MODELLING OF A ROTATING PLATFORM AND ITS POSITION CONTROL USING VARIOUS CONTROL ALGORITHMS

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of
Dokuz Eylül University
in Partial Fulfilment of Requirements for
the Degree of Master of Science in Mechanical Engineering,
Machine Theory and Dynamics Program

by
Levent ÇETİN

June, 2001
İZMİR

M.Sc THESIS EXAMINATION RESULT FORM

We certify that we have read this thesis and **"Modelling of A Rotating Platform And Its Position Control Using Various Control Algorithms"** completed by **Levent Çetin** under supervision of **Prof. Dr. Erol Uyar** and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Erol Uyar

Supervisor

Yrd. Doç. Dr. Erginer UNGAN

(Committee Member)

Yrd. Doç. Dr. M. Evren TOYGAR

(Committee Member)

Approved by the

Graduate School of Natural and Applied Sciences

Prof. Dr. Cahit HELVACI

Director

ACKNOWLEDGMENTS

First of all, I would like to thank to my family, for their endless morale support and encouragement. It might be harder without them.

From the design step to realization, lots of unexpected things and events occurred, I certainly needed advice and guidance for both determining and eliminating them. I would like to thank to my supervisor, Prof. Dr. Erol Uyar, for his guidance and support throughout this study.

I wish to express my warmest gratitude to my friends, Alper Ayberk and Aytaç Gören for their help and special thanks to Engin Özçivici and Taner İzgi for their assistance during experiments.

ABSTRACT

This study describes the application of various techniques to control the position of a rotating platform which simulates motion of an airplane or a ship on the plane perpendicular to their movement axis.

The first step of the implementation is constructing an electromechanical model for experiments which consists of mechanical Simulator, a platform and a pulley and belt mechanism and a DC motor. Control system consists of data acquisition cards for data input and output and PC itself as a controller. There is a motor driver circuit consists a pulse width modulator to speed control and a half bridge circuit to drive motor.

The second part of the study is to settle a mathematical model for the system. This model helps to determine system characteristics and coefficients of the controllers. To obtain a mathematical model results of two experiments are used. The first experiment is determining belt velocity change against change in computer output and the other is calculating platform transfer function by the mediation of impulse response.

Third step is control applications, first a PID controller based on simulation is implemented and tuned then fuzzy controller algorithms; fuzzy PD, fuzzy P and fuzzy P+D is implemented and results are given.

According to results of experiments, fuzzy controllers have improved system performance then conventional PID controller especially from the point of time and noise sensitivity. Fuzzy P controller maybe a good alternative for controlling such systems.

ÖZET

Bu çalışmada gemi ve uçakların hareket eksenlerine dik düzlemdeki salınım hareketine benzerlik gösteren yatay eksenin ortasından yataklanmış bir platformunun pozisyonun değişik metotlar ile kontrolü incelenmiştir.

Çalışmanın ilk aşamasında uygulama için elektro mekanik bir model ve kontrol düzeneği kurulmuştur. Bu sistemin ana parçası bir platform ve üzerine yerleştirilmiş kayış kasnak mekanizması ve bu mekanizmayı hareket ettiren doğru akım motorundan oluşmaktadır. Kontrol düzeneği ise kontrol aygıtı olarak kullanılan bilgisayar ve model ile onun veri aktarımını sağlayan analog dijital ve dijital analog çeviricilerden oluşmaktadır. Ek olarak üzerinde darbe genişliği ile hız kontrol sinyali üreten devre ve motor sürücü devresi bulunan bir elektronik kart da kontrol sisteminde bulunmaktadır.

İkinci aşama da sistemin özelliklerinin ve uygun kontrol katsayılarının bulunması için bir matematiksel modeli kurulmuştur. Sistem elemanların teknik özelliklerinin bilinmemesi nedeni ile yapılan iki deney ile bilgisayar çıktısına karşılık kayışın aldığı yol ve darbe girdiye verdiği cevap yardımıyla platformun transfer fonksiyonu hesaplanarak matematiksel model oluşturulmuştur. Bir sonraki aşamada sisteme oransal integral türevsel kontrol simulasyon sonuçlarına göre uygulanmış ve gerçek sistem üzerinde kontrol katsayıları daha uygun değerlere ayarlanmıştır. Sisteme bunların yanında bulanık mantık ile kontrol algoritmaları da uygulanmıştır.

Deneylerin sonuçlarına dayanarak bulanık mantık kontrol algoritmaların nonlineerlikler içeren sistemlerde PID kontrolden –özellikle gürültü hassasiyetleri ve zaman açısından- daha iyi sonuçlar verdiği söylenebilir. Bulanık mantık ile desteklenen oransal kontrol algoritması bu tip sistemler için iyi bir kontrol seçeneği olabilir.

CONTENTS

Page

Chapter Four
SAMPLED DATA SYSTEMS

Chapter Five
MATHEMATICAL MODEL OF CONTROL SYSTEM

Chapter Six
PID CONTROL THEORY

Chapter Seven
PID CONTROL OF THE MODEL

Chapter Eight
FUZZY CONTROL THEORY

Chapter Nine
FUZZY CONTROL OF ROTATING PLATFORM

Chapter Ten
CONCLUSIONS

LIST OF FIGURES

LIST OF TABLES

CHAPTER ONE

INTRODUCTION

1.1 Introduction

As a typical balancing system, a rotating platform is used as a simulator for angular position control systems such as pitch control of an airplane. The natural dynamics of such systems are coupled linear equations subject to complicated friction and damping effect but on the model of the real system, performance and effectiveness of the control methods are verified clearly because of the simplicities of the structure.

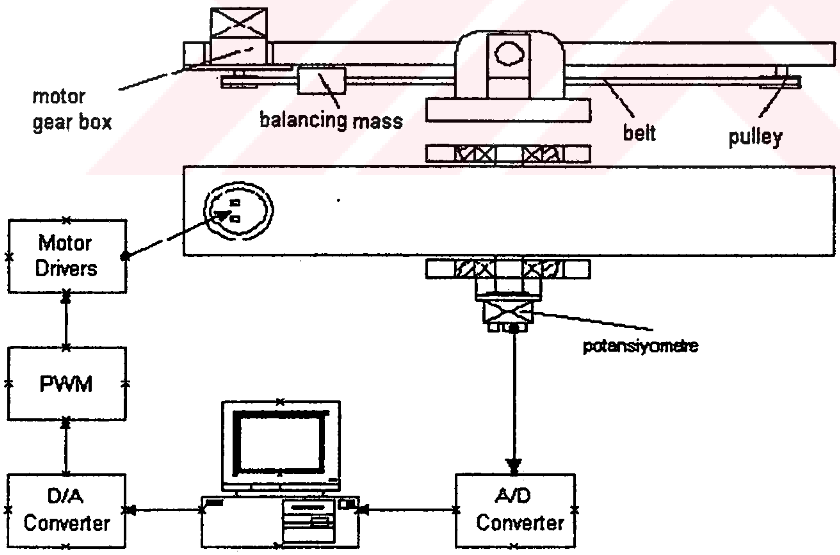


Figure 1. 1 Mechanical Model for Simulation

In this study computer based control system is developed in order to stabilize rotational motion of a simulator mechanism shown in figure 1.1. The simulator mechanism consists of platform with one degree of freedom (one axis rotation), and DC motor driven linear motion mechanism and an electronic card for driving motor. Control system also consists of data acquisition cards for discrete to analogue signal conversion for motor control voltage and analogue to discrete signal conversion for reading feedback values from potentiometer coupled to platform and PC itself as a controller. A schematic representation of this communication system is given in figure 1.2.

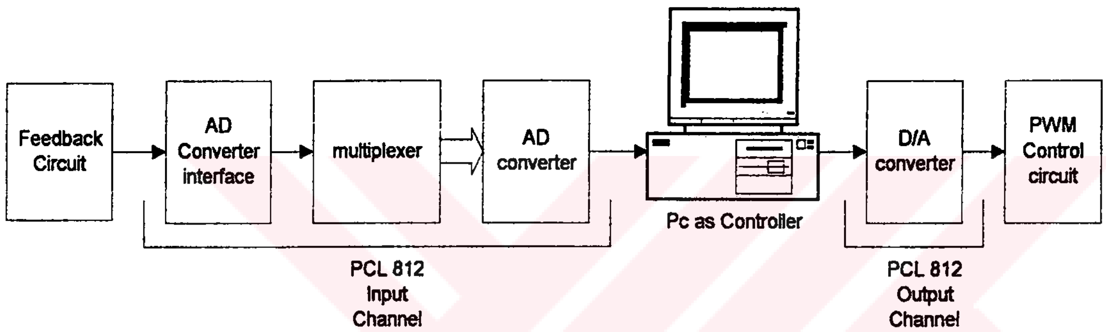


Figure 1.2 Feedback and control circuits with PC and I/O modules

For pulse width modulation, which is chosen for driving DC motor, an electronic circuit is designed. The purpose of this circuit is to demonstrate the basic principles of using an half bridge and PWM to control the speed of a DC motor. L298 from National Semiconductor is used as the half bridge. The L298 uses three digital control signals to control the turn on and off of those four switches of the half bridge. Details can be consulted from the chapter 3 to find out various control types that are possible through changing the logic status of those digital signals. Software compatible with motor characteristics and integrated circuit has been written in C++ programming language.

Classical automatic controllers are used in many manufacturing processes. These are generally of the PID type because they are standard industrial components. Moreover, owning to modeling uncertainties, a more sophisticated control scheme is not necessarily more efficient than a well-tuned PID controller. Alongside the advantages, however, the problem of tuning PID controllers has remained an active research area. Since the early work of Ziegler and Nichols, many techniques have been proposed for the manual or automatic tuning of PID controllers. The majority of the regulators used in industry are tuned using frequency response methods, because the modeling errors and the application specifications can be expressed directly in the frequency domain. Because of the nonlinearity effects, it is generally difficult to obtain an exact mathematical models of plant. Additionally, like the auto-tuning technique, frequency response methods are difficult to apply to the nonlinear system. In this study, first a mathematical model is settled then optimal gains are calculated by the help of Ziegler Nichols tuning rules. Afterwards, these theoretical gains are used in controller algorithm better system responses obtained by hand tuning techniques.

Despite a lot of research and the great number of different solutions proposed, most industrial control systems are based on conventional PID regulators. However, PID controllers cannot provide an optimal solution to all control problems. The processes involved are in general, complex and time-variant, with delays and non-linearity, and often with poorly defined dynamics. When the process becomes too complex to be described by analytical models, it is unlikely to be efficiently controlled by conventional approaches. In this case, a classic control methodology can simplify the plant model, but not provide good performance. Therefore, an operator is still needed to have control over the plant. Human control is very dependent on an operator's experience and qualifications, and as a result, many PID controllers are poorly tuned in practice. A quite obvious way to automate the operator's task is to employ an artificial intelligence technique. Fuzzy control, occupying the boundary line between artificial intelligence and control engineering, can be considered as an obvious solution, which is confirmed by engineering practice.

Fuzzy logic control is one of the most interesting fields where fuzzy theory can be effectively applied especially in systems which involves nonlinearities. Fuzzy logic techniques attempt to imitate human thought processes in technical environments. In doing so, the fuzzy logic approach allows the designer to handle efficiently very complex closed-loop control problems, reducing, in many cases, engineering time and costs. Fuzzy control also supports nonlinear design techniques that are now being exploited in motor control applications. Software based implementations of these techniques have been mainly used in industrial automation for relatively slow processes. Fast fuzzy control usually requires the use of a specific hardware processor.

Initially fuzzy control was found particularly useful to solve nonlinear control problems or when the plant model is unknown or difficult to build. In this study it will be shown that these techniques can also be useful in applications where classical control performs. Fuzzy Logic allows a simpler and more robust control solution whose performance can only be matched by a classical controller with adaptive characteristics, much more difficult to implement. This study reports work that is being done in order to apply fuzzy techniques to adjust the speed of a motor which also controls position of the rotating platform.

Gain scheduling is also common PID advancement used in many applications to overcome nonlinear process characteristics through the changing controller gains over local operating bands. This scheduling is complicated by the need for detailed process knowledge to define operating bands and open loop tests which must be performed to locally calibrate the controller gain within each band. An alternative method is fuzzy logic approaches have been shown to be a simpler alternative to improve conventional PID control performance. Performance improvements for such a problem are usually demonstrated by reductions in the amplitude of undesirable oscillations in the manipulated variable around the set point, shorter times to converge to the set point, and the maintenance of control stability seen in conventional PID control.

The layout of the study is as follows. The mathematical modeling is done as the first step for control algorithms (chapter 5).The control of the angular position process by conventional PID is achieved and results are given for various controller settings (chapter 7). A fuzzy "PD" like controller, which has error and change of error as input, fuzzy P controller and fuzzy P +D controller is implemented and results for different parameters are demonstrated (chapter 9).Comparison and conclusion of the experiments and suggestions for further improvement are given (chapter 10).

CHAPTER TWO

MECHANICAL COMPONENTS OF THE MODEL

2.1 Introduction

In this study a mechanical model for a rotating platform which simulates rotation of planes and ships on the plane perpendicular to their motion axis is designed and constructed. It is necessary to control its angular position in order to maintain good moving capability and comfort of people.

Mechanical model consists of two parts: Rotation Simulator and Control Mechanism. Rotation simulator is a metal printer head motion mechanism, which is mounted on the middle and has holes to put a metal block to obtain disturbance torque. Control mechanism is a linear motion mechanism consists of DC motor-coupled gearbox with a belt-pulley connection which drives a balancing mass (figure 2.1).

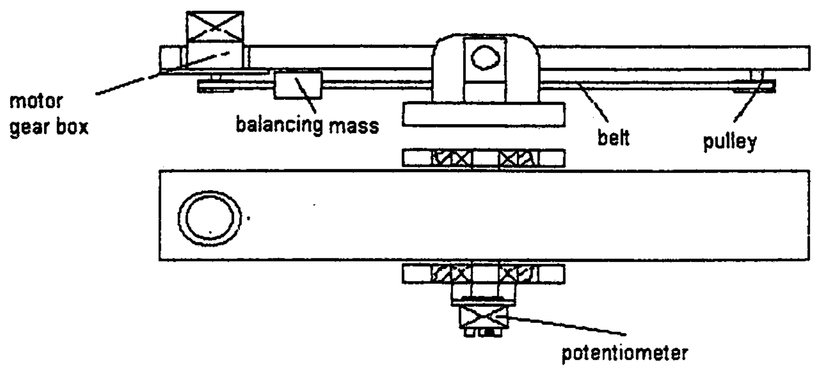


Figure 2.1 Schematic Representation of Mechanical Model

This simulator is aimed to balance the platform excited by a disturbance torque which is simulating huge waves or powerful winds.

