volt range in either direction of

voltage (V)	CW		CCW	
	current (mA)	speed (rpm)	current (mA)	speed (rpm)
1	14.5	18	15.5	30
2	15.3	60	16.6	66
3	15.8	90	17.5	90
4	16.9	108	18.4	126
5	18.0	150	19.3	150
6	19.5	168	20.7	180
7	21.3	198	21.8	210
8	23.3	216	23.7	228
9	26.4	240	26.5	252
10	30.2	260	30.0	270
11	35.4	288	34.2	300
12	40.8	312	39.5	324

Figure 5.1: A sample Speed Voltage characteristic of the DC Motor which was used

rotation. An advantage of this motor is that in either direction of rotation, its current consumption remains practically the same for the same applied voltage to its terminals. This means that the response of the motor to an error is independent of the direction of its rotation. Also, the speed of the motor is variable and monotonic within its operational range. Hence, we have a system where, for the maximum error voltage, the speed of the motor is also maximum. A variable speed property with respect to voltage is essential for the PID algorithm in motor position control. The voltage speed characteristic is tabulated below. This was carried out at no load condition.

5.2 Driver

For the motor in question, we have to find an appropriate driver. What is a driver? It is a circuit or a system which operates the motor. The motor is a device which requires current to energise the stator and rotor poles and cause rotation of its shaft. Thus, any circuit which is to operate a motor should be capable of supplying adequate current to the motor.

For a bidirectional motor driving configuration, we have to use a circuit which is able to change the direction of the current through the motor. Change in direction of current implies change in direction of rotation and hence reversal of direction of rotation is achieved.

Let us suppose the motor is connected between points A and B. now if the voltage at $A > B$ then motor rotates in one direction. If something cause voltage of $B > A$ then the current flows in the direction opposite to the above and hence, there is a reversal in direction of rotation. We keep one of the voltages constant, a reference value. Then the other voltage can be an input from the previous stage of the circuit, or a variable voltage provided by the user. This is the principle of operation of a complementary symmetry driver.

Hence our requirements for the driver could be listed as follows: It should be able to supply at least the maximum current drawn by the motor during operation.

It would be more convenient if it contained two op–amps, since two are needed, one for the reference voltage and the other for the input voltage. A popular driver op–amp that we found suited to these needs was the TLC272. The TLC272 has output current capability up to 1A, houses two such op–amps and operates at the low voltages we would be using in our system.[1]

---

[1] TLC272 Data Sheet is part of Appendix B

This circuit gives bidirectional motion, but it only operates at positive voltages. Between 0 and 7.5 volts, we get rotation in one direction, which reverses above this value from 7.5 to 15 volts. The dead zone, the voltage range for which motor does not respond is at and around 7.5 volts. The 1Ω resistor in series with the motor plays the part of a load.

5.3 Transducer

The position of the motor has to be relayed to the main PID loop as a feedback from the process. We know that every position of the motor should have a one-one correspondence with a voltage. One component which is capable of providing us with such equivalence is a potentiometer. If we are to use the motor shaft position itself as a position reference, then it would be difficult to demonstrate the precision of the system.

Thus we decided to use the motor shaft itself driving a bigger pulley, through a belt drive, which would be attached to the potentiometer, for the position. The position feedback for a particular angle would be given by the potentiometer only when its pulley reached that angular position. Because of the high pot pulley to motor pulley radius ratio, the potentiometer would take more time to attain its position and the position attained would be more obvious and easier to demonstrate. This potentiometer is connected to the extremes of the set point range, and hence the set point and potentiometer voltages are compatible.

In fact the set point range is kept slightly lesser than the feedback range. This is to ensure that the feedback transducer is never pushed beyond its physical limitations by the driver, and hence the operation is always along expected lines.

5.4 Why Use Non–Standard Components

We have considered a non standard driver–motor–transducer configuration here. Commonly used combinations in the industry involve use of a driver card with its own error correction algorithm such as PID fitted into it. The motor is also not necessarily a DC motor and hence the drivers needed become more and more complicated as we extrapolate the system to commonly used motors of higher rating, or AC motors. However, we have not ventured into that because the motor position control was an application built for the sake of demonstration. Other position control systems involve processes such as valves, and pneumatic actuators. In fact, a process equation can be easily derived for a standard motor as opposed to this much smaller demonstration oriented model. The parameters of a DC motor are discussed in brief to explain what goes into arriving at an equation for a motor.

Chapter 6

Monitoring Card

The monitoring card has the responsibility of making sure that the PID card is working on signals that are within its prescribed range. It has to check various signals that are important for the proper action of the motor control mechanism and take one of two decisions; whether to shut the system down or to switch cards. We can do this by monitoring signals from all the cards all the time. This approach is fine for a 2 or 3 card system. As the number of cards goes up the hardware required to keep a check on all signals at similar points on different cards starts to become more and more complex. Some time division multiplexing scheme may be required. A more practical approach to the problem is to monitor the active card. When an error is detected in the active card, the one which is currently supplying the motor, the monitoring circuits switch control to the other card. The erroneous card is now offline, so that it can be replaced by the operator as and when suitable.

Signals which are common inputs to both cards will cause undesirable outputs regardless of which card is online. Thus the remedy for such signals is to turn the output to motor off when a fault is detected. This is equivalent to shutting down the system, since the motor stops at present position when no energy is supplied to keep it running. We have chosen three signals which can indicative of the nature of the fault with the system:

1. Setpoint of the primary loop
2. Feedback from the transducer
3. Setpoint of the secondary loop

6.1 Setpoint Of The Primary Loop

This is the system setpoint. It is the voltage equivalent of the angular position of the motor. There are three ranges which we account for when deciding the limits of the setpoint:

Measurement range is the total range over which measurement can be performed in an ideal case. For a motor this range is 360 degrees.

Physical range is the range over which the transducer or the process can actually provide any measurement. A linear potentiometer can only travel 270 degrees, so even if the motor is capable of complete circular motion the physical range of the device on the whole is restricted to 270 degrees. It is smaller than or equal to measurement range.

Control range is the range over which we will allow the controller to give commands for angular position. The control range has to be smaller than both the physical and measurement ranges. Keeping the control range equal to the physical range implies that a control command slightly exceeding the physical range will put the device in danger of getting damaged. It also accounts for small calibration errors.