2.2 Platform

Rotational Simulator is a platform which is mounted in the middle of its longitudinal axis. This mechanism has only one degree of freedom, that is rotation on mentioned axis. There is a pulley and belt mechanism, that drives a sledge and balancing mass, reciprocally along the platform. This mechanism is driven by a 12V DC motor. There are also holes drilled constant distances for putting another block on it. This block simulates disturbance, and also input of the simulation. And there is a single turn linear potentiometer (270°) as angular displacement sensor

Potentiometers are variable resistors. They have three terminals with the center terminal being a center tap contact that slides across an element of constant resistance. A rotary potentiometer can be used to measure the rotation of a shaft. It is easiest to use if the shaft being measured does not need to rotate continuously, but rather would rotate back and forth. Use of gears is a simple way to lock the rotation of the shaft being measured to the potentiometer. By using a gear ratio other than 1:1, a shaft that needed to rotate more than the 270 degrees or so of the pot could be measured.

2.3 DC Motors

Because of the simple reverse drive and good torque speed characteristic, a DC Type of motor is chosen for control application. The used DC motor consists of two parts: Stator and Rotor. Usually, the Stator is the fixed outer part which rotor rotates inside.

DC motors are not used in ordinary applications because it is much easier to use the AC current which is readily available almost everywhere. However, in special heavyduty applications like electric trains, steel mills, etc, the DC motors are preferred because their speed and torque may be easily varied without suffering a reduction in the efficiency.

A DC motor is driven by DC current supplied to the rotor windings. This is called armature current. The rotor is placed inside the magnetic field of the stator. The interaction between the current-carrying armature windings and the electro magnetic field of the stator produces a force (Lorentz Force) that rotates the rotor. The stator field is maintained by either permanent magnets or by a field current passing through the stator windings (field current).

In normal mode, a DC motor converts the electrical (armature current) to mechanical (rotor torque). However, as shown in the figure (2.2), the same machine can be used in generator mode when the mechanical power is transformed back to electrical. regenerative braking mode, converting the potential energy of the bucket to electricity.

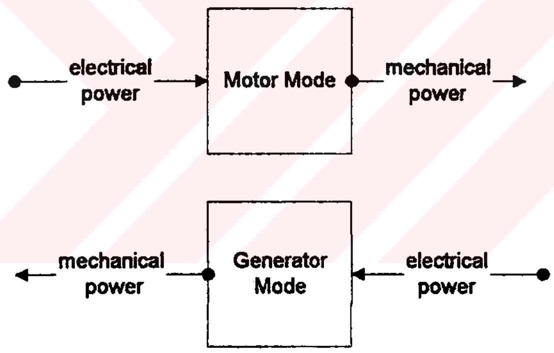


Figure 2.2 Operating modes of DC Machines

2.3.1 DC Motor Types

The armature and field circuits can be connected in a number of ways, leading to different types of DC motors:

Separately Excited DC Motors: The field and armature circuits are totally separate. The field current is supplied from a secondary source (or by permanent magnets).

Self-excited DC Motors: The field and armature currents are provided by the same source. These are usually smaller motors and the field and armature may be connected in three different ways:

- Shunt Motor: The field and armature windings are in parallel
- Series Motor: They are in series
- Compound Motor: Both shunt and series windings are used.

In a shunt motor, the field windings are "shunted" across the voltage supply to the armature. In other words, the field and armature windings are in two parallel paths as shown in the following figure (2.3): A shunt-connected DC motor is designed for a specific voltage and is rated to deliver a specific power at a specific speed. This speed is referred to as the base speed and it occurs when I_f is near its maximum value.

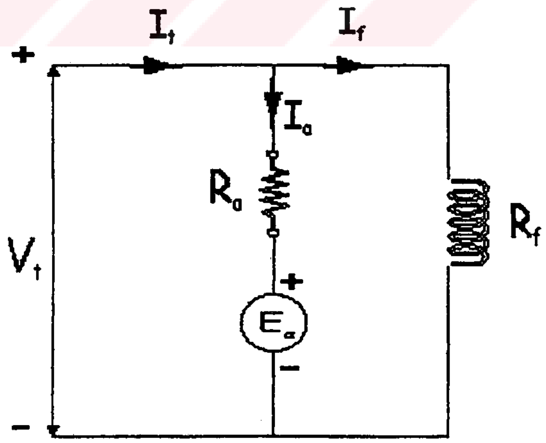


Figure 2.3 Electrical circuit of DC Shunt Motor

In the series-wound motor the armature and the field windings are in series. (figure 2.4) Compared to shunt motors, the field winding in series-wound motors are wound using wire of larger diameter. That is because the field winding sees the full load current in series motors. We will see that this arrangement gives excellent starting characteristics (high torque at the start) but tends to run away when the load vanishes.

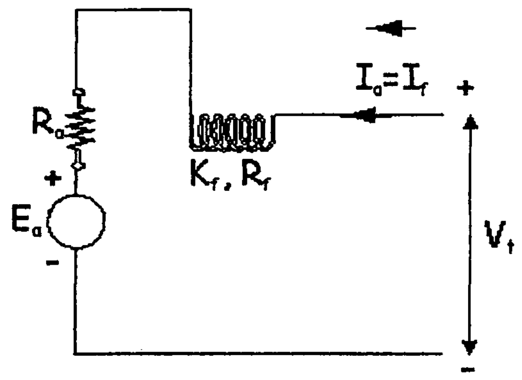


Figure 2.4 Electrical circuit of DC Series Motor

The advantage of a series motor is the very high torque available at start-up. This makes them particularly suitable for applications like trains, automobile starter motors, hoists, industrial mixers, etc.

In a Permanent Magnet motor a coil of wire (called the armature) is arranged in the magnetic field of a permanent magnet in such a way that it rotates when a current is passed through it. When a coil of wire is moving in a magnetic field a voltage is induced in the coil - so the current (which is caused by applying a voltage to the coil) causes the armature to rotate and so generate a voltage.

The value of the back emf is determined by the speed of rotation and the strength of the magnet(s) such that if the magnet is strong the back emf increases and if the speed increases, so too does the back emf. It follows from this that if you use a weaker magnet to make a particular motor, you will get a higher speed motor.

If you apply now a load to the armature, it will slow down. The back emf will decrease so the difference between applied voltage and back emf will increase. It is this difference that causes the current in the armature to flow - so the current will increase as you increase the mechanical loading. It should be apparent therefore that an unloaded motor will take little current. It should also be clear that if you apply more voltage the motor will speed up, apply less and it will slow: this is what the speed controller does: it varies the voltage applied to the motor. Speed control methods will be explained in the next chapter.

CHAPTER THREE

DC MOTOR SPEED CONTROL THEORY

3.1 Introduction

The second part of the model is electronic circuits. The control action which is planned to be occurred is speed control of the balancing mass according to derivation angle. Since it is obvious that analog outputs of the data acquisition cards are mostly within a range of −10 to 10V or smaller. Though it seems to be enough to control a 12 V DC motor, it is not possible to drive motors directly from the analog outputs because of the high operating and start up currents. Common way is to design a driver circuit between data acquisition card and motor.

3.2 Motor Driving Circuits

For designing driver circuits it is possible to deal with the problem by several technologies The DC motor speed can be controlled in a number of ways. The most common options are listed below:

- Field Control
- Armature resistance Control
- Armature Voltage Control
- Ward-Leonard Drive
- Solid-State Control

3.2.1 Controlling The Speed By Controlling The Field

It has already demonstrated that the speed varies with the inverse of the field:

$$\omega_m = \frac{V_t}{K_a\Phi} - \frac{R_a}{(K_a\Phi)^2}T \tag{3.1}$$

Therefore, for a given torque, the speed should decrease if the field is weakened. This increase in speed will, however, will come at the expense of the torque because the torque is directly proportional to the field.

For a shunt motor, the field is weakened by putting a variable field resistance in series with the field windings (figure 3.1)

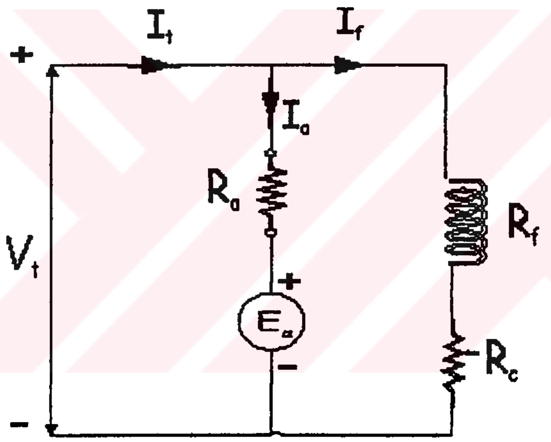


Figure 3.1 Field Control in a shunt motor

This circuit can be implemented as a resistance switching mechanism similar to the armature resistance switching during start-up. It is easy to do and inexpensive because the control is implemented at the low-power part (in the field circuit) of the system. However, because of the large inductance in the field circuit, the response to resistance changes will be slow.

The same field weakening effect can be obtained by in series motors by diverting some of the field current from the field windings (figure 3.2) In this instance, the control device is referred to as a field diverter.

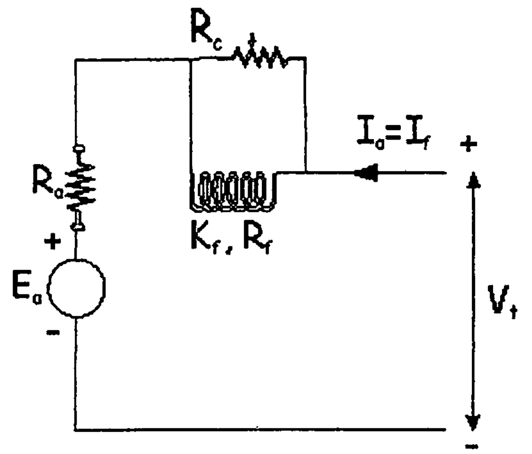


Figure 3.2 Field control in a series motor

3.2.2 Speed Control By Controlling The Armature Resistance

Changing the armature voltage is another way of controlling the speed. The easiest way to control the armature voltage is by connecting a variable external resistance in series with the armature.

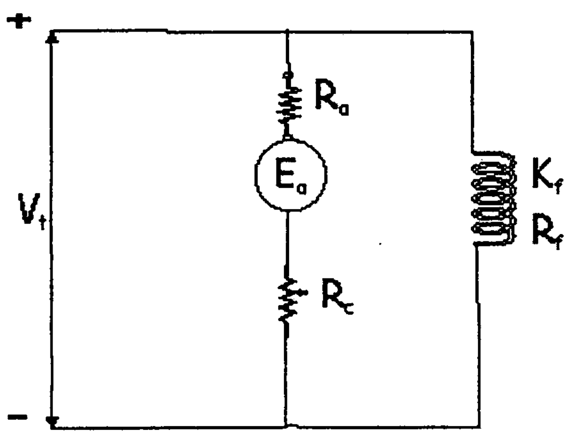


Figure 3.3 Speed Control by Controlling the Armature Resistance

This easy to implement. It is not efficient because of the insertion of a higher resistance into the armature circuit and the subsequent I^2R losses on that resistance.

As the armature resistance is increased, and assuming the supply voltage and the field stay constant, the new speed is calculated by the following equation where Ra+ Rc is substituted for Ra.

$$\omega_m = \frac{V_t}{K_a \Phi} - \frac{R_a}{(K_a \Phi)^2} T \qquad (3.2)$$

3.2.3 Controlling the Speed by Modulating the Armature Voltage

The most common method of controlling the speed is by directly controlling the armature voltage:

$$\omega_m = \frac{V_t}{K_a \Phi} - \frac{R_a}{(K_a \Phi)^2} T \qquad (3.3)$$

By changing the armature voltage from 0 to its rated value, the speed can be varied from 0 to the rated speed. The armature voltage control can be combined with field control to give the ability to control motor speed from 0 to several times its rated value. This is seen in the following chart:

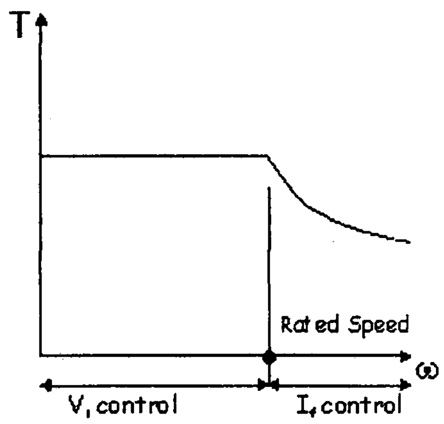


Figure 3. 4 Armature voltage control combined with current control

A potential divided is commonly used in shunt motors to provide the armature voltage control. The use is limited to smaller motors because of the high current ratings for larger machines make such dividers impractical. For larger motors, the same effect is achieved by a slightly more complicated Ward-Leonard drive system.

3.2.4 Ward-Leonard Systems

Ward-Leonard systems were introduced in 1890s. Schematically, the operation of the system is as follows:

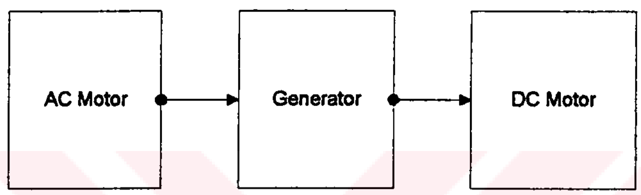


Figure 3.5 Ward Leonard system

The AC motor drives the generator. The generator generates the terminal voltage for the DC motor. This voltage can be modulated by modulating the field current on the Generator.

Ward-Leonard drive provides very smooth and reliable speed control with no loss of motor efficiency. They are highly complex systems, however, and this makes them suitable only for high-price high-quality applications. They are very common in mine sites driving shovels, mine cages and mills.

3.2.5 Solid-State Speed Control

Solid-state converters and rectifiers have become available in recent years even in high-power circuits. Such devices are gradually replacing the Ward-Leonard systems based on dedicated motor generator sets.

These controlled rectifiers are commonly referred to as Silicon Controlled Rectifiers or SCRs. By chopping the supply voltage, they produce a pulse train for the armature voltage rather than a continuous supply:

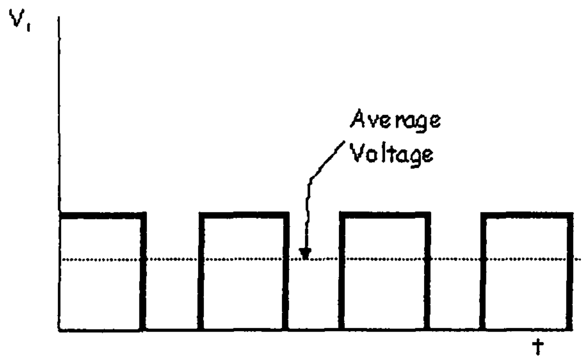


Figure 3.6 Pulse train

By modulating the pulse train frequency, the average armature voltage is modulated. The shape of the pulses is not always square. For example, a SCR chopping an AC signal to provide rectified excitation for a DC motor produces pulses of a half-sine shape.

The standard voltage-speed-torque relationships apply with the substitution of the average values:

$$V_t = E_a + I_a R_a$$
$$E_a = K_a \Phi \omega_m$$
$$T = K_a \Phi I_a \qquad (3.4)$$

3.3 Basics of Pulse Width Modulation

The control of electric motors is something which interests nearly everyone involved with model building. On the face of it, a simple controller with a voltage supply and a switch or potentiometer is sufficient. Every model, however, its own motor requirements with regard to the space available, the power of the motor, its speed, whether it must stop and start frequently, and the need for reduction gearing.