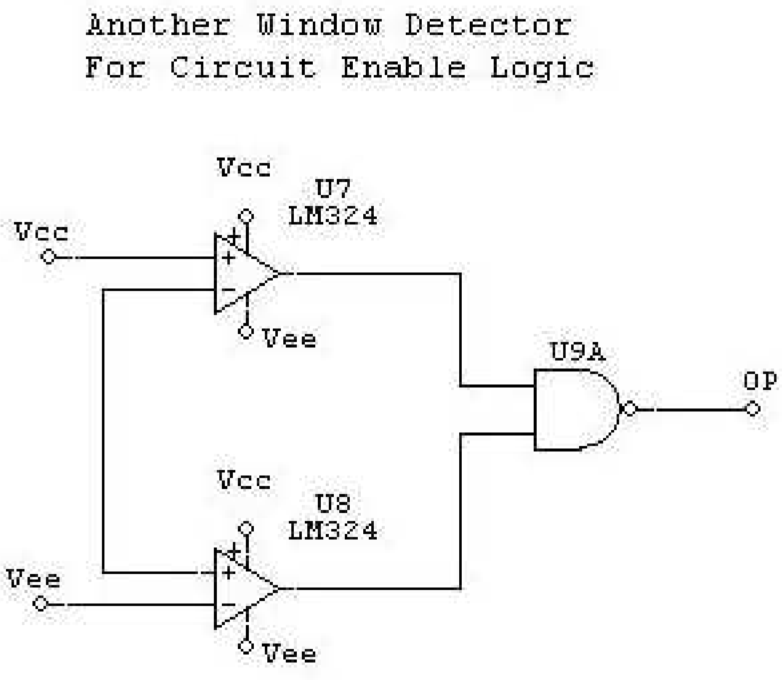


Figure 6.1: A Window Detector Implementation for the Monitoring Circuit

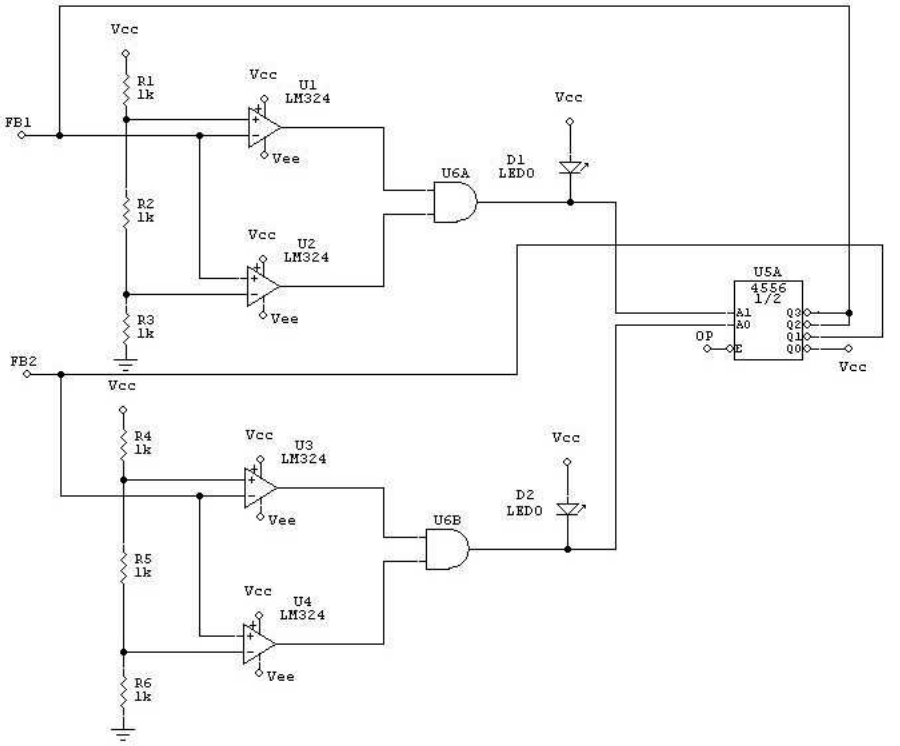


Figure 6.2: The Basic Monitoring Circuit

The setpoint is measured over control range. A simple window comparator circuit checks if the setpoint is within this control range. A low pass capacitive filter can be put into place just before giving the system setpoint. This prevents high frequency transients from raising a false alarm about setpoint being out of range. The action taken by the circuit is to disable the multiplexer[1]which passes its output to the motor. Setpoint range check error causes a system shutdown. This gives the operator an opportunity to correct the input source to the PID card and adjust it for range.

6.2 Feedback From Transducer

The transducer is a single turn potentiometer with a 270 degree range. Transducer feedback is essential for the primary feedback loop. If this feedback fails then the system must shut down immediately. Not doing so will result in an unwanted control signal since the configuration now becomes an open loop one. This is why we have provided for redundancy in feedback as well. There are two transducers which have the same ratio of pulley to shaft diameters, thus providing identical feedback at all times. The feedback from both sources is not given as a one–one correspondence to the two cards. Instead the feedbacks come to the monitoring card.

On the monitoring card are two identical range check circuits which see if the feedbacks are in range or not. We want the system to operate on the working feedback at all times. So if any of the two feedbacks is within range the system on the whole will still be operational. This is an OR logic function. The signals go to two points on the monitoring card:

---

[1] Circuit Implementation ensures that Analog input levels are compatible with digital outputs.

1. Window comparators to check the range
2. Multiplexer. The mux select lines are operated by the state given by the comparator circuits.[2] If the indication is that both feedbacks have failed (00 state), the main multiplexer which sources driver current to the motor is disabled and the system stops working. Thus, though each cards get a common feedback, the system has feedback redundancy.

6.3 System Shutdown

System shutdown has to be clarified in the context of the motor used. When the main multiplexer is disabled, the voltage across motor terminals drops to a level very close to zero. Now the motor operates at voltages greater than 0.5 volts. The 7.5 volt region is another voltage band at which there is no motion. This level can also be used at shutdown state. However it is more convenient to stick to a zero volt level since there is no motor motion at that voltage in any case.

6.4 Setpoint Of The Secondary Loop

This range check is a safeguard against the failure of the primary loop PID. Failure of the secondary PID will be indicated by the feedback range check itself. If the outer PID fails it will result in a control signal to the inner loop, on to the driver, and ultimately the motor which is not along expected lines. The inner PID takes care of current and load fluctuations and its corrective action does not consider what the intended position of the motor is. To let

---

[2] Circuit Implementation ensures that Analog input levels are compatible with digital outputs.

the inner loop make the correct calculation for driver input, its own input should be accurate and within bounds. Hence we check inner loop setpoint.

If the inner loop setpoint is out of range, the chances are that it is a card malfunction. Control is immediately given to the next card. If both cards malfunction then the system shuts down. The action taken on single card malfunction is to activate the error indication LED for that card and change the state of the switch or mux. A manual reset button can be used to turn the LED off when a faulty card is replaced.

Analog multiplexer is used to receive inner PID setpoints of both cards. Only the setpoint of the active card is checked for range. This is done by connecting both the multiplexers, one for supplying the motor and the other for receiving inner setpoint to the same switching logic. Both change state at the same time.[3]

This completes the description of the redundant system. Once the basic system is set up it is modelled and tuned for performance.

---

[3] Circuit Implementation ensures that Analog input levels are compatible with digital outputs.

Chapter 7

DC Motor Modelling And Tuning

There are various standard methods of determining the performance of a system. One of the easiest methods to visualise is that of judging the step response of a system. When we define a step $u(t)$, we are implying a sudden change in the input to a controller. The controller output is normalised to make for convenient representation on the same scale as the step input. The controller takes some finite time to attain its final position which is reached when its output steadies to a value close to the intended output. Between the application of an input to the attainment of a steady state, there are a number of variables which when observed help define the response. These variables constitute the model of the process. Tuning is the process of making changes to the values of these parameters till the response of the system is within defined limits. Standard process models can be tuned by open and closed loop step responses and tuning maps which give values of K T_i and T_d gains in terms of the parameters.