On the face of it, simple methods of control are perfectly adequate, with a regulated voltage supply, a simple on off switch, and the means to reverse the motor. Speed can be controlled with a potentiometer. In reality, this provides very unrealistic results. The main problem is poor starting performance, the motor tending to jump almost instantly from a stationary position to what is often more than half speed. The main cause seems to be the starting characteristic of the motor itself which when under load seems reluctant to start. A motor has a relatively low resistance when it is stationary. As the speed control is advanced, the current through the motor increases, but the voltage across the motor remains quite low. The speed control therefore has to be well advanced before the voltage and power fed to the motor are high enough to overcome its reluctance to start. As the motor speed and the load on it changes, there are changes in the internal resistance. Speed regulation is not very good under these circumstances, particularly at low speed

PWM is a common technique for speed control. A good analogy is bicycle riding. You peddle (exert energy) and then coast (relax) using your momentum to carry you forward. As you slow down (due to wind resistance, friction, and road shape) you peddle to speed up and then coast again.

The duty cycle is the ratio of peddling time to the total time (peddle+coast time). A 100% duty cycle means you are peddling all the time and 50% only half the time.

PWM for motor speed control works in a very similar way. Instead of peddling, your motor is given a fixed voltage value (say +5 V) and starts spinning. The voltage is then removed and the motor "coasts". By continuing this voltage on-off duty cycle, motor speed is controlled.

In the above diagrams, V is the voltage across the motor and t is time. By switching quickly, we can create an average voltage across the motor. The speed of the motor can be adjusted by changing the pulse-width ratio:

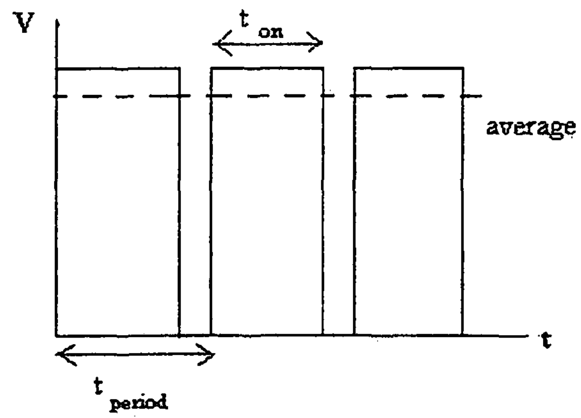


Figure 3.7 Pulse Width Modulation parameters

The concept of PWM inherently requires timing. The classic 555-timer chip, operational amplifiers and some potentiometers can be used to generate PWM. The pots are manually adjusted for the desired duty cycle. However, if a PC is used, it can automatically change the duty cycle and control your motor's speed.

3.4 General Description of Circuit

In this experiment, the PWM is used to control the duty cycle of PM motor armature voltage. A fifty- percent duty cycle is brake since effective voltage equals to zero and when it is less then 50 percent, the rotating direction is clockwise. Counterclockwise rotating direction is obtained with higher duty cycle.

Because the duty cycle determines the averaged DC voltage the armature winding of DC motor receives and the speed of the DC motor depends on the armature voltage, by changing the duty cycle of the PWM waveform, the speed of DC motor can be adjusted or controlled.

Electronic circuit for Pulse Width Modulation consists of three cascaded parts. The First is a precise timed trigger generator with LM 555 timer chip and the second is a sawtooth pulse oscillator with an RC and a transistor triggered by LM 555 output and the last one is an opamp circuit which compares our control voltage and sawtooth pulse then obtains PWM. At the output of this opamp there is a relay which controlled by one bit of the digital output channels to control the direction of the by applying modulated signal to the correct pin of H-Bridge.

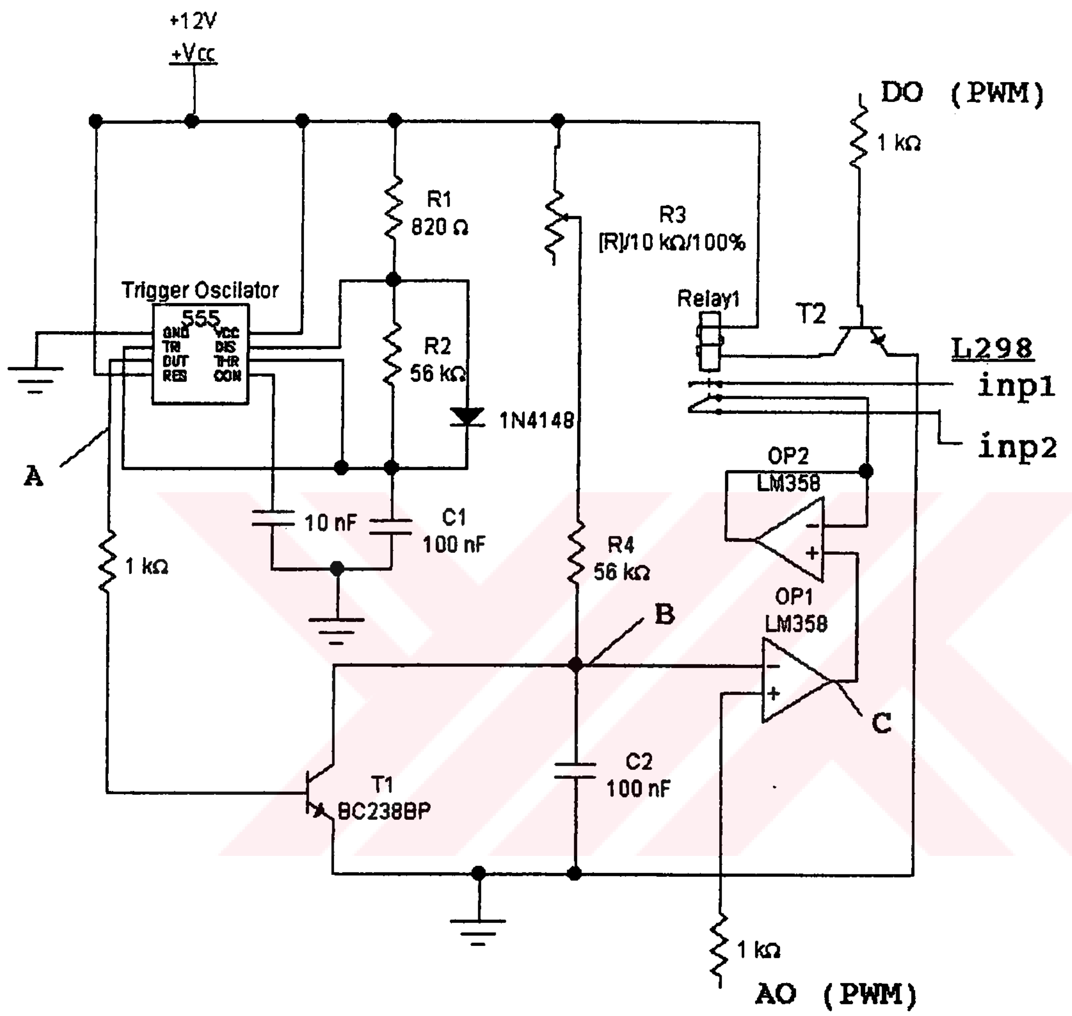


Figure 3.8 PWM Circuit for driving L298

3.5 Operation of Circuits

The first part of circuit is a trigger generating oscillator. A LM 555 Timer Chip is connected as in figure (pins 2 and 6 are connected each other) it will trigger itself and run as a multivibrator. External capacitor charges through $R_1 + R_2$ and discharges through R_1 .Thus the duty cycle can be precisely adjusted by the ratio of these two resistors according to given formulas. The charge time when the output is low is given by:

$$t_1 = 0.693(R_1 + R_2)C_1 \quad (3.5)$$

$$t_1 = 0.693.(820 + 56000).100.10^{-9}$$
$$t_1 = 3.938 \text{ ms}$$

And the discharge time when the output is high by:

$$t_2 = 0.693(R_1)C_1 \quad (3.6)$$

$$t_1 = 0.693.820.100.10^{-9}$$
$$t_1 = 0.0568 \text{ ms}$$

And the frequency of oscillation is:

$$f = \frac{1.44}{t_1 + t_2} \quad (3.7)$$

$$f = \frac{1.44}{3.938 + 0.0568}$$
$$f = 360 \text{ Hz}$$

This frequency determines the frequency of PWM signal.It must be chosen convenient to motor characteristics.

In this mode of operation, the capacitor charges and discharges between 1/3 V_{cc} and 2/3V_{cc} .The charge and discharge times are independent of the supply voltage.

Next part of the circuit a typical RC connection which is used for generating a almost linear sawtooth wave form. This wave form is used as the template of Pulse Width Modulation output.

When base of the transistor T_1, output of the LM 555, is in low state, the capacitor is begin to charge and the voltage of the point A begins to increase and when base of transistor T_1 goes high, the charge in capacitor flows directly to ground with in very short time because of the small resistance of the transistor. It is clear that obtaining a linear wave at maximum PWM voltage amplitude is quite important. Regarding this fact, linearity of the curve is tried to achieve by varying time constant choice adjusted from P_1 which also effects amplitude of the output curve. P_1 adjusts linear part of 12 V exponential by determining cut off time. After proper settings a quasi linear wave form and 5.124 Volt amplitude is obtained. Frequency of the wave is the same as the timer frequency which is a necessity for stable operation of the entire system.

The last part of the PWM circuit is a comparator with ½ of LM358 (Low power quadratic opamp), and a buffer with other ½ of LM 358 and a relay. Signal at, point B, is applied to inverting input of OP1 and analogue output of PCL 812 Data Acquisition Card which occurs as a result of controller calculations in PC is applied to non inverting input. Since LM 358 output is 0 V when inverting input is bigger than non-inverting input and V_{DC} when non-inverting input is bigger. As we compare the sawtooth wave with our analogue input, output is a signal that has a duty ratio (eg: %20 on %80 off) which is appreciate to AO voltage value.

This signal is connected to a buffer circuit, a unity gain amplifier (OP2), to protect the circuit voltage changes in L298 motor driver circuit. Relay is used for changing modulated signal pin from inp1 to inp2 which are logical control circuit inputs for motor driving circuits.

3.6 Principles Of Motor Control

Figure3.9 illustrates driving a DC motor using a power transistor bridge. By driving the four transistor in the correct sequence the direction of current flow through the motor is reversed, consequently reversing the direction of the motor's rotation. The motor torque is a function of the current amplitude, the motor's internal parameters, and the external load. The resistive torque is dependent on the motor's internal friction. The current level can be controlled with current chopping. The controller checks the current level by monitoring the sense resistor voltage and then drives the appropriate power transistor. On the other hand this means that when current does not flow in the sense resistor it is not possible to measure the current level and thus limit it.

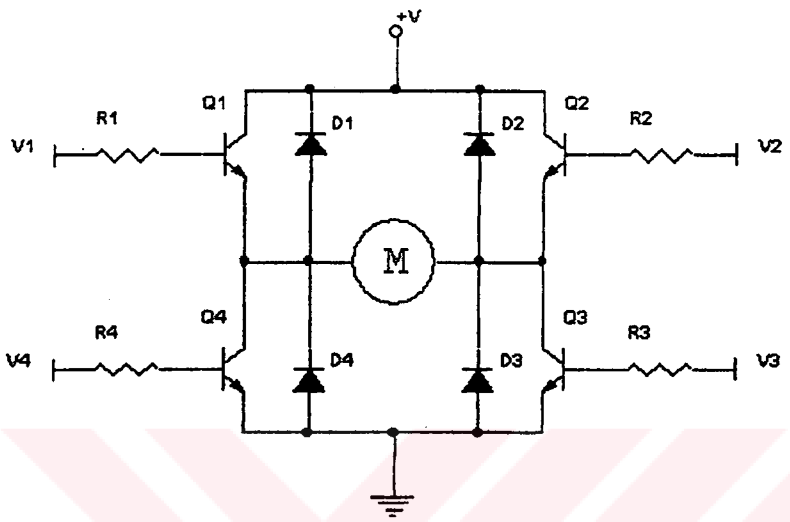


Figure 3.9 H Bridge circuit

A more general application circuit includes an external control loop. Data relating to the actual motor speed is transmitted to the controller by the system which stabilizes the current in the bridge as a function of the requested rotation velocity. In this case also the current is limited through chopping.

Electrically a DC motor can be viewed as a series RL network with a voltage generator V(w). The generator represents the back electromotive force (BEMF) generated by the motor's rotation and which opposes the electromotive force of the supply. The value of the BEMF is a function of the motor's angular velocity. If the motor has no external load and its velocity is not limited, it will accelerate up to the velocity w such that V(w) equals the supply voltage Vs. In this situation the two EMF's cancel each other and thus the motor torque responsible for acceleration will go away. In reality V(w) is always slightly less than Vs in which case a small motor torque is necessary to compensate resistive torque due to internal friction. Thus it can be seen that the motor's BEMF can reach elevated values which in some cases can create application problems due to a certain type of stress. Solution to this problem is the is to put what is known as a flyback diode in the reverse direction across the inductive load, so that the voltage spike will forward bias the diode, creating a return path for the current. Fig(c) shows how a flyback diode is connected.

3.7 Motor Driver Circuit

The L298 is an integrated monolithic circuit It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the connection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

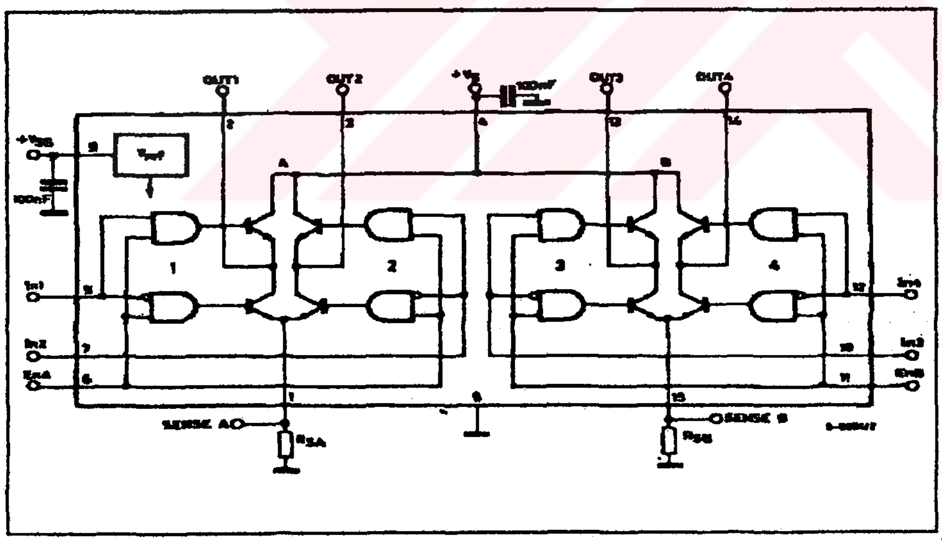


Figure 3.10 Schematic Circuit of L298

The L298 integrates two power output stages (A; B).The power output stage is a bridge configuration and its outputs can drive an inductive load in common or differential mode, depending on the state of the inputs. The current that flows through the load comes out from the bridge at the sense output : an external resistor allows to detect the intensity of this current.