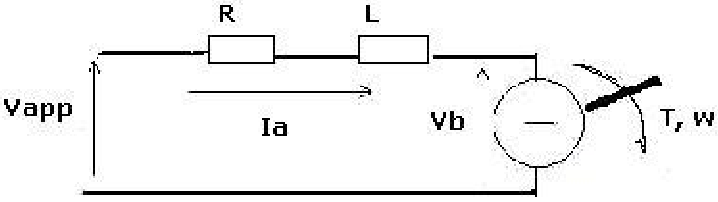


Figure 7.1: Eqivalents Circuit of a DC motor

7.1 Parameters Of DC Motor

To tune a process such as a DC motor, we need to know the equation which defines the motor mathematically. The DC motor transduces an electrical quantity into a mechanical quantity through interaction of two fields. One field is produced by a permanent magnet assembly; the other field is produced by an electrical current flowing in the motor windings. These two fields result in a torque which tends to rotate the rotor. As the rotor turns, the current in the windings is commutated to produce a continuous torque output.

Parameters of a motor are essential when we want to find the transfer function of a motor. Seven parameters are of importance in describing the functioning of the DC motor: K_b K_t R J_o L D and K_f The first few DC motor parameters are encountered in the electrical equation of the permanent magnet DC motor:

$$V_a = L\frac{\mathrm{d}I_a}{\mathrm{d}t} + RI_a + K_b\omega$$

- L is the armature winding inductance
- R is the armature resistance
- K_b is a measure of the voltage per unit speed generated when the rotor

is turning.

- K_t is the motor torque constant that is a measure of the torque per unit current produced by the motor.

The next three parameters are expressed in a single equation for the motor current in the frequency domain:

$$I_a(s) = (sJ_o + D)\frac{\omega(s)}{K_t}$$

- J_o is motor's moment of inertia.
- D is motor viscous friction (damping); a function of the motor's velocity and polarity.
- T_f is constant friction torque in the motor; a function of polarity. It is dropped during the derivation for linear motor transfer function, since it is a nonlinear entity.

The final transfer function can be shown to be:

$$G_m = \frac{\omega(s)}{V_a(s)} = \frac{K_t}{((sL + R)(sJ_o + D) + K_b K_t)}$$

Mathematical modelling is an elegant method of finding tuning parameters in a PID loop.

7.2 Measurement Of DC Motor Parameters

The parameters of the DC motor itself are such that they need to either be specified by the manufacturer or need testing procedures and equipment which are extremely difficult to realise using the motor available to us. DC motor parameters are required for a complete process model.

Terminal Resistance R can be determined with acceptable accuracy by measuring the resistance across the motor's terminal leads with a multimeter. A more accurate reading can be achieved after the motor has run for a period of time, and has adjusted to ambient temperature conditions. Typical values for DC motors range from 1 to 10Ω.

Back emf constant K_b and the torque constant K_t The back emf constant, also commonly called the voltage constant, can be tested and determined by running the motor as a generator and measuring the generated voltage, V_b, while measuring the shaft speed, n. The constant Kb is obtained from the following equation

$K_b = \frac{V_b}{n}$ [V/krpm; V,krpm]

The motor under test can be driven as a generator by coupling its shaft to another motor that is under speed control. A calibrated tachometer attached to the test motor's shaft, a hand tachometer, or a stroboscope can be used to determine the shaft speed, n. The benefit of testing for K_b is that it directly determines the torque constant K_t through the following relationships:

$K_b = \frac{V_b}{n}$ [V/krpm; V,krpm]

$K_t = K_b$ [metric units; Nm/A; V/rad s]

Mechanical and Electrical time Constants T_m and T_e If the mechanical and electrical time constants are significantly different in magnitude $T_m < 10\,T_e$ then the two time constants can be determined through frequency response tests. In addition, the mechanical time constant can be computed from a velocity step response in the time domain. Frequency testing requires a function generator and an oscilloscope, or a frequency analyzer such as the HP 35665A Dynamic Signal Analyzer. To identify the break frequencies, F_b in a frequency response test, the motor is excited with sine waves of unit amplitude over a range of frequencies. Stochastic test signals may also be used and have the advantage of easily spreading the excitation energy over a band of frequencies.

Friction terms T_f and D Motor torque losses are generally specified as static (breakaway) torques or dynamic (running) torques. Breakaway torque is a function of cogging (changes in magnetic circuit reluctance), brush friction, and bearing friction. These are affected by bearing type and preload, brush material and force, air gap flux density, and the magnetic circuit configuration. Breakaway torque may be 1.5 times the catalog motor friction value. The viscous friction term, D is related to no load current and rotational losses. If a motor is rotating with no load on its shaft, a small current will still be drawn. This can be tested for by adjusting the voltage in increments up to a rated speed , or the maximum safe speed, and recording the no load current. A plot of K_t x no load current versus speed can be made to determine the rotational losses. The slope of this line is the viscous damping factor, D in units oz–in/rpm.

7.3 Limitations Of Parameter Measurement For Non Standard Motor

There are two constraints on the estimation of the motor parameters for the motor we have used. A commonly available motor has been used to demonstrate the working of the controller. The specification for our design has been that of making a reliable system as opposed to a control system to suit one particular application. The first constraint is that of the unavailability of manufacturer's specifications for the motor. This is required for the estimation of different torques acting on the motor and indeed the value of viscous friction which depends on the motor friction value catalogued by the manufacturer.

The second constraint is that of equipment. Instruments such as tachometers and dynamic signal analysers were not at our disposal for the statistical analysis required to estimate time constants, back emf and torque constants. Terminal resistance of a motor is estimated more easily as $3 - -4\Omega$ which is a general value for the bidirectional motor we have used. Thus, we were not able to generate all the data needed to model the process completely. In the face of this, standard methods of tuning were not available to us. This led us to the use of a trial and error approach to getting the best result from the controller.

7.4 Tuning Of The Controller

Trial and error method involves continuously changing the parameters till a suitable response is obtained. The sequence of tuning generally remains the same regardless of the method of tuning used. When tuning a cascaded loop system the following procedure is generally followed:

1. The inner loop is first tuned using an internal setpoint. It is often tuned to critical or overcritical damping to reduce the oscillatory behaviour to a minimum
2. The inner loop, once tuned is cascaded with the outer loop. This con-

troller is given a setpoint equal to that of its process variable and that the control signal is equal to the setpoint of the inner loop.

3. The primary loop is set up such that it reacts to external setpoint changes and the inner loop is tuned such that its control signal does not exceed the limit of the driver input.

Our driver works to give a clockwise rotation to the motor for voltages between 0.5–7.5 volts and anticlockwise rotation from 7.5–15 volts. In reality there exists a small buffer region around 7.5 volt which is the dead zone for the driver. Primary loop output should never go outside these limits. Moreover, our attempt is to not let the actuator reach its performance limits in any case to avoid damage to the process or the motor. For this reason we prevent the control signal to driver going below 0.5 volt and above 14.5 volt. The inner loop gains are adjusted accordingly. Similarly the outer loop is tuned such that the motor does not physically exceed its operating limits on receiving system control command. This is taken care of by the setpoint monitoring feature explained in the section on monitoring card. The monitoring card makes sure that the setpoint itself is limited to well within the physical range of the transducer. Any change in setpoint range or driver used will imply a change in the tuning gains and the procedure will have to be repeated taking into account new constraints. The relationship between gains and voltage limits may be derived from tabulating a number of such tuning experiments.