When the supply and the logic supply voltage applied and the motor is connected to output pins of any of two bridges. Circuit is ready to operate. Enable pins controls whether the bridge is on work . In order to explain operating of the circuits, it is assumed that input1 and enable A is high and input2 is low. This logic signals are connected to input pins of four And-logic gates directly or inverted. In given conditions, Transistor A1 and A4 is triggered and current flows through output1 to output2 and motor turns counter clockwise. If input2 is high and input1 is low, transistors A2 and A3 are conductive this causes current flows output2 to output1 and motor turns clockwise. In each case current flows on external resistor R_s, connected to current sense output, through ground. The sense output voltage can be used to control the current amplitude by chopping the inputs, or to provide over

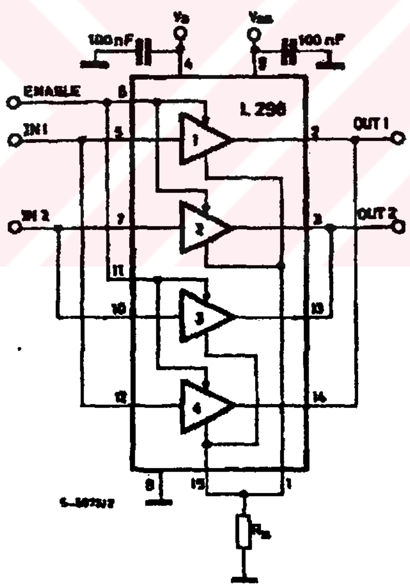


Figure 3.11 Power setup of L298

current protection by switching low the enable input. As the repetitive peak current needed from the load is higher than 2 Amps, a paralleled configuration is chosen.(figure 3.11) This solution can drive motors up to 3 amperes in DC operation.

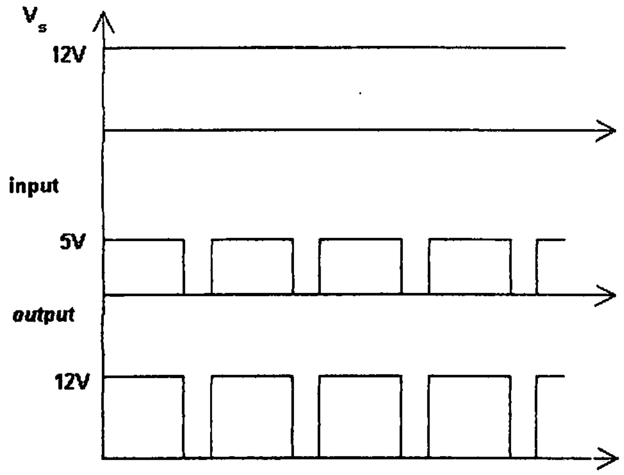


Figure 3.12 PWM Signal at the output of L298

When L298 and PWM circuit combined modulated signal goes to one of the input pins and the other is connected to ground through relay. At this situation, supply voltage modulated by PWM signal and resulting signal flows on to motor. and determines the angular velocity of motor (figure 3.12).

CHAPTER FOUR

SAMPLED DATA SYSTEMS

4.1 Introduction

In the last years Industrial PC I/O interface products have become increasingly reliable, accurate and affordable. Because of this, PC-based data acquisition and control systems are widely used in industrial and laboratory applications like monitoring, control, data acquisition and automated testing.

Selecting and building a Data Acquisition and Control system that actually does what you want it to do covers some knowledge of mechatronics and computer engineering. (Transducers and actuators, signal conditioning , control hardware and software)

The computer as controller unit contains data acquisition (DAQ) cards by which the position signal is transmitted to computer where a user-specified control algorithm can be implemented. The control signal is redirected, through the DAQ card, to the electronics box, which produce PWM signal and powers the actuator (drive motor). The computer is capable of executing control laws at a sampling rate and this causes the implementation to be modeled as discrete time. Two digital-to-analog converters (DAC's) provide for real-time analog signal measurement and controller output.

Analog signals are converted by sampler to discrete variables that change only at specified intervals are subject to discrete control. A discrete control system can manipulate a discrete variable only when the schedule calls for the next operation.

Continuous and discrete control systems behave very differently and are generally designed according to different mathematical principles. However, the two come together in sampled control applications where the process variables change continuously, but can only be measured at discrete intervals.

All computer-based controllers perform sampled control. Whether it is part of a distributed control system, a single-loop controller, or a PC-based controller, a control computer must wait to measure the process variables until its program calls for the next round of sensor readings.

4.2 Sampler

The sampler is essentially a switch, operating usually at fixed intervals of time. When the 'switch' closes, it grabs or samples the output of the transmitting device. It then transfers the sampled signal to a receiver. The sampler can operate on both continuous or discrete signals.

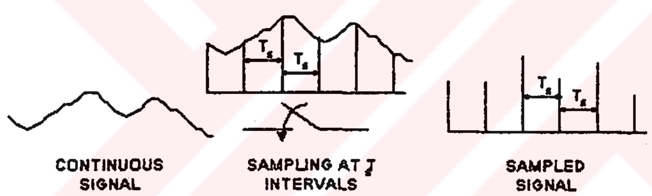


Figure 4. 1 Implementation of a Sampler

Thus if the source signal is continuous, the output of the sampler is a series of pulses, and the magnitude of each pulse is equal to the magnitude of the continuous signal at the instant of sampling as shown in the figure 4.1.

Once the measurements have been read into memory, the computer must take time to analyze the data and compute the appropriate control actions. Only then can the computer return to reading the sensors. It may take milliseconds between readings, but during that sampling interval, the computer skips what's going on in the process. Because of this, the design of any computer-based control system must address the scarcity of data that results from sampling. The simplest design strategy is to run the computer so fast that the sampling interval approaches zero and the sampled signal appears continuous. This allows the computer to use control algorithms based on the more traditional principles of continuous control.

Shortening the sampling interval also seems like a good way to prevent sudden changes in the process variables from going entirely unnoticed between samples. However, excessively fast sampling rates waste computing resources that might be better used for other purposes, such as interfacing with the operator or logging historical data.

The rate at which you need to sample an analogue signal, to get a good digital representation, depends on how fast the signal is changing. For faster signals this will be determined from the maximum frequency component in the signal, but for slowly changing signals from industrial plants you can use the maximum expected rate of change To get an accurate picture of the waveform you will need a sampling rate of 10 to 20 times the highest frequency.

Most industrial processes involve some combination of friction, inertia, and system stiffness that prevents a process variable from changing rapidly. For processes like these, high-frequency fluctuations in the process variable simply aren't possible. Fast sampling is not required to capture the entire signal. In such cases, The required sampling rate can be determined experimentally.

For modeling sampled signal some mathematical theory are developed parallel to laplace transform of the linear time invariant systems. The Z-transform which will be defined can be utilized in the analysis of discrete-time systems modeled by difference equations;

$$x(k) = b_n e(k) + b_{n-1} e(k-1) + \dots + b_0 e(k-n) - a_{n-1} x(k-1) - \dots - a_0 x(k-n) \qquad (4.1)$$

This is the general form of an n.th order linear difference equation. A transform is defined for number sequences as follows. The function "E(z)" is defined as a power series in z^{-k} with coefficients equal to the values of the number sequence . This transform, called the z-transform, is expressed by the transform pair:

$$E(z) = Z\{e(k)\} = e(o) + e(1).z^{-1} + e(2).z^{-2} + ... = \sum_{k=0}^{\infty} e(k).z^{-k} \tag{4.2}$$

$$e(k) = Z^{-1}\{E(z)\} = \frac{1}{2\pi.j}\oint E(z).z^{k-1}dz \tag{4.3}$$

The z-transform is defined for any number sequence and may be used in the analysis of any type of system described by linear time-invariant difference equations. As it will be introduced former, when dealing with digital or sampled control systems, one has to deal with a system which has some of its components working in the discrete-time domain, some others in the analog one.

At this point of the presentation it is of some interest to notice that the designing of a digital control system can be viewed as the design of an Infinite Impulse Response filter (which thus has some poles). Dealing with the design of a filter in which we will have to deal with some information in the discrete-time domain and some other in the analog domain: the choice here is to use the Bilinear transform in order to meet requirements.

Many analysis and design techniques for continuous-time linear time-invariant systems are based on the property that in the s-plane the stability boundary is the imaginary axis. Thus these techniques cannot be applied to linear time-invariant discrete-time systems in the z-plane, since the stability boundary is the unit circle. However, through the use of the transformation The unit circle of the z-plane transforms into the imaginary axis of the s-plane.

$$z = \frac{1 + s/2}{1 - s/2} \tag{4.4}$$

$$s = 2\frac{1 - z^{-1}}{1 + z^{-1}} \tag{4.5}$$

4.3 Analog to Digital And Digital to Analog Converters

Analog to Digital Converters (ADCs) converts sampled voltage or current signals their binary equivalent while Digital to Analog Converters (DACs) converts binary

CONVERSION CODE

RANGE OF ANALOG INPUT VALUES	DIGITAL OUTPUT CODE
4.5 _ 5.5	0 ... 101
3.5 _ 4.5	0 ... 100
2.5 _ 3.5	0 ... 011
1.5 _ 2.5	0 ... 010
0.5 _ 1.5	0 ... 001
0 _ 0.5	0 ... 000

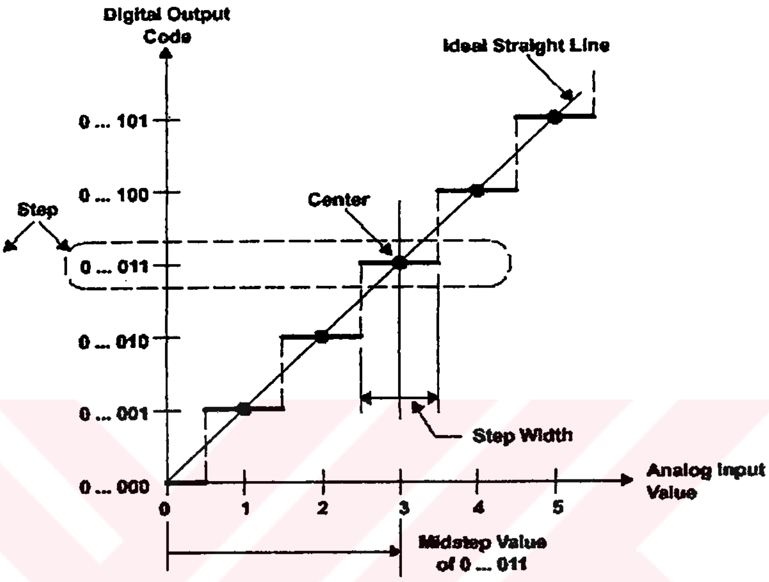


Figure 4. 2 Graphical representation of ADC

signals to continuous signals such as voltages or currents. These converters provide the interface between a computer and the external environment.

An ideal ADC uniquely represents all analog inputs within a certain range by a limited number of digital output codes. The diagram in Figure 1 shows that each digital code represents a fraction of the total analog input range. Since the analog scale is continuous, while the digital codes are discrete, there is a quantization process that introduces an error. As the number of discrete codes (or another words) increases, the corresponding step width gets smaller and the transfer function approaches an ideal straight line. The steps are designed to have transitions such that the midpoint of each step corresponds to the point on this ideal line.

The resolution of an ADC is usually expressed as the number of bits in its digital output code. For example, an ADC with an n-bit resolution has 2^n possible digital codes which define 2^n step levels. However, since the first (zero) step and the last step are only one half of a full width, the full-scale range (FSR) is divided into $2^n - 1$ step widths.

A DAC represents a limited number of discrete digital input codes by a corresponding number of discrete analog output values. Therefore, the transfer function of a DAC is a series of discrete points as shown in Figure 2. For a DAC, 1 LSB corresponds to the height of a step between successive analog outputs, with the value defined in the same way as for the ADC. A DAC can be thought of as a digitally controlled potentiometer whose output is a fraction of the full scale analog voltage determined by the digital input code.

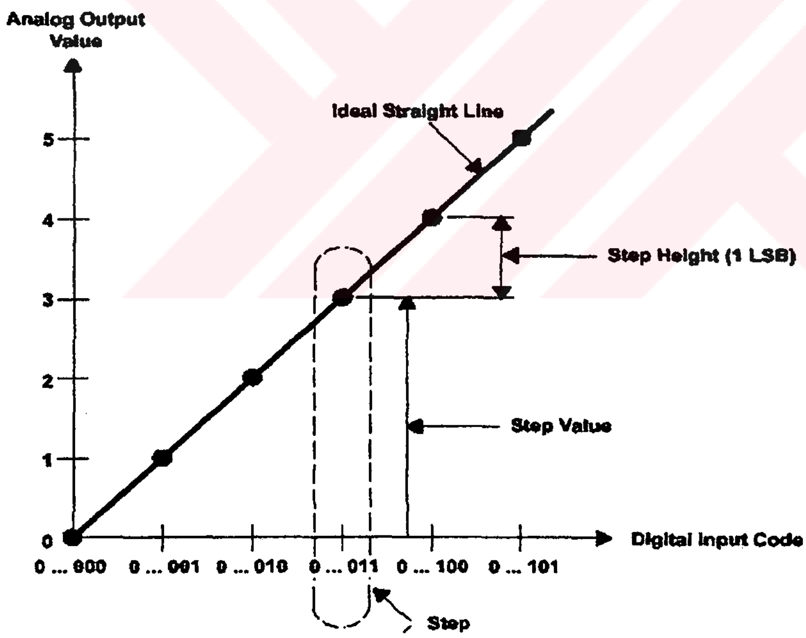


Figure 4. 3 Graphical Representation of DAC

4.4 Signal Hold Devices

The output of a sampler is a train of pulses, regardless of whether the source is continuous or discrete. Thus the output of a computer after digital-to-analog conversion is also a train of pulses. If this is a control signal, then unless the device receiving this signal, say a pump or valve, has integration capabilities, then the process will be driven by pulses. This is obviously not acceptable. So, in process control applications, the signal from the DAC is always 'held' using hardware known as signal hold devices. The most common is the Zero-Order-Hold, where each pulse is held until the next pulse comes along.(figure 4.4).