Chapter 8

A User Interface

Installation and servicing are the two most important pre and post design routines respectively for any control system application. Before installation, we need to know if all the functions envisaged by the design are being carried out as expected by the hardware. During servicing, when the control system is detached from a previous stage, which is responsible for providing us with the setpoint, we require to run a diagnostic test to find out the exact nature and source of the flaw in operation. The monitoring of the system is a vital aspect of any critical application. The PID cards, driver and monitoring card have circuits which produce and monitor the position control action. As explained in the previous sections, checks such as limit checks are performed to keep the inputs to the PID within bounds. Based on the data from these checks, the monitoring circuit decides if the controlling card needs to be changed by switching from output of the present card to that of the redundant card. Before putting our system into use right away, we need to have some means to find out if the operations of switching in case of limit crossing and turning off the system in case of a potentiometer failure are being carried out.

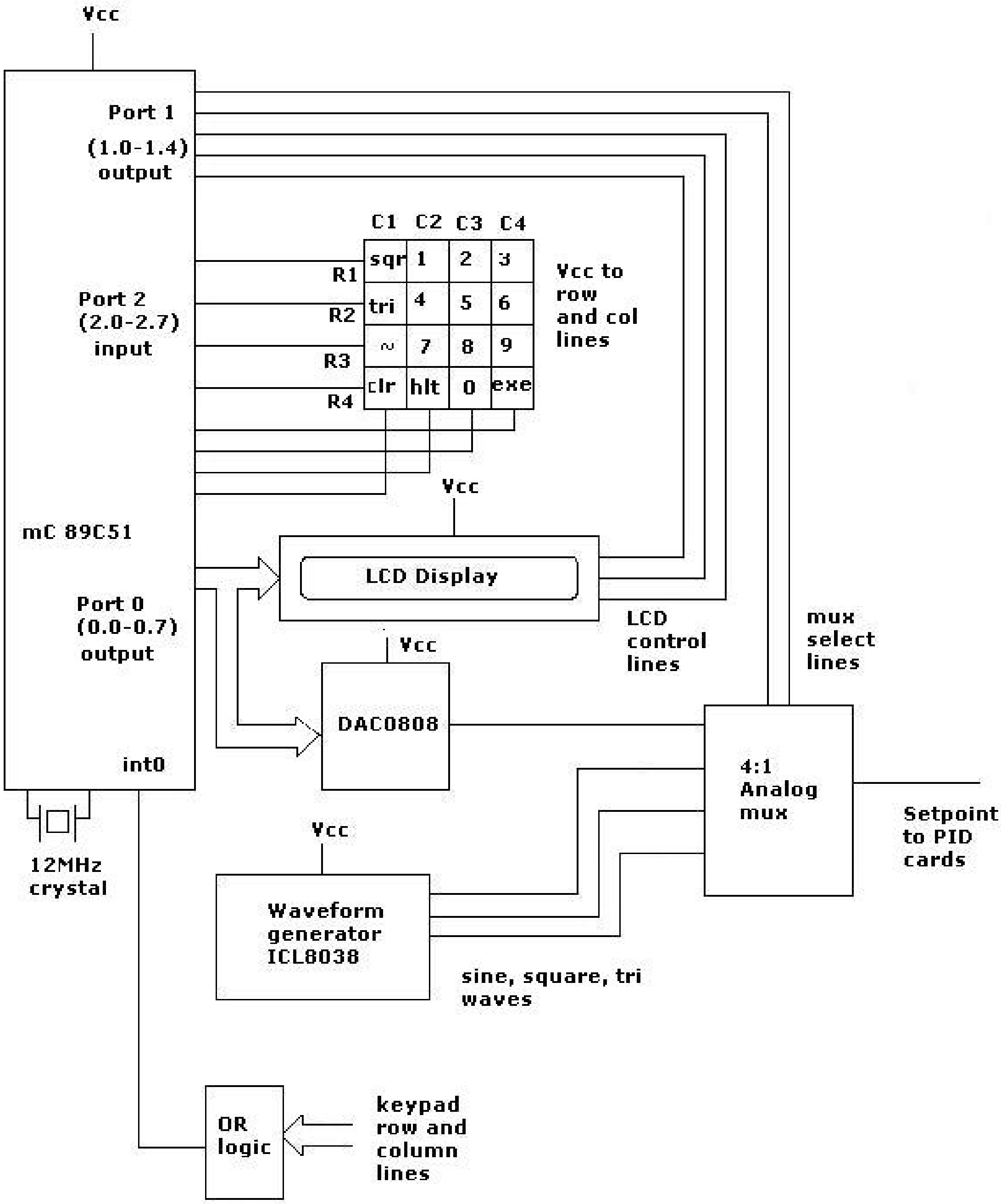


Figure 8.1: The UI Layout

This facility is provided by the user interface card. The UI is standalone or card based interface to the controller, which can be used for tuning and error location during installation and servicing. This interface is user serviceable, which implies that an operator handling the system can manually decide the waveforms which will help during troubleshooting or asserting the quality of control provided.[1] The ultimate goal is to bring the device to a fixed angular position, so there must be some way for the operator to input an angle value and see if the final position corresponds to that particular value.

The above two requirements of this card, testing using waveforms and position input to position output, can be fulfilled if we have:

1. A waveform generator to generate standard test waveforms, such as triangular, square and exponential.
2. A keypad matrix and display from where the function can be decided and angle value can be entered. The keypad matrix and display can be implemented using a series of digital components such as buffers and memory elements. However it is muchmore convenient to use a microcontroller for the same purpose.
3. The microcontroller allows us to have a flexible algorithm for the working of the user interface.
4. The microcontroller also allows for software key debounce, which preventsa key from being read more than the number of times it is intended to be read. This effect is caused by the switching transients which accompany the mechanical action of pressing and releasing a key. The 8051 is an ideal architecture for our use, since it has input output ports which can be easily interfaced with both a keypad matrix and display.

---

[1] Analog devices have input levels compatible with Digital levels

However the 8051 does not have sufficient on–chip memory. Hence we use the 89C51.

89C51[2]

- 4K of Flash erasable and programmable ROM
- 3 8 bit bidirectional ports
- On chip oscillator which operates up to clock speeds of 25 MHz. We are using a crystal frequency of 12MHz. The 89C51 is interfaced with three hardware components:
- Keypad matrix
- LCD display
- DAC0808

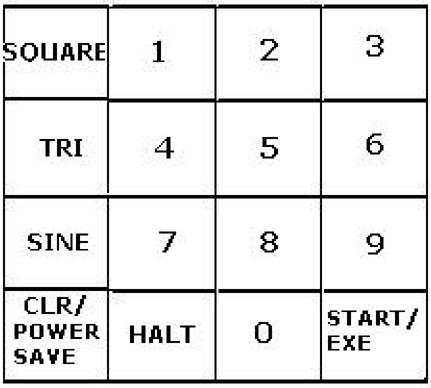
8.1 Keypad Matrix

Keypad uses single key operation for generating continuous waveforms and numeric keypad with keys for Clearing and accepting user input data. The keys are divided into two types, control keys (Clr/PowerSave and the START/EXE key) and the data keys (0. . . 9, SQUARE, TRI, SINE) and the HALT key. Control key START/EXE should always be the first key in any command execution. Only those inputs which begin with this key press are treated as valid inputs by the microcontroller.

The data keys are for inputting the angle value. This angle value is later converted to a setpoint, which is an analog level, by the DAC. Valid range

---

[2] 89C51 Data Sheet is part of Appendix B



SQUARE	1	2	3
TRI	4	5	6
SINE	7	8	9
CLR/ POWER SAVE	HALT	0	START/ EXE

Figure 8.2: The Keypad Layout

for the angle value is from 000 to 180, the numbers standing for the angle value in degrees.

The SQUARE, TRI and SINE keys are the means for selecting which of the continuous waveforms available is required at the setpoint of the PID cards. They pass the output of that particular waveform generated by the signal generator to the input stage of the PID.

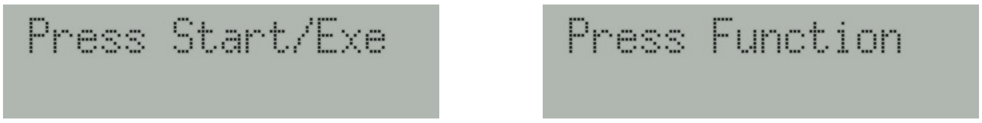
Examples of input key sequence :

(START/EXE) → (SIN) → (START/EXE)
Start a never ending Sine wave input to the controller.

(START/EXE) → (150) → (START/EXE)
Sets the Setpoint to 150 degrees.

(START/EXE) → (234) → (START/EXE)
Does nothing and displays that the angle is invalid.

HALT key switches the user interface off effectively. The UI draws no power when the HALT key is pressed. The microcontroller is switched off.



Press Start/Exe

Press Function

Display after Initialisation or Restart

After Pressing the Start/Exe Key. . . Waiting for input

60
Start/Exe to Run

Press Clr to Fin

After Entering an angle of 60°

On Pressing Clr, the controller resets

Figure 8.3: LCD Display for a Sequence of Sample KeyPresses

The HALT key is of use when the system has to be cascaded with an external source for setpoint generation. In such a case the user interface ceases to function and the external source takes over as the provider of the system setpoint.

8.1.1 Interfacing the matrix with the 89C51

The matrix is a 4×4 element card with slots for putting in the switches to the keypad. There is a line leading from each of the four rows and each of the four columns. A switch is located at the intersection of each row and column line, that is at matrix coordinates(x, y). The columns are connected to output pins, and the rows are connected to input pins.

Each column is sequentially driven to a low voltage while at the same instance the four rows are sampled. Since the rows are all held high with pull–up resistors, all four inputs will normally be high. If a key is pressed in a column which is at a low level, that low level will be conducted to the input pin through the closed key and the corresponding row will be sensed as a low. This is called the sampling method of keypad matrix interfacing. Pull up resistors are used at the lines where a V_{cc} level is applied. The interval between two keys being read is decided by the software routine and constitutes software debounce.

The second method of reading a key is the interrupt method which we have used. In this scheme a key is read due to interrupt int0 being activated. Int0 should be activated at any key press so that the microcontroller reads the key and its software evaluates it to decide the appropriate action. Hence we OR the row and column lines and then AND the two resulting lines. The output of this AND gate is the input to the int0 pin on the 89C51. The second interrupt int1 is unused.

8.2 LCD Display

The LCD display holds some significant advantages over the LED display. It allows character display which is difficult to achieve using LEDs. The LCD display also consumes less power than the LED display. We have chosen a character type Dot Matrix LCD module, which has a 2 lines of 16 character display area. [3]More remote locations may make it difficult to gain immediate access to the control system to find the nature of the problem immediately. In this case the LCD display can be used to display error messages indicating what has gone wrong with the system eg. Which card has failed, if there is a potentiometer malfunction, etc. Hence is the reasoning that a LCD display gives greater scalability for the project and at the same time is easy

---

[3] Data Sheet is part of Appendix B

to program. It is controlled by address lines coming from the microcontroller. Each address corresponds to a unique character. The input levels are also compatible with the microcontroller.

8.3 The DAC0808 DAC

The DAC0808 is an 8-bit monolithic digital-to-analog converter (DAC) featuring a full scale output current settling time of 150 ns while dissipating only 33 mW with ±5V supplies.[4] Some of its other features include:

- Relative accuracy: ±0.19% error maximum
- Full scale current match: ±1 LSB typ
- Fast settling time: 150 ns typ
- Noninverting digital inputs are TTL and CMOS compatible
- High speed multiplying input slew rate: 8 mA/us
- Power supply voltage range: ±4.5V to ±18V
- Low power consumption: 33 mW at ±5V

It uses an internal R–2R conversion ladder. This 8 bit device accepts the input from the microcontroller after it has converted the input angle to its 8 bit representation and converts it to a voltage corresponding to the setpoint levels accepted by the system. Care is taken to ensure that the setpoint level does not exceed the range of values which are valid. The output of the DAC is a current level which can be easily converted to a voltage using a I–V converter.

---

[4] The DAC0808 Data Sheet is part of Appendix B

The hardware described above constitutes mainly the section where a user defined angle value is given to the card and an analog output is obtained from it. However, once the operator decides to test the circuit with a continuous waveform, we need the use of a waveform generation IC to deliver a continuous analog output. The chosen waveform generator is the most common waveform generation IC, the ICL8038.

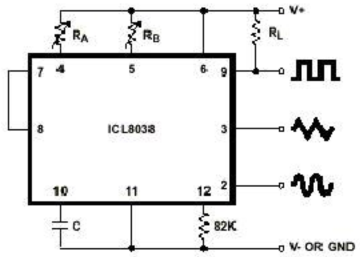
8.4 The ICL8038 Waveform Generation IC

The ICL8038 waveform generator is a monolithic integrated circuit capable of producing high accuracy sine, square, triangular, sawtooth and pulse waveforms with a minimum of external components. The frequency (or repetition rate) can be selected externally from 0.001Hz to more than 300kHz using either resistors or capacitors, and frequency modulation and sweeping can be accomplished with an external voltage.[5]

The ICL8038 is fabricated with advanced monolithic technology, using Schottky barrier diodes and thin film resistors, and the output is stable over a wide range of temperature and supply variations. These devices may be interfaced with phase locked loop circuitry to reduce temperature drift to less than 250 ppm/°C. One feature of the 8038 which requires additional circuitry on our part is that it continuously gives all the three wave outputs. This becomes an issue when we consider how the user interface works. The keypad operates in a logical fashion where it first checks what is desired by the user: an angle value or one of three waveforms. These are supposed to be exclusive–OR operations, only one of them actually providing any analog value to the final output point on the UI. However, the DAC and the 8038

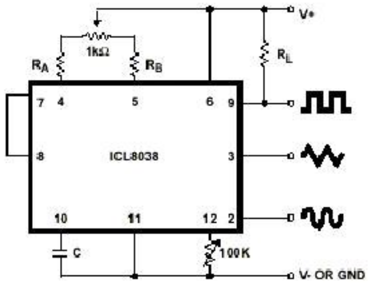
---

[5] The ICL8038 Data Sheet is part of Appendix B



$$t_1 = \frac{C\times V}{I} = \frac{C\times frac13\times V_S\times R_A}{0.22\times V_S} = \frac{R_A\times c}{0.66}$$

Figure 8.4: Example Configuration of 8038 Signal generator.



$$t_2 = \frac{C\times V}{1} = \frac{C\times \frac{1}{3}\times V_S}{2(0.22)\frac{V_S}{R_B}-0.22\frac{V_S}{R_A}} = \frac{R_A R_B C}{0.66(2R_A-R_B)}$$

Figure 8.5: Example Configuration of 8038 Signal generator.

are both operational as long as they are being run by a V_{cc}. This calls for an analog multiplexer just before the setpoint of the PID card. This will enable us to decide which of the four simultaneously available analog voltages is the one the user wants to select.[6] The 8038 output amplitude is a function of the value of V_{cc}. For a sine and triangular wave the line of symmetry is located at $V_{cc}/2$. the square wave has its high level at V_{cc} and the low level close

---

[6] Circuit Implementation ensures that Analog input levels are compatible with digital outputs.

to ground. If a dual supply is used, the waves are symmetrical about V_{cc}, which is advantageous for a system with a voltage range of both positive and negative values of setpoint.