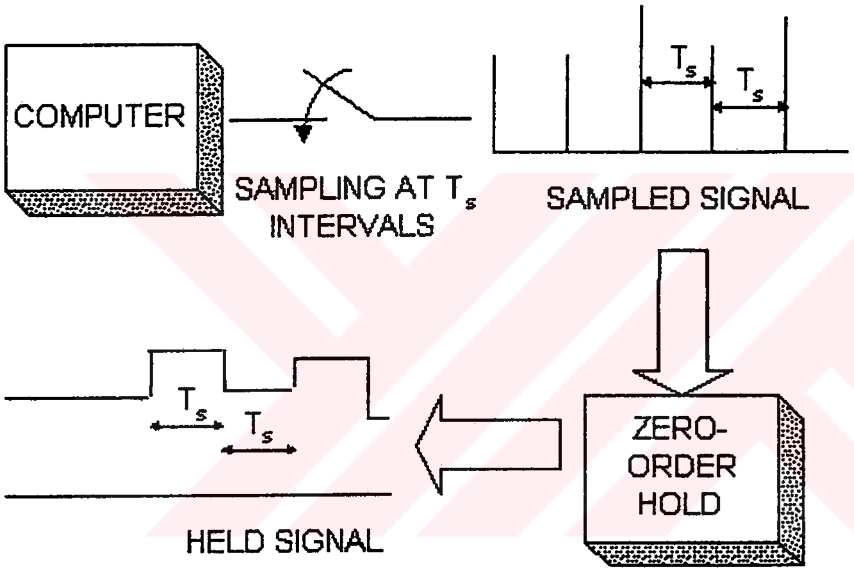


Figure 4. 4 Output Signal of ZOH

When defining mathematical model for computer simulation, system is considered as continuos system because the sampling time is 0.0036 s and the period of the response curve is 2.32 s. The ratio is about 750 so we can assume that system is supposed to be linear. But is must not be neglected that to obtain a computer algorithm for PID controller by bilinear transform.

CHAPTER FIVE

MATHEMATICAL MODEL OF CONTROL SYSTEM

5.1 Introduction

When constructing a mechanical model, it is hard to calculate the system characteristics when they are related in to assembly of the system. Though dimensions of the system are strictly given in technical drawings, it is only possible to manufacture parts with in a tolerance limit. And actuator, which generally bought appropriate to system characteristics, is only limited technical information obtained. In these conditions, methods for calculating system performance can be used to determine the characteristics of the unknown system

Since the acquisition system helps to get dynamic data from mechanical model, it is possible to calculate parameters of the system via time domain analysis methods.

It is usually of interest to evaluate the state and output responses with respect to time which is used as an independent variable in most control systems. In analysis, a reference input is applied to a system, and the performance of the system is evaluated by studying system response in time domain. The time response of the system is generally divided in to two parts: the transient response and the steady state response. Steady state can be defined as the fixed response when time reaches infinity and the transient response is defined as the part of the system that goes to zero as the time becomes very large.

5.2 Components of Model

Controller consists of two separate parts: Data Acquisition Card for Analog input and output and C++ program with different control algorithms. It will be denoted as G_c. As mentioned in previous chapter, it will be considered as analog controller because of the fast sampling rate. For basic calculations; only proportional controller is implemented. This causes that control output will be constant times more than error. When we define this mathematically G_c will be;

$$G_c = K_p \tag{5.1}$$

Controller does not generate output with a range. Theoretically this is not supposed to be a problem but digital to analog converter has 0 to +5V range which resulted two application problem: The first problem is normal controller has both outputs positive and negative therefore a one bit digital output is added, which controls relay to change the modulated signal connection to L298 input 1 or input 2 and absolute value of the controller output is sent as analog output. This problem cause no mathematical difference in equations. Other problem is the limited scale which can let only 5V maximum. This nonlinearity is described as saturation and it places at the output of the controller.

Electronic card designed for pulse width modulation and motor driver can be thought as an amplifier with gain K_1 since it changes analog output voltage to motor voltage.

$$\begin{aligned} U_{PWM}(t) &= K.U_m(t) \\ G_D(s) &= K \end{aligned} \qquad K = \frac{12}{5} = 2.4 \tag{5.2}$$

Differential equations of Permanent Magnet DC motor is as following ;

$$L_a \frac{di_a}{dt} + R_a i_a(t) = e_a(t) - K_b w_m(t) \tag{5.3}$$

$$J\frac{dw_m}{dt} + Bw_m(t) = K_i i_a(t) \tag{5.4}$$

$$G_{me}(s) = \frac{1}{L_a s + R_a} \tag{5.5}$$

$$G_{mm}(s) = \frac{K_i}{Js + B} \tag{5.6}$$

But because of the nonlinearity of the electronic components and insufficient technical data of motor it was not possible to evaluate or determine the coefficients mentioned above. Instead of this mathematical modeling; an experimental procedure is followed.

Characteristics of the PWM and motor driving circuits and motor is determined by means of input *u* (pulse width modulation voltage) output *v* (linear velocity of the belt).

For this experiment; PC analog output module of data acquisition card, PWM and motor driving circuit and motor is used. In PC, a computer program which gives a constant output for a time which is also measured by program itself runs and this voltage is transmitted to electronic cards. By the effect of voltage, motor turns and belt moves. When program is interrupted by keystroke, elapsed time (Δt) occurs on display. Afterwards position change (Δx) is measured by a ruler. Counter clockwise rotation is assumed to be positive. Data, obtained from experiments, are given in following tables.

Table 5.1: Data from modeling experiment of actuator block (cw)

U (volt)	Δt (second)	Δx (millimeter)	V (mm/s)
-5.0	1.86	162	87.10
-4.5	2.03	156	76.85
-4.0	2.20	145	65.91
-3.5	3.52	192	54.55

-3.0	4.28	189	44.16
-2.5	5.11	175	34.25
-2.0	4.26	111	24.34
-1.5	10.27	145	14.12
-1.4	2.00	0	0

By the help of these data; a polynomial is fitted according to least squares method:

$$V = 22.5717 \times U + 23.3469 \quad (5.7)$$

Table 5.2: Data from modeling experiment of actuator block (ccw)

U (volt)	Δt (second)	Δx (millimeter)	V (mm/s)
1.5	2.00	0	0
1.9	8.57	152	17.74
2.0	5.05	100	19.80
2.5	4.12	130	31.55
3.0	4.78	205	42.89
3.5	3.08	168	54.55
4.0	3.46	224	64.74
4.5	1.59	119	74.84
5.0	2.14	175	81.78

$$V = 22.8265 \times U - 27.3299 \quad (5.8)$$

From the values it is obvious that motor output has a dead zone around "0V" level. The major reasons for this nonlinearity are inertia of gearbox coupled and friction between mechanical components.

This kind of nonlinearity can be defined as dead zone. As stated above equations for negative and positive values of voltages are similar but not to same. To obtain a model for the motor; another approximate equation for system can be calculated by average of the coefficients of two formulas (5.7) and (5.8).

$$v = 22.5217 \times u \mp 25.3387 \qquad (5.9)$$

Block diagram of the motor and electronic circuits according to equation (5.9) and output of the model is given in figure (5.1).

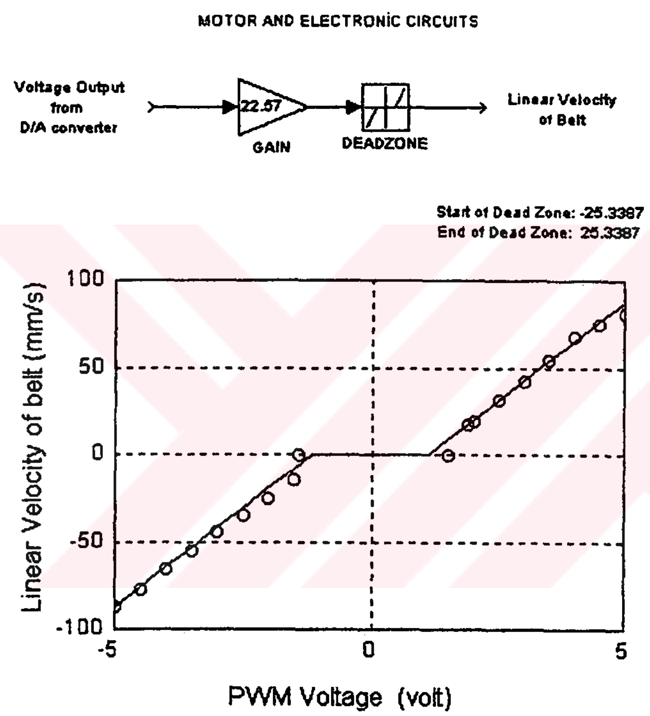


Figure 5.1 Block Diagram of The Motor and Simulation Output

In further developments of mathematical model and for simulation this block diagram is used instead of PWM circuit, motor driving circuit and motor.

For the main part of the mechanical model, platform, there are not any technical specifications from which a mathematical model can be composed. To obtain transfer function for this mechanism another experimental study has been made. By observing recorded free motion of the platform, (figure 5.4) its differential equation is supposed to be second order, though there are no actual damper or spring. But friction on bearings the feedback potentiometer obviously cause damping effect. According to this assumptions, form of the differential equation:

$$\sum T = I \times \frac{d^2\theta(t)}{dt^2} + C \times \frac{d\theta(t)}{dt} + K\theta(t) \tag{5.10}$$

Since we have in sufficient data to calculate coefficients of the differential equation, time domain analysis of the system will useful to determine the transfer function of the system.

Time response of a system has some important technical criteria. These are performance specifications such as logarithmic decrement, steady state time and peak value and peak time. When an impulse input applied to underdamped ($0<\xi<1$) second order system, which model is assumed to be approximate, solution of the system is:

$$X_a(t) = \frac{K \times W_n}{\sqrt{1-\xi^2}} \times e^{-\xi \times W n \times t} \times \sin(W_n \times \sqrt{1-\xi^2} \times t + \Phi) \tag{5.11}$$

$$\Phi = \arctan(\xi) \tag{5.12}$$

As plotted in figure (5.2), and stated above equation, the theoretical response is damped sinusoid. Resulting function is a combination of an exponential envelope and a sinus wave. Constants w_n, ξ are physically significant. Since they are directly effecting response characteristics, it is possible to evaluate them by means of performance specifications.

One of them can be logarithmic decrement; if the amplitudes on any two successive peaks are measured, the ratio of these amplitudes is constant.

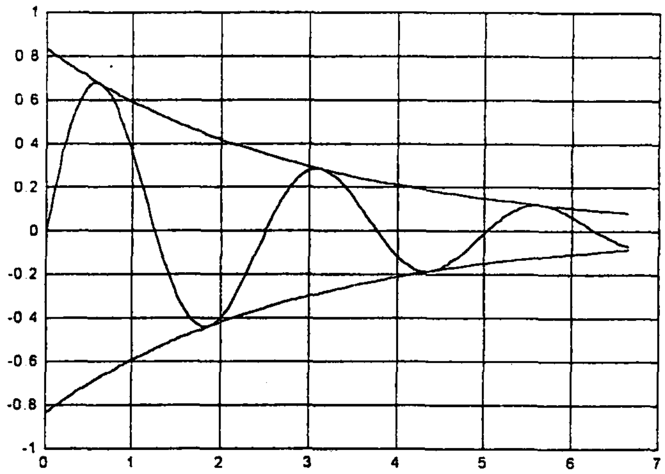


Figure 5.2 Theoretical Impulse Response of A Second Order System

The maxima and minima of the system occur at periodic intervals and have the values:

$$C(t)\Big|_{\max\, or \min} = (-1)^{n-1}.e^{-\frac{n\pi\zeta}{\sqrt{1-\zeta^2}}} \tag{5.13}$$

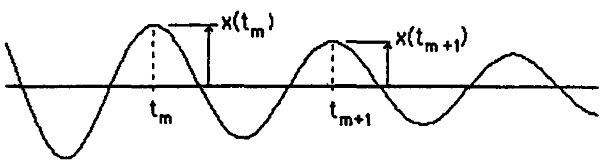


Figure 5.3 Foundation of Logarithmic Decrement

The two successive peaks used to find the log decrement. The log decrement will be found to be the same for any two peaks.

$$\mathrm{MP1} = e^{-\frac{\xi\pi}{\beta}} \tag{5.14}$$

$$\mathrm{MP3} = e^{-3.\frac{\xi\pi}{\beta}} \tag{5.15}$$

The ratio of two successive peak values is the decrement on the response curve. Because the values are exponential in envelope equation (5.13), logarithmic value of the ratio is more reasonable as performance specification.

$$\mathrm{A} = \frac{\mathrm{MP1}}{\mathrm{MP3}} = \frac{e^{-3.\frac{\xi\pi}{\beta}}}{e^{-\frac{\xi\pi}{\beta}}} \tag{5.16}$$

$$\mathrm{a} = \ln(A) = 2.\frac{\xi \times \pi}{\sqrt{1-\xi^2}} \tag{5.17}$$

Referring to damped free vibration theory logarithmic decrement can be used to find damping ratio.

$$\xi = \sqrt{\frac{a^2}{a^2 + 4\pi^2}} \tag{5.18}$$

The other unknown system parameter is natural frequency and constant coefficient. Natural frequency can be calculated from any of the other performance specifications.

Time, passed from the excitation till when the amplitude of the time response reaches value of certain percentage of input value (2 % or 5 %).is determined as steady state time for impulse response .

$$X_a(t) = \frac{W_n}{\sqrt{1-\xi^2}} \times e^{-\xi \times W n \times t} \times \sin(W_n \times \sqrt{1-\xi^2} \times t + \Phi) = 0.2 \tag{5.19}$$

It is clear that solving this equation is quite hard even damping ratio is known. But with an approximation it can be assumed that the amplitude of the sinus curve (exponential envelope) is equal to desired value. Then equation will be;

$$X_a(t) = \frac{W_n}{\sqrt{1-\xi^2}} \times e^{-\xi \times W n \times t} = 0.2 \tag{5.20}$$

With proper arrangements T_{ss} can be calculated from the equation

$$T_{ss} = -\frac{1}{\xi \times W_n} \ln \frac{\sqrt{1-\xi^2} \times 0.2}{W_n} \tag{5.21}$$

When steady state time is known, W_n can be found by solving this equation with the help of advanced mathematical tools .

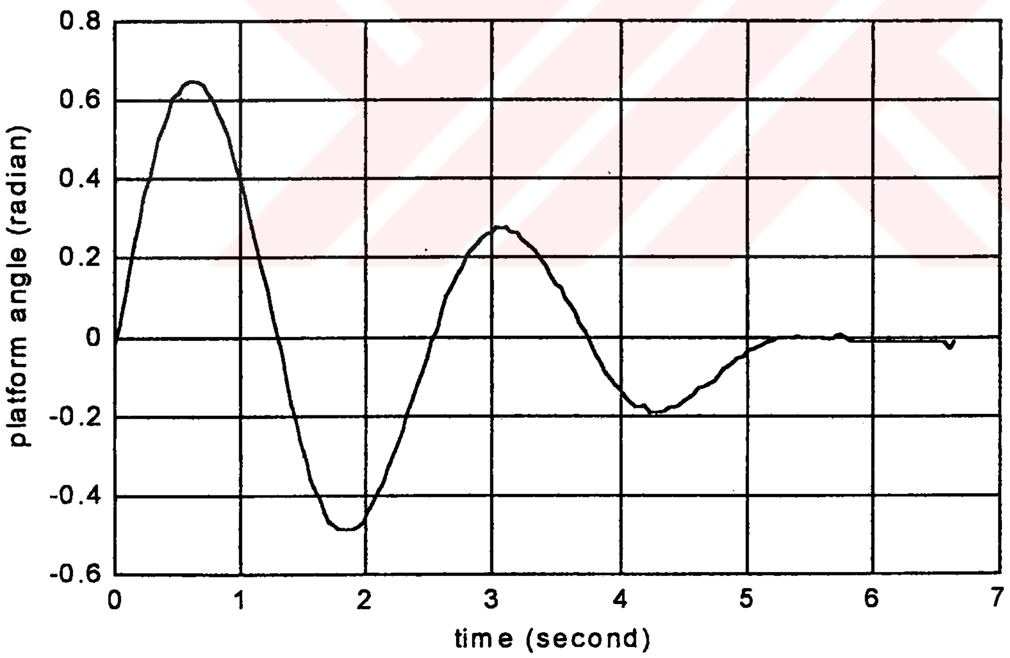


Figure 5.4 Time Response of The Real Model (platform)

The First part of the response of the real model (platform) is similar to the theoretical System response described above for determination of the parameter ξ from logarithmic decrement. Because the period of the sinusoid is approximately equal.