The waveform generator has the concept of timing resistors deciding the frequency of the output wave. R_A and R_B along with C fix the rise and fall time of the waves. For the square wave these can be referred to the 1 and 0 state times. Two schemes of connecting the timing resistors are used.

Best results are obtained by keeping the timing resistors R_A and R_B separate. R_A controls the rising portion of the triangle and sine wave the 1 state of the square wave. The magnitude of the triangle waveform is set at 1/3 V Supply; therefore the rising portion of the triangle is The falling portion of the triangle and sine wave and the 0 state of the square wave is:

A 50% duty cycle is obtained with $R_A = R_B$. A slightly more convenient approach is adopted when a 50% duty cycle is needed all the time. This is what we need, since an asymmetrical waveform is not of much significance.

A 1k potentiometer may not allow the duty cycle to be adjusted through 50% on all devices. If a 50% duty cycle is required, a 2k or 5k potentiometer should be used.

This completes the summary of the hardware components on board the user interface. Once all of these are in place, it is the job of the software to coordinate the action of these components and ultimately decide how the card works.

8.5 Software

The primary function of the software is to allow easy control over establishing the setpoint of the controller for the purpose of tuning the controller. To accomplish this:

- It uses software key debounce routines.
- Function tips, depending on the key pressed on the LCD
- Power save mode, when there is an inactivity of greater than 15 seconds or power save using the Keyboard.

The program was compiled using a cross compiler which can generate code compatible with Intel 8085 family of microcontrollers. The source code was written in C language.[7] This implementation provides us with the flexibility and the ease of programming inherent to higher level languages as compared to assembly.

Main Routine

```
01: Power ON
02: Initialise LCD display.
03: Initialise a 15 second Timer and wait for it to expire.
    When time expires, the UI goes into Power save mode and
    all functions are suspended. LCD is switched off.
    The uC only wakes up on a key press, this key is IGNORED.
04: Halt the uC.
```

Key Press

```
01: Restart the 15 sec Timer.
02: If we are waking from power save, ignore key, goto 01:

03: See if the sequence is (Enter/Exe)[NNN](Enter/Exe)
```

---

[7] C language source code for the program is part of Appendix A

```
    [NNN] => [N] OR [NN] OR [NNN], Where (N = 0..9).

       01:if NNN <= 180
            01: Execute the command.
            02: Display command accepted.
            03: Exit this sub--loop.
       02: Display error.
       03: Exit this sub--loop.

04: See if the sequence is (Enter/Exe)[Wave](Enter/Exe)
    [Wave] is Sine or Tri or Square.

       01: Execute command and display.
       02: Exit this sub--loop.
```

Chapter 9

Further Developments

The PID controller itself will remain unchanged in other applications except for the tuned values of the components, which change according to the process. However, we might in certain cases not need the I or D amplifier in the design. A switch may be provided in this configuration to keep that particular amplifier out of the circuit without having to hardwire an altogether different circuit for the same.

9.1 A Scalable Model For Greater Redundancy

As the discussions on the project have highlighted, system redundancy is the most important design feature in the project. The monitoring card has combinational logic on board to switch between cards. This is easy to implement for a two card system. For higher number of cards, we need to have logic gates working in tandem to coordinate the switching. The multiplexers used also need to be able to accommodate a higher number of lines. As the card number increases we require hardware like a counter to select the combination to let the required output pass through to the next stage.

If a large number of redundant cards have to be used in the system, it may be a better option to use an alternative to the multiplexer. Cascading all the redundant cards in series may be the solution. These cards could pass their output to the driver on a priority basis. The card physically farthest from the driver could have the highest priority. This card would get the set point. If that card fails, then it transfers the set point to the next card without processing it in any manner. This continues to the last card in the circuit.

Now, any card just needs to know if the card of the higher priority attached to it is working properly. Hence, even if we use, say 10 cards in a redundant configuration, each card needs only a one bit signal, a yes or no answer about the health of the immediate higher priority card, to take control of the output. Card replacement will become easier, since there will be no need to reset the system after a replacement.

The monitoring card will cease to exist in such a system. All the monitoring will be on board for every card. This is the trade off for using a series arrangement as opposed to a parallel one. The hardware per card will go up substantially with the monitoring being done on a per card basis. Hence card replacement becomes much more costly than what it is for a parallel redundant system.

9.2 A Bus–Based System

One of the noticeable aspects of this system is the number of common signals which have to be passed to different cards. The set point is a common input to cards and so is the feedback. 5 or 12 volt supplies are required by different

ICs. Different monitoring signals are passed between the monitoring card and the PID cards. The user interface is an optional feature. The usability of the system goes up if we have a common structure on which all of the cards can be housed and then each signal travels on a standard path.

Such a structure can be envisaged to be similar to a motherboard with slots in which cards can be inserted. The tracks etched between cards will have a fixed bus–like layout where the signal on every line of the bus is predefined. So a card replacement will not involve additional effort in wiring everything together. A card will only have to be inserted into its slot. Replacing even vital cards like the driver and the monitoring card will be much simpler.

9.3 Provisions For A Standard Driver

Standard drivers available in the market have three or four different standards to which they confirm to. They accept inputs in ranges like 4–20mA, 2–12 volt, ś15 volt. All applications in industry related products will use drivers with these standard input specifications. Hence an additional card may be designed which will be connected to the PID card. The circuits on this card will convert inputs from the inner PID loop to inputs in the acceptable driver range. The output of this card can be tapped as per the driver used.

9.4 A User Interface With Greater Functionality

Presently, our user interface provides the user with an opportunity to select an angle value for the sake of checking angular precision, or selecting a standard waveform which can be used to gauge response of the circuit. To make the interface more interactive and to increase the user friendliness of the system, a set of signals can be taken from the monitoring card. These can be interpreted and if an error condition occurs, that error can be displayed on the LCD display. External set point, feedback and other parameters which could be of use could be displayed and logged for future reference. If it is feasible, some sort of self test routine can be run by the user interface on the rest of the system which will make the diagnostic feature more automatic than before.

The above improvements can be made to the present model of the position control system.

Chapter 10

Conclusion

We were asked to design a high availability analog system which would control motor position using the PID algorithm. This project introduced to us the modular approach to planning and implementation of a project. An analog PID controller with redundancy was broken down into a set of smaller problems such as making the cards which would be integrated into the bigger system. These cards by themselves worked to drive the motor, calculate the PID control signal based on error signal, or provide a set of test signals for the convenience of the operator. The card circuits themselves were found to be little more than the comparators, adders, integrators or waveform generators which we had come across in engineering theory. Hence a system with its own specifications had shown us how analog circuits with different functionalities can be made to work under a single roof. Conversely, how a bunch of common circuits could be used to make a more versatile and complicated one.