But as shown in figure 5.4, time response of the real model does not fit the theoretical curve. Especially last part of the response differs because of the nonlinear effects damps the small amplitude vibrations. Because of the friction and damping of potentiometer and bearings, the energy of the system is not enough to move platform away from equilibrium point. This reduces the reliability of T_{ss}.

Because of this gap, an exponential curve is fitted to absolute peak values of the sinusoid. This gives more reliable envelope function.

$$|\Theta(t)| = 0.8332 \times e^{-0.3433 \times t} = \frac{K \times W_n}{\sqrt{1-\xi^2}} \times e^{-\xi \times Wn \times t} \tag{5.22}$$

From the coefficients of the function two, unknown parameters W_n and K, can be solved.

$$W_n = \frac{0.3433}{\xi} \tag{5.23}$$

$$K = \frac{0.8332 \times \sqrt{1-\xi^2}}{W_n} \tag{5.24}$$

Three parameters are solved from equations (5.18), (5.23) and (5.24). Results are;

$\xi = 0.1359$ Ns/m

$W_n = 2.5269$ rd/s

$K = 0.3267$

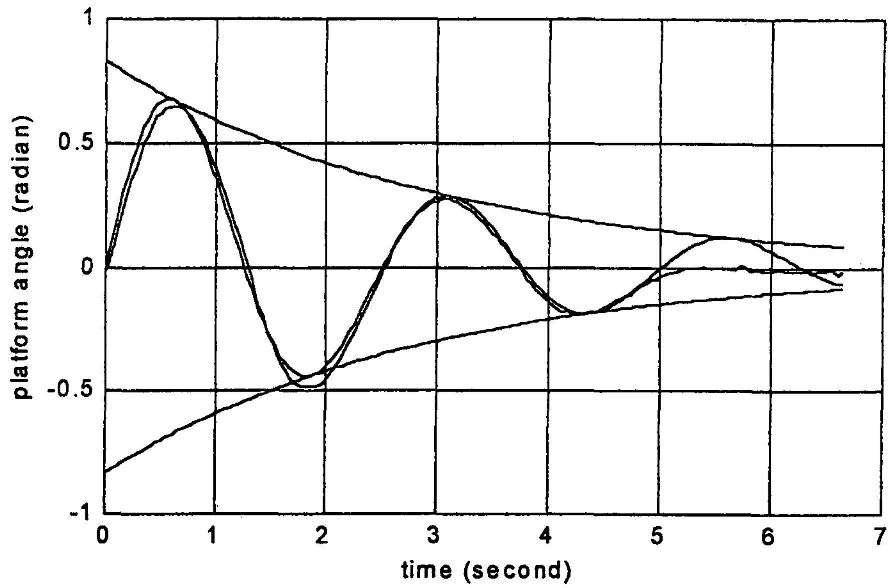


Figure 5.5 Result of Mathematical Modelling of the platform

Theoretical curves and the real curve for the platform are given in figure 5.5.As obviously seen; for the first two period fitted curve and the real one is similar but when oscillation has little amplitude around zero point friction effect increases and system does not move.

According to this modeling operation approximate transfer function of the platform is calculated as given in the formula;

$$G_p(s) = \frac{K \times W_m{}^2}{s^2 + 2 \times \xi \times W_m \times s + W_m{}^2} \tag{5.24}$$

$$G_p(s) = \frac{2.0747}{s^2 + 0.6868.s + 63852} \tag{5.25}$$

5.3 Results Of Mathematical Modeling

Transfer functions of the components and the mechanical constraints have been modeled by means of experimental and theoretical methods. According to these study model of the whole system is as following block diagram.

For comparing the mathematical model and real system excited by a step disturbing input of 0.2381 Nm.

The first implementation has not been successful. The amplitude was much smaller and the system was not fast enough to follow the real curve by changing gain of the platform, more proper curve is obtained.(figure 5.7) New gain and the transfer functions are;

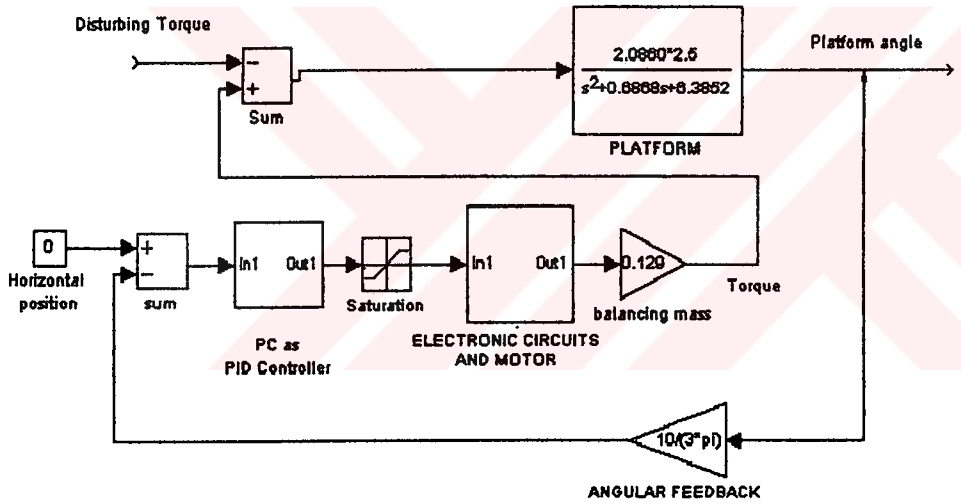


Figure 5.6 Block Diagram of the System

$$K \quad = 0.3267 \tag{5.26}$$

$$G_p(s) = \frac{5.1868}{s^2 + 0.6868.s + 63852} \tag{5.27}$$

As seen above, for the first two oscillation there is a slight lag between real curve and mathematical model. And after third period real system unable continue oscillating because of the low amplitude values. This difference is reasonable since 0.05 radian is practically "0" for the real time model.

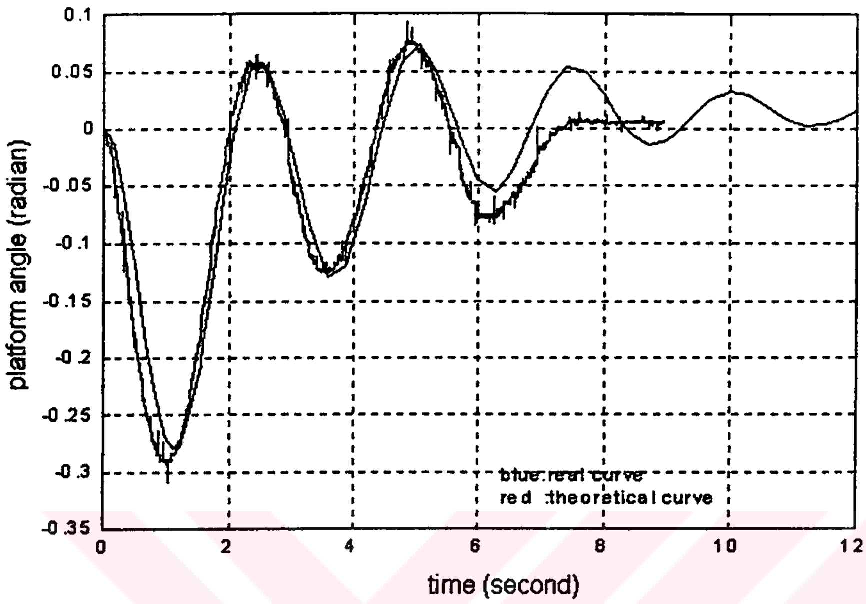


Figure 5.7 Step Responses of the Real and Mathematical Model

For further calculations; this mathematical model supposed to help to design proper controllers or initial coefficients for tuning controllers.

CHAPTER SIX

PID CONTROL THEORY

6.1 Introduction

A feedback controller is designed to generate an "output" that causes some corrective effort to be applied to a "process" so as to drive a measurable "process variable" towards a desired value known as the "setpoint." Figure 6.1 shows a typical feedback control loop, with blocks representing the dynamic elements of the system and arrows representing the flow of information, generally in the form of electrical signals.

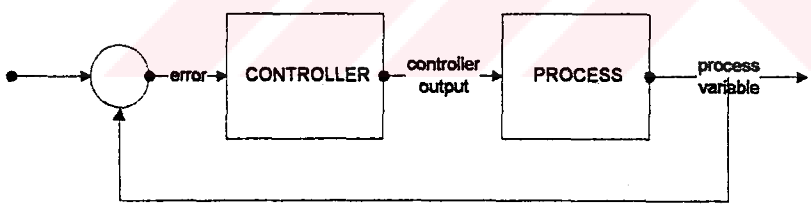


Figure 6. 1 Typical Feedback System

Virtually all feedback controllers determine their output by observing the error between the setpoint and the actual process variable measurement. A position controller, for example, uses the motion system to correct the position in a process comprised of a drive and load. It sends an electrical signal (an output) to turn on the motor when the error between the actual position (the process variable) and the desired position (the setpoint) is too high.

A proportional-integral-derivative or PID controller performs much the same function as described above but with a more elaborate algorithm for determining its output. It looks at the current value of the error, the integral of the error over a recent time interval, and the current derivative of the error signal to determine not only how much of a correction to apply, but for how long. Those three quantities are each multiplied by a tuning constant and added together to produce the current controller output u(t) as in equation (6.1). In this equation, P is the proportional tuning constant, I is the integral tuning constant, D is the derivative tuning constant, and the error e(t) is the difference between the setpoint R(t) and the process variable X(t) at time t. If the current error is large or the error has been sustained for some time or the error is changing rapidly, the controller will attempt to make a large correction by generating a large output. Conversely, if the process variable has matched the setpoint for some time, the controller will leave well enough alone

$$u(t) = K_p(e(t) + \frac{1}{T_i} \times \int e(t).dt + T_d \times \frac{de(t)}{dt}) \tag{6.1}$$

$$e(t) = R(t) - X(t) \tag{6.2}$$

Conceptually, PID controller is easy to determine mathematically. The complex part is tuning it; i.e., setting the P,I, and D tuning constants appropriately. The idea is to weight the sum of the proportional, integral, and derivative terms so as to produce a controller output that steadily drives the process variable in the direction required to eliminate the error.

The simple solution to this problem would be to generate the largest possible output by using the largest possible tuning constants. A controller thus tuned would amplify every error and initiate extremely aggressive efforts to eliminate even the slightest discrepancy between the setpoint and the process variable. However, an over aggressive controller can actually make matters worse by driving the process variable pass the setpoint as it attempts to correct a recent error. In the worst case, the process variable will end up even further away from the setpoint than before.

On the other hand, a PID controller that is tuned to be too conservative may be unable to eliminate one error before the next one appears. A well-tuned controller performs at a level somewhere between those two extremes. It works aggressively to eliminate an error quickly, but without over doing it.

How to best tune a PID controller depends upon how the process responds to the controller's corrective efforts. Processes that react instantly and predictably don't really require feedback. No subsequent corrections are required to achieve the desired illumination. On the other hand, the car's cruise controller cannot accelerate the car to the desired cruising speed so quickly. Because of friction and the car's inertia, there is always a delay between the time that the cruise controller activates the accelerator and the time that the car's speed reaches the setpoint. A PID controller must be tuned to account for such lags.

6.2 Proportional Part

With proportional control, the controller output is proportional to the error or a change in measurement (depending on the controller).

$$u(t) = K_p \times e(t) \tag{6.3}$$

With a proportional controller offset (deviation from set-point) is present. Increasing the controller gain will make the loop go unstable. Integral action was included in controllers to eliminate this offset.

6.3 Integral Part

With integral action, the controller output is proportional to the amount of time the error is present. Integral action eliminates offset.

$$u(t) = K_i \int e(t).dt \tag{6.4}$$

The effects of the integral part is that the offset (deviation from set-point) in the time response plots is extinct and the response is somewhat oscillatory and can be stabilized some by adding derivative action. Integral action gives the controller a large gain at low frequencies that results in eliminating offset. The controller phase starts out at -90 degrees and increases to near 0 degrees at the break frequency. This additional phase lag is what is built up by adding integral action. Derivative action adds phase lead and is used to compensate for the lag introduced by integral action

6.4 Derivative Part

With derivative action, the controller output is proportional to the rate of change of the measurement or error. The controller output is calculated by the rate of change of the measurement with time.

$$u(t) = K_d \times \frac{de(t)}{dt} \tag{6.5}$$

Derivative action is active when a change in error exists. Thus derivative takes action to inhibit more rapid changes of the measurement than proportional action. When set-point change occurs, the derivative action causes the controller gain to move the wrong way when the measurement gets near the set-point. Derivative is often used to avoid overshoot.

Derivative action can stabilize loops since it adds phase lead. Generally, if you use derivative action, more controller gain and reset can be used.

We are concerned with digital control, and for small sampling periods T_s, the equation may be approximated by a discrete approximation. Replacing the derivative term by a backward difference and the integral by a sum using rectangular integration, an approximation to equation (6.1) is;

$$u_n = K_p(e_n + \frac{1}{T_i}\sum_{j=1}^{n} e_j.T_s + T_d\frac{e_n - e_{n-1}}{T_s}) \tag{6.6}$$

Index n refers to time instant.

6.5 Basics of PID Control

Though PID means Proportional, Integral and Differential, it probably should have been called IPD because the Integral term is most effective at low frequencies, the Proportional term at moderate frequencies, and the Differential term at higher frequencies. These frequencies are relative to the bandwidth of the motor or process.

In order to discuss the effects of PID, it is necessary to look at a basic closed loop with gain and the equation for its closed loop response. The Bode diagram of the given system is shown below;

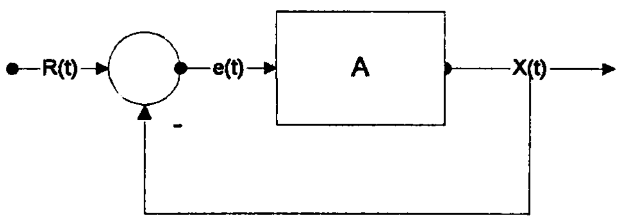


Figure 6. 2 Basic Close Loop System with Gain

$$| F(w) |= \frac{A}{1+A} \tag{6.6}$$

where A denotes 1/w.