Some constraints are not imposed by soundness of design or feasibility. Such constraints are those of availability of components, their economic viability, their overall effect on reducing complexity and resulting hardware.

Hence the time spent in searching for the right components was a learning experience by itself, since we found out about a much larger number of components we could not use, and why we could not use them. It also generally educated us on the various types of products available in the market, and the vastness of the field of electronic system design. We would like to conclude by stating that some of the best things we feel we learnt from this venture are what are not written about in this report. We would once again like to thank everyone who helped us along the way.

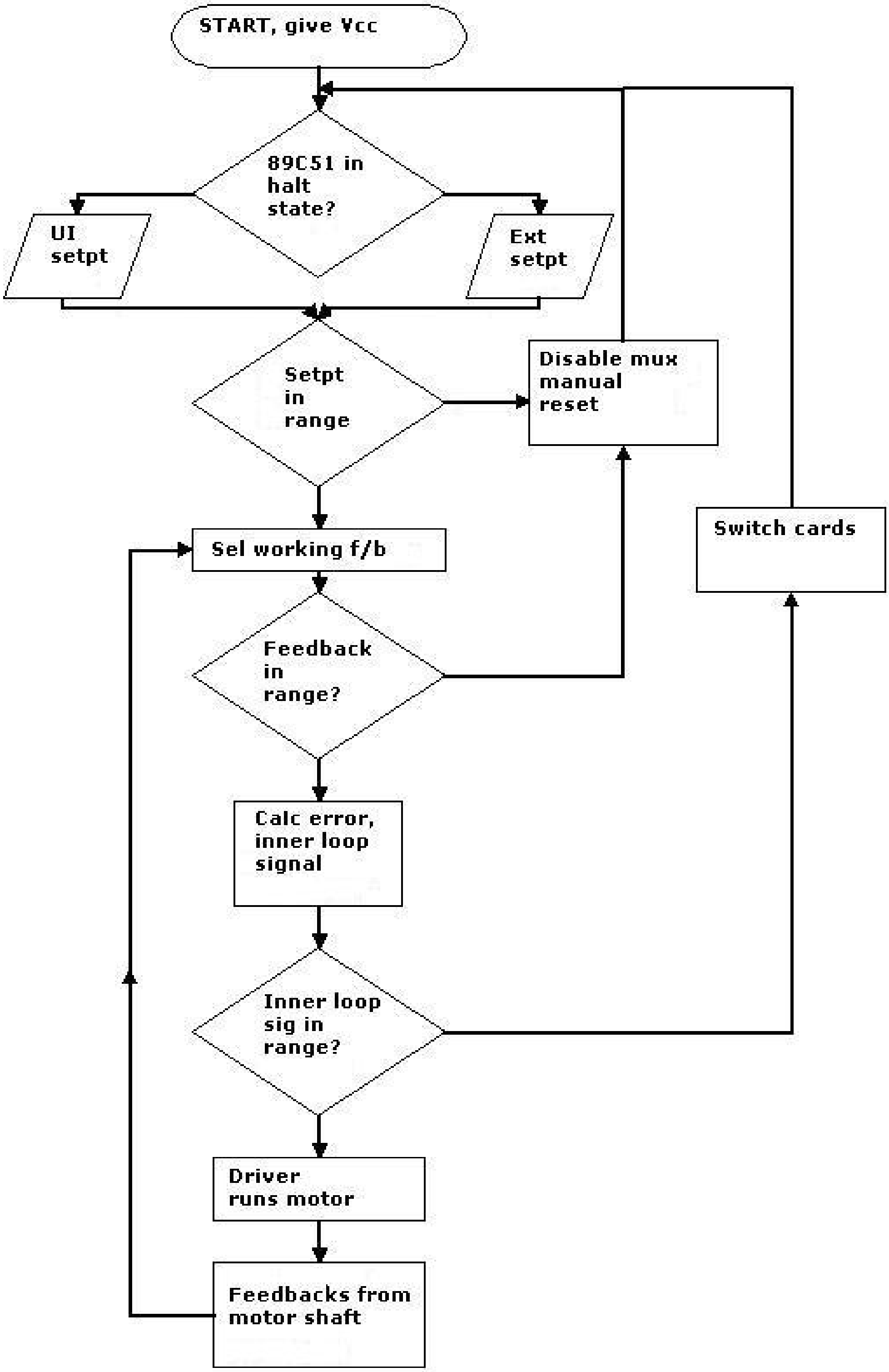


Figure 10.1: A Flowchart representation of the System

Appendix A

The C Language source code for the User Interface

This program was compiled using the Small Devices Cross Compiler on a GNU/Linux based system.

```
#include<at89x51.h>


void init_lcd( void );
void delay_us(unsigned int delay);
void delay_T0( void ) _naked ;
void write(unsigned char *ps) ;
//void Timer0_isr( void ) interrupt TF0_VECTOR _naked ;
void External0_isr( void ) interrupt IE0_VECTOR ;

#define LCD_DB P0

/******/
#define _LCD_RS P1_0
```

```
#define _RS_D 1
#define _RS_I 0

#define _LCD_RW P1_1
#define _RW_R 1
#define _RW_W 0

#define LCD_WI _LCD_RW=_RW_W ; _LCD_RS=_RS_I ;
#define LCD_WD _LCD_RW=_RW_W ; _LCD_RS=_RS_D ;
#define LCD_RI _LCD_RW=_RW_R ; _LCD_RS=_RS_I ;
#define LCD_RD _LCD_RW=_RW_R ; _LCD_RS=_RS_D ;
/******/

/******/
#define _LCD_EN P1_2
#define _HI 1
#define _LO 0

#define strobe _LCD_EN=_HI ; _LCD_EN=_LO ;
/******/

/******/
#define FS_8bit 0x10
#define FS_4bit 0x00

#define FS_2line 0x8
#define FS_1line 0x0
```

```
#define FS_5x11 0x4
#define FS_5x8  0x0
/******/

/******/
#define OF_gen 0x8

#define OF_show 0x4
#define OF_hide 0x0

#define OF_scur 0x2
#define OF_hcur 0x0

#define OF_blink 0x1
#define OF_solid 0x0
/******/

/******/
#define EM_gen 0x4

#define EM_inc 0x2
#define EM_dec 0x0

#define EM_son  0x1
#define EM_soff 0x0
/******/
```

```
#define CLEAR_disp 0x1
#define clear_display(port_add) \
        LCD_WI ;\
        port_add = CLEAR_disp ;\
        strobe ; \
        delay_T0();



#define home_add 0x02
#define go_home(port_add) \
        LCD_WI ;\
        port_add = home_add ;\
        strobe ;\
         delay_T0();



#define set_entry_mode(port_add) \
        LCD_WI ;\
        port_add = EM_gen | EM_inc | EM_soff ;\
        strobe;\
        delay_us(40);
```

```
#define  set_display(port_add) \
         LCD_WI ;\
         port_add = OF_gen | OF_show | OF_scur | OF_blink ;\
         strobe;\
         delay_us(40);



/*Cursor or Display shift*/



/*Function Set*/
/*Part of lcd_init as of now*/



/*Set DRAM Address                   */
#define goto_add(port_add,dram_add) \
        LCD_WI ;\
        port_add = 0x80 | dram_add ;\
        strobe ;\
        delay_us(40);
```

```
/* Read Busy Flag and Address Counter*/
#define read_busy() LCD_RI ; strobe ; delay_us(40);



/*Write Data to RAM                 */
#define write_data(port_add,data) \
        LCD_WD ;\
        port_add = data ;\
        strobe ;\
        delay_us(50);



/*Read Data from RAM                */
/*Not implemented                   */


#define KEY_DB P2
#define DAC_DB P0

#define _DAC_EN P1_3
#define _DAC_RW P1_4

#define WAV_SQR P1_5 = 0 ; P1_6 = 0 ;
#define WAV_TRI P1_5 = 0 ; P1_6 = 1 ;
#define WAV_SIN P1_5 = 1 ; P1_6 = 0 ;
```

```
#define WAV_DAC P1_5 = 1 ; P1_6 = 1 ;

#define WAV_ENA P1_7 = 1 ;
#define WAV_DIS P1_7 = 0 ;


const char _ready[]  = "Press Start/Exe";
const char _func[]   = "Press Function ";
const char _wait[]   = "                Start/Exe to Run";
const char _over[]   = "                Press Clr to Fin";
const char _sine[]   = "Sine";
const char _square[] = "Square";
const char _tri[]    = "Tri";

unsigned short int key ;
unsigned short int keycount ;
unsigned short int wave ;
unsigned short int angle ;
unsigned short int inout ;


void delay_T0( void ) _naked
{
_asm

  push IE              ;Save Interrupt Register
  mov  IE  ,0b10000010  ;Set EA,ET0
```

```
  push TMOD
  mov  TMOD,0b00000001  ;Set T0 Mode 1

  mov  TL0,0x00         ;We are sleeping for
  mov  TH0,0x80         ;32K cycles of 1uS, i.e. 32mS

  setb TR0              ;Start the timer
  setb IDL              ;Idle around

  pop  TMOD             ;Wake up and continue
  pop  IE               ;Restore Interrupt Register

  ret

 _endasm;
}


void init_lcd( void )
{
  /* Init routine as explained in the Data sheet */
  /* LCD_DB is the address of the the port where */
  /* the display is connected.                   */

  delay_T0();

  /* Set Function */
```

```
  LCD_DB = FS_8bit | FS_2line | FS_5x11;
  strobe;
  delay_us(40); /* we wait for 39 uS ( 40 actually ) */

  /* Bring on the lights */
  set_display(LCD_DB);

  /* Clear the Display */
  clear_display(LCD_DB);

  /* Entry Mode */
  set_entry_mode(LCD_DB);
}



void delay_us(unsigned int delay)
{
  for ( ; delay > 0 ; --delay)
    ;
}



void write(unsigned char *ps) {

  go_home(LCD_DB);

  goto_add(LCD_DB,0x00);
```

```
  while(*ps)
    {
      write_data(LCD_DB,*ps);
      read_busy();
      if(LCD_DB > 0x0f)
goto_add(LCD_DB,0x40);
    }

}

/*
void Timer0_isr( void ) interrupt TF0_VECTOR _naked
{
_asm
iret
_endasm;
}
*/

void External0_isr( void ) interrupt IE0_VECTOR {

  unsigned short int i,j,k;

  EA = 0 ;

  if(keycount > 5)
```

```
    return;

  key = KEY_DB;

  if(key & 0x7f)            /*Number Key*/
    {
      i=0;
      k=0;
      j=0;

      while(i < 8)
{
  if(key & 0x01)
  {
    if(i < 4)
    k = i ;
    else
    j = i ;
  }
  else
  {
    i++;
    key>>=1;
  }
}

      if(k != 3)
```

```
{
  i = ( 3*k + (7-j) );
  angle += ( 3*k + (7-j) ) * (0x01 << keycount);
}
      else
i = 0;

      goto_add(LCD_DB,(keycount-1));
      write_data(LCD_DB,(i+(int)'0'));

      DAC_DB  = angle;
      _DAC_EN = 1;
      WAV_DAC;

      keycount++;
    }

  if(key & 0x88)            /* Clr Key Pressed */
    {
      wave = key = angle = keycount = 0;
      WAV_DIS;
      write(&_ready);
      return;
    }
  if(key & 0x81)            /* Wave Type */
    {
      write(&_square);
```

```
      keycount ++;
      WAV_SQR;

    }
  if(key & 0x82)
    {
      write(&_tri);
      keycount ++;
      WAV_TRI;
    }
  if(key & 0x84)
    {
      write(&_sine);
      keycount ++;
      WAV_SIN;
    }

 if(key & 0x18)             /* Start/Exe has been Pressed */
    {
      if (inout)
{
  inout = 0;
  WAV_ENA ;
  write(&_over);
  return;
}
      else
```

```
{
  inout = 1;
  write(&_func);
  keycount ++;
}
    }
}

int main ( void ) {

 init_lcd();                /* Initialise the LCD Screen    */
 go_home(LCD_DB);           /* Goto First Line First Column */
 write(&_ready);
 return 0;
}
```

Appendix B

Data Sheets for Reference

Data Sheets of the following components are included for reference to the pertinent sections in the report.

1. **T**CL272 – DC motor Driver IC
2. **8**9C51 – 8051 based uC used for User Interface Card
3. **J**M162 – LCD Dispaly modules used for the User Interface
4. **D**AC0808 – Digital to Analog Converter IC
5. **I**CL8038 – Waveform Generator IC

Bibliography

Books

Norman A Anderson , Instrumentation For Process Measurement And Control, Chilton Company, (1996).

Karl Astrom , Pid Controllers, Instrumentation Society of America, (1995).

B. Wayne Bequette , Process Control- Modeling, Design And Simulation, Eastern Economy Edition, (1999).

Paul Horowitz , The Art Of Electronics, Winfield Hill Cambridge Low Price Editions, (1995).

Curtis D Johnson , Process Control Instrumentation Technology, Prentice Hall Publications, (1994).

David F Stout , Handbook Of Operational Amplifier Circuit Design, McGraw-Hill, (1995).

Documents

- 386 EX MATRIX KEYPAD DRIVER, JK Microsystems.
- MULTIPLEXING LED DRIVE AND 4X4 KEYPAD SAMPLING, Microchip Technology Inc.

- CHEMICAL ENGINEERING 162 – DYNAMICS AND CONTROL OF CHEMICAL PROCESSES, Regents of the University of California.

Webpages

PID tutorials

- http://www.jashaw.com/pid/tutorial/
- http://www.expertune.com/tutor.html
- http://www.shu.ac.uk/schools/eng/teaching/rw/pidmatlab.htm

Motor modelling and control

- http://www.hitex.co.uk/c166/pidex.html
- http://mechatronics.me.vt.edu/book/Section3/motormodelling.html
- http://hpme16.me.cmu.edu/matlab/html/examples/motor2/motor.html

DC motor parameters

- http://mechatronics.me.vt.edu/book/Section3/motormodelling.html

PID implementation

- http://hpme16.me.cmu.edu/matlab/html/PID/PID.html
- http://hpme16.me.cmu.edu/matlab/html/digital/digital.html
- http://newton.ex.ac.uk/teaching/CDHW/feedback/controltypes.html
- http://sdcc.sourceforge.net/
- http://www.8052.com/
- http://www.atmel.com/products/8051/