It is evident that A/(1 +A) is approximately equal to 1 when A is large and to A when A is small - as shown on the diagram above.

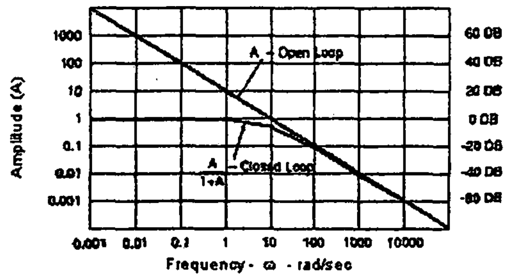


Figure 6. 3 Bode Diagram of The System

The Bode diagram shows A approaching infinity as the frequency approaches zero. Theoretically, it does go to infinity at DC because if one put a small error into an open loop drive/motor combination to cause it to move, it would continue to move continuously. This is why a motor is classified as an integrator itself - it integrates the small position error. If one closes the loop, this has the effect of driving the error to zero since any error will eventually cause motion in the proper direction to bring X(t) into coincidence with R(t). The system will only come to rest when the error is precisely zero But in actual practice the error does not go to zero. In order to cause the motor to move, the error is amplified and generates a torque in the motor. When friction is present, that torque must be large enough to overcome that friction. The motor stops acting as an integrator at the point where the error is just below the point required to induce sufficient torque to break friction.

One can introduce another Integrator (the motor being the first one) at the error point which recognizes and acts upon the least significant increment of the error. As long as an error exists, it will integrate (continually sum) that error (within certain limits) until motion takes place. A second Integrator shows up on a Bode diagram with a plot of gain that has twice the slope previously shown in figure(6.5) which shows that the low frequency dynamics are dramatically improved in addition to the error being greatly reduced.

There is a problem with A' having two integrators. As mentioned above, an integrator results in its output being 90° phase lagged from its input. By putting two integrators in series, a 180° phase lag results (expressed as A'-180°). Under this condition, the closed loop gain becomes:

$$| F(w) |= \frac{A < -180}{1 + A < -180} \tag{6.7}$$

As A approaches 1 on the Bode diagram, the denominator becomes 1 + 1 -180° = 1-1 = 0 and |F(w)| becomes infinite. This will result in severe oscillations. In order to maintain a stable system, the denominator must not be allowed to approach 0. When the term "phase margin" is used, it expresses how close the phase shift of A is to -180° when A = 1 in magnitude. A commonly accepted design goal is for A to have -135° of phase shift or less (45° of phase margin).

As the phase margin gets larger, the amount and number of overshoots diminish. As the phase margin gets smaller, the overshoots get larger and will last for longer periods until finally constant oscillation will occur.

In order to remain stable and limit the overshoot response to a step input, the second integrator must be removed in the vicinity of A = 1. This will allow the phase to approach the -90° associated with the single integration of the motor. Typically, the "break point" at which the Integrator would be removed would be about a factor of bandwidth.

The Integral and Proportional terms have provided a system which is considerably more accurate and responsive at lower frequencies, yet stable based on the phase margin criterion near servo bandwidth frequencies. It may also be desirable to further improve the phase margin so that the bandwidth can be extended or the overshoot of the step response minimized.

This is possible by introducing positive phase to improve the phase margin by means of a Differentiator. Earlier it was mentioned that an Integrator results in the output lagging the input by 90 degrees. Conversely, a Differentiator has an output that leads the input by 90°. An Integrator, with frequency, takes the form of

$$\frac{K_i}{w}\angle -90 \tag{6.8}$$

With its gain decreasing with frequency and its output having a 90° phase lag. A Differentiator takes the form

$$K_d.w\angle -90 \tag{6.9}$$

With its gain increasing with frequency, but its output having a 90° phase lead. By designing the Differentiator so that it is effective for frequencies, relatively high in the Bode diagram, the phase lead will benefit the phase margin near the bandwidth frequencies. There is an undesirable effect from this, however. Since the Differentiator causes the gain to increase with frequency, it also increases some of the machine resonances. A Bode diagram of the whole PID compensation network would be:

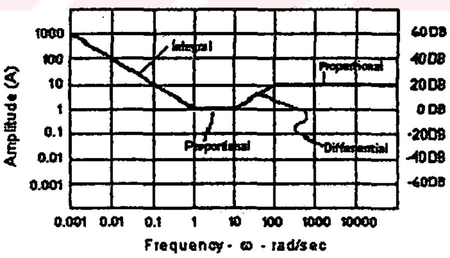


Figure 6. 4 Bode Diagram of PID Controller

Combining this with the first Bode diagram for the loop illustrates that this PID compensation has resulted in a closed loop system with a wider bandwidth and a greater gain (thus accuracy) within that bandwidth.(figure 6.5)

Clearly, the relative importance of each term in the controller's output depends on the behavior of the controlled process. Determining the best mix suitable for a particular application is the essence of controller tuning.

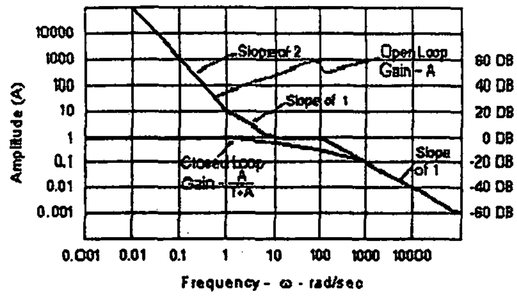


Figure 6. 5 Bode Diagram of the PID Controlled Loop

The PID network is shown below illustrating the effect of changing only the proportional factor (K_P). As can be seen, increasing K_P not only increases the midfrequency range proportional factor, but also lowers the frequency at which the integration factor ceases effectiveness and raises the frequency at which the differential factor begins to kick in. The effect of lowering K_P is also illustrated.(figure 6.6)

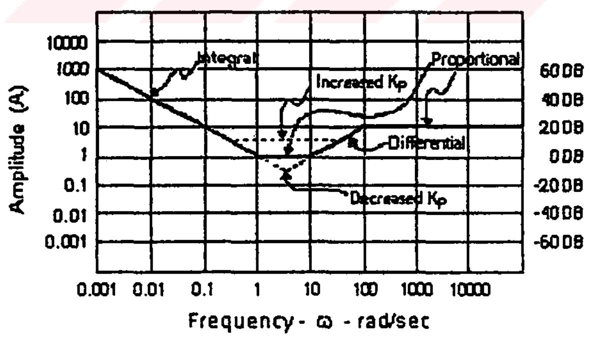


Figure 6. 6 Effect of Proportional Gain to PID Controller

Changing the integration factor (K_I) has the impact shown below. Not only does it affect the low frequency gain, but it changes the frequency at which the proportional factor (K_P) becomes effective. It would be most desirable to raise K_I as high as possible, but the higher it gets, it causes overshoots and oscillations.(figure 6.7)

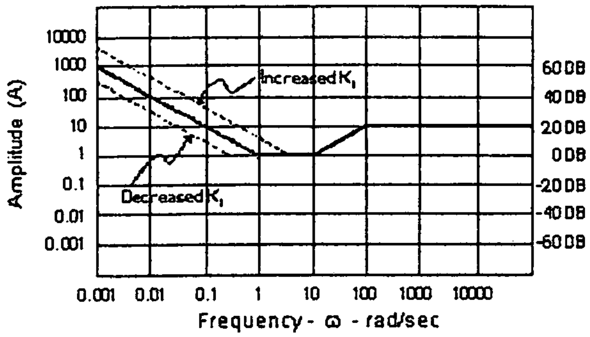


Figure 6. 7 Effect of Integral Coefficient to PID Controller

Increasing the differential gain (K_D) lowers the frequency at which it has more impact than the proportional factor (K_P). This introduces positive phase shift in the proportional range to help offset the negative phase shift from the integration factor K_I, thereby improving the phase margin and reducing the oscillating. It has the undesirable effect, however, of increasing the high frequency gain, making the system more noise sensitive and encouraging the undesirable effects of natural resonances.

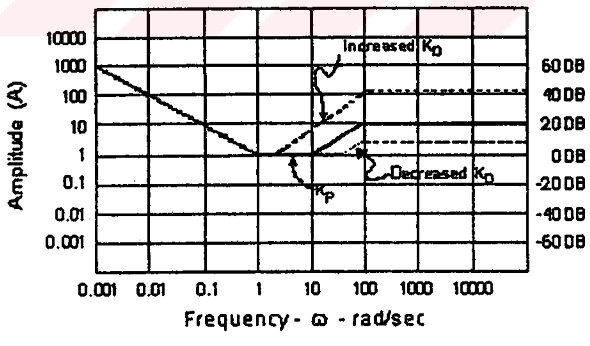


Figure 6. 8 Effect of Derrivative Coefficient to PID Controller

It is important to have a procedure to follow when working with them all together. The Bode diagram below, shows the loop gain with the PID plus the integration resulting from the motor. It also shows the effect of increasing each of the three PID gain factors.

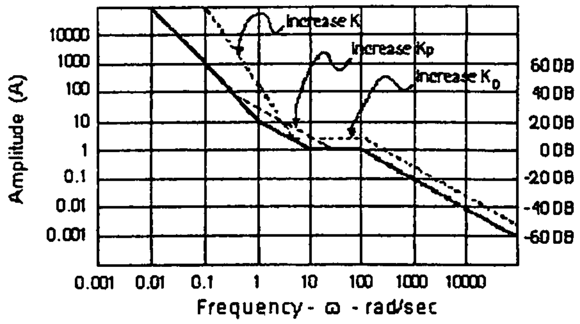


Figure 6. 9 Effect of the PID cooefficients

6.6 PID Tuning Techniques

There are basically three schools of thought on how to select P, I, and D values to achieve an acceptable level of controller performance. The first method is simple trial-and-error- -the hand tuning constants and watch the controller handle the next error-. If it can eliminate the error in a reasonable time or if it proves to be too conservative or too aggressive, increase or decrease one or more of the tuning constants. Unfortunately, intuitive tuning procedures can be difficult to develop since a change in one tuning constant tends to affect the performance of all three terms in the controller's output. Hand-tuning is based on certain rules of thumb used by experienced process engineers (Table 6.1). The tuning is a compromise between fast reaction and stability. There are exceptions to the rules in the table. If, for example, the process contains an integrator, an increase in K_p often results in more stable control.

Table 6 1 Basic rules for tuning PID controllers

ACTION	Rise Time	Overshoot	Stability
Increase K_p	Faster	Increases	Gets worse
Increase K_d	Slower	Decreases	Improves
Increase K_i	Faster	Increases	Gets worse

Procedure for hand tuning can be defined as following;

(a) Remove all integral and derivative action by setting K_i and K_d zero.

(b) Tune the proportional gain K_p to give the desired response, ignoring any final value offset from the setpoint.

(c) Increase the proportional gain further and adjust the derivative gain K_d to dampen the overshoot.

(d) Adjust the integral gain K_i to remove any final value offset.

(e) Repeat until the proportional gain K_p is as large as possible.

The advantage of hand-tuning is that a process engineer can use the procedure right away, on-line, and develop a feel for how the closed loop system behaves. A disadvantage is that it may take a long time to develop this feel, and it is difficult to sense whether the final settings are optimal.

The analytical approach to the tuning problem is more complicated. It involves a mathematical model of the process that relates the current value of the process variable to its current rate of change plus a history of the controller's output. If a differential equations can be defined for a process, its behavior can be quantified by analyzing the model's parameters. A model's parameters in turn dictate the tuning constants required to modify behavior of a process with a feedback controller.

The third approach to the tuning problem is Ziegler Nichols and something of a compromise between trial-and-error techniques and the more analytical techniques. It is more common than the others because of its simplicity and its applicability to wide range of process. Basic procedure for Ziegler Nichols tuning is;

(a) Increase the proportional gain until the system oscillates; that gain is the ultimate gain K_u.

(b) Read the time between peaks T_u at this setting.

Table 6.2 gives approximate values for the controller gains according to this experimental parameters.

Table 6 2 Controller coefficients of Ziegler Nichols

CONTROLLER	K_p	T_i	T_d
P	$0.5K_u$		
PI	$0.45K_u$	$T_u/1.2$	
PID	$0.6K_u$	$T_u/2$	$T_u/8$

The gains found by either method, must sometimes be regarded as approximate values a starting point for a hand-tuning because of the diversity of the systems. Application of PID controller to real time model and results are given in following section.

CHAPTER SEVEN

PID CONTROL OF THE MODEL

7.1 PID Control Algorithm

As mentioned in chapter 6, PID controller consists of Proportional, integral and derivative and the mathematical representation is given such as;

$$u(t) = K\left[e(t) + \frac{\int e(t)dt}{T_i} + T_d \frac{de(t)}{dt} \right] \tag{7.1}$$

$$e(t) = reference - platform\ angle \tag{7.2}$$

This equation is the super position of three independent parts. The first component is proportional equation second is integral and third is derivative. Control method is determined with the combination of the parameters (P, PI ...etc.) and the definition of the parameters determines the PID control problem. When equation (7.1) is arranged compatible with computer programming techniques in discrete domain, then it will be easy to intervene controller during experiments for small sampling periods T_s, the equation may be approximated by a discrete approximation. Replacing the derivative term by a backward difference and the integral by a sum using rectangular integration, Result of the approximation is.

$$u_n = K_p(e_n + \frac{1}{T_i}\sum_{j=1}^{n} e_j.T_s + T_d \frac{e_n - e_{n-1}}{T_s}) \tag{7.3}$$

Above equation is taken as basic for a PID Control algorithm, in order to make easy to follow the reaction of the model another mathematical arrangement has been made. According to these calculations new algorithm is:

$$u_k = K_p e_k + K_d (e_k - e_{k-1}) + K_i (\sum_{n=0}^{k} e_n) \tag{7.4}$$

where K_p, (proportional gain) K_i (integral coefficient), K_d (derivative coefficient) is defined by T_i (integral time), T_d (derivative time) and T_s (sampling time):

$$K_p = K_p \tag{7.5}$$

$$K_i = K_p \frac{T_s}{T_i} \tag{7.6}$$

$$K_i = K_p \frac{T_d}{T_s} \tag{7.7}$$

7.2 Control Program

As previously mentioned, there is a single turn potentiometer (270°) as angular displacement sensor for converting angular movement of the platform to voltage. Voltage equivalent of angular displacement is transferred to PC via analog digital converter module of data acquisition card.

In computer program, there are three modules running consequently. The first is a preprocessor. It reads the analog input port and calculates necessary data for controller module which is the second module. In the controller module, a control output is calculated according to selected algorithm (P,PI,...etc.).At the last module (postprocessor); controller output is scaled to PWM voltage range and this new value is sent to analog output module (Digital to analog converter of Data acquisition card).(figure 7.1)

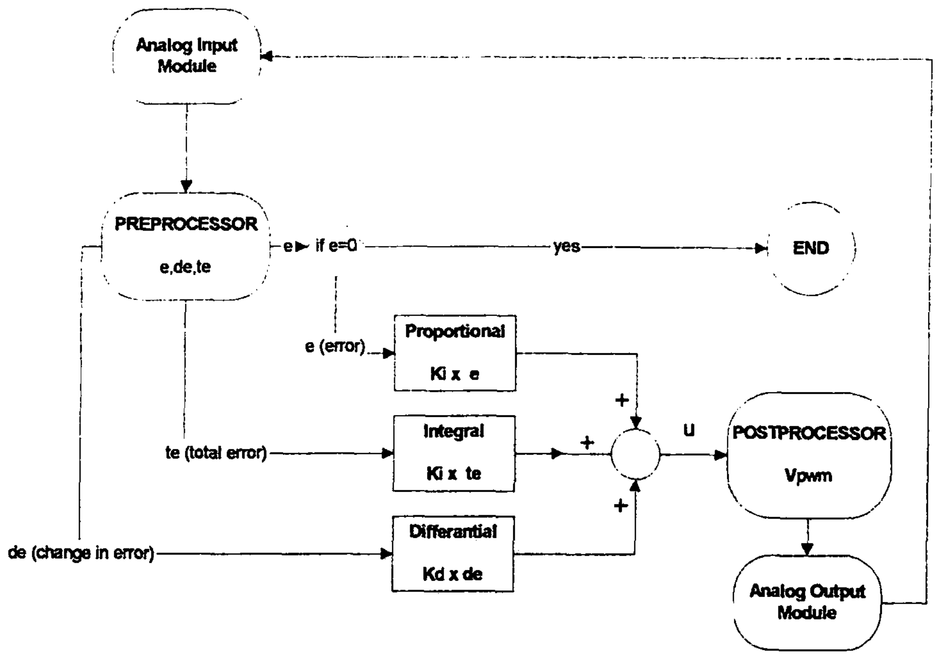


Figure 7. 1 BlockDiagram of the Computer Program

7.3 Implementation of PID Control

Referring to theory, Proportional control is the first step of the control procedure. Then it will follow as PI, PD and PID controller.

To have some initial values for controller, mathematical model has been used to find ultimate gains for Ziegler Nichols tuning procedure.When stability limit is occurred the ultimate gain and period has been found. For P controller, according to Table 6.2, control coefficient has been calculated ;

$$K_u = 39.25 \tag{7.8}$$

$$T_u = 2.4952 \tag{7.9}$$

$$K_p = 0.5 \times K_u = 19.25 \tag{7.10}$$

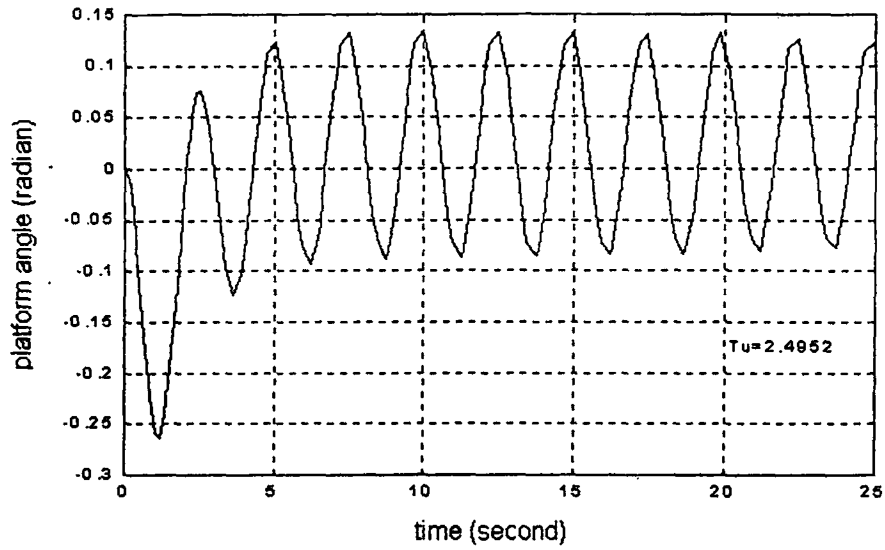


Figure 7. 2 Response of the Mathematical Model with Ultimate Gain Controller

And for PI Controller –for PI, PD and PID control sampling time is 0.055s-;

$$K_p = 0.45 \times K_u = 17.775 \tag{7.11}$$

$$K_i = K_p \frac{T_s \times 1.2}{T_u} = 0.4702 \tag{7.12}$$

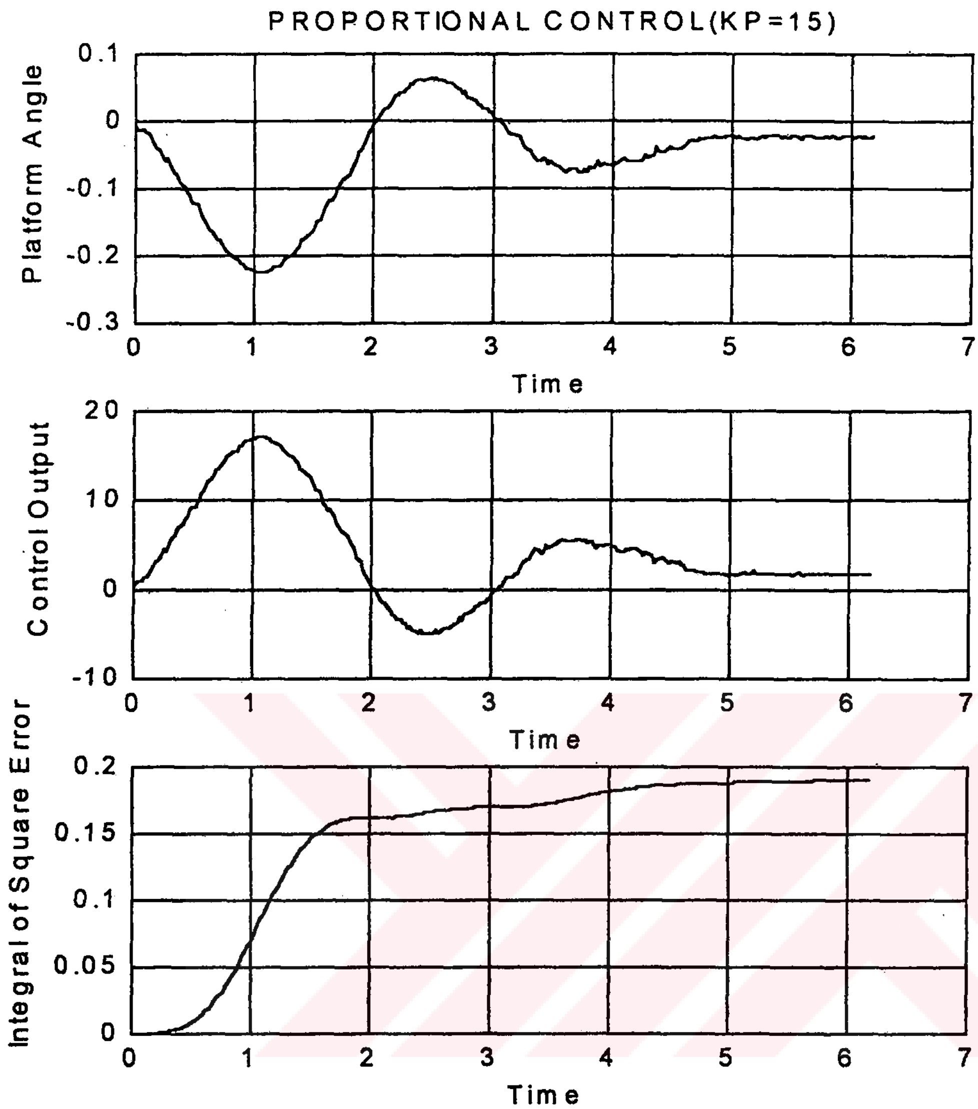
And for PID Controller;

$$K_p = 0.6 \times K_u = 23.70 \tag{7.13}$$

$$K_i = K_p \frac{T_s \times 2}{T_u} = 1.045 \tag{7.14}$$

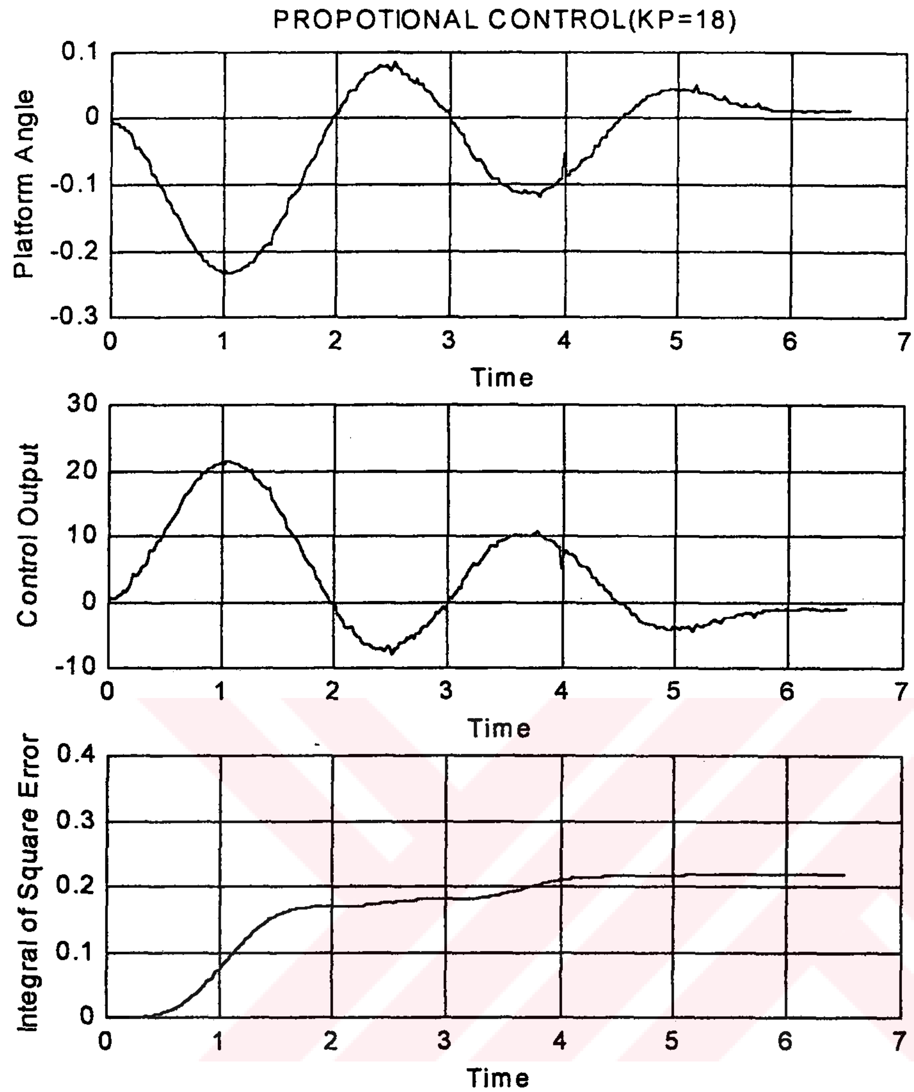
$$K_i = K_p \frac{T_u}{T_s \times 8} = 134.400 \tag{7.15}$$

Results of the implementation of P controller is given in following. The first curve is change of the platform angle, the second is the controller output scaled as motor voltage and the third is the integral of the square error as performance criteria.



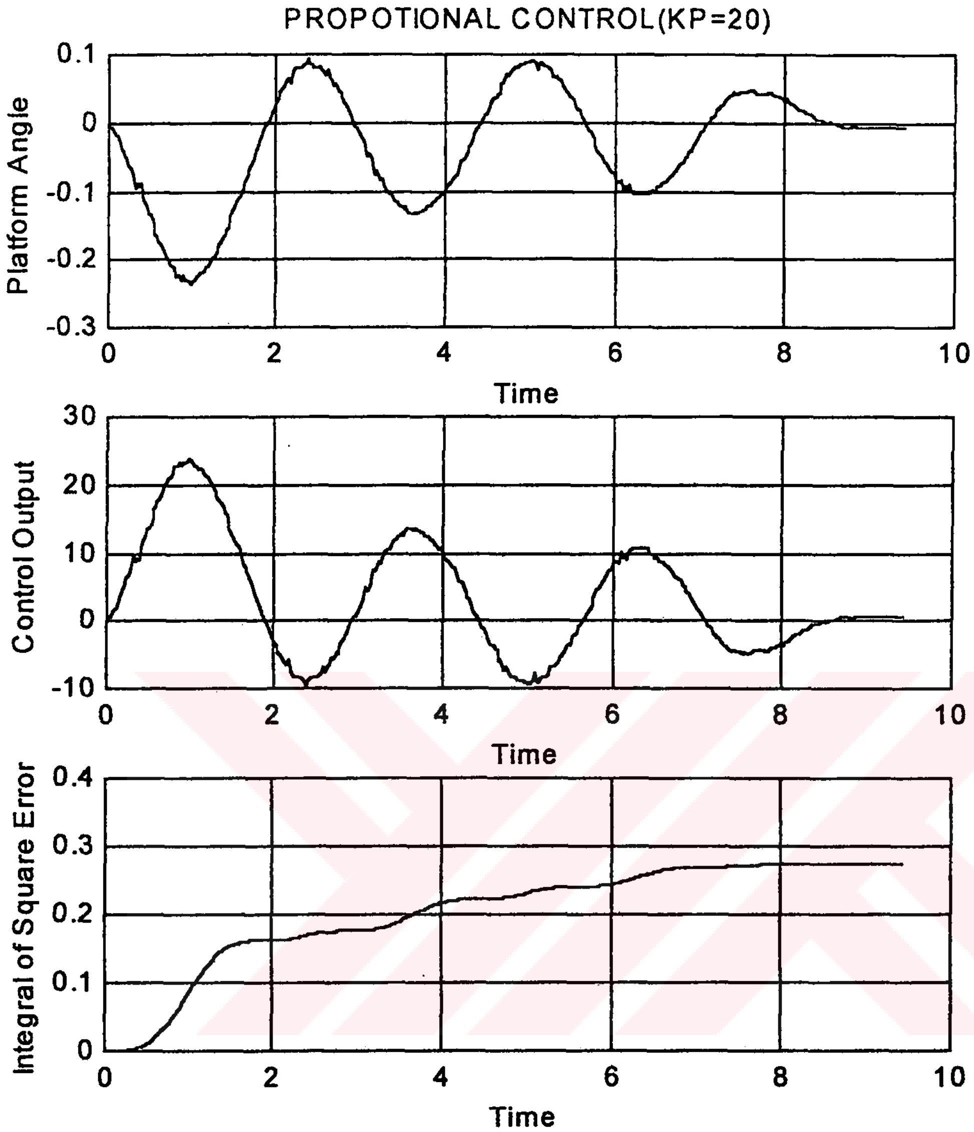
	Value	Time
Maximum	0.0652	2.5380
Minimum	-0.2345	1.1844
Steady State	0.0262	5.1220

Figure 7. 3 Step Input (0.1991Nm) Response of the System (P=15)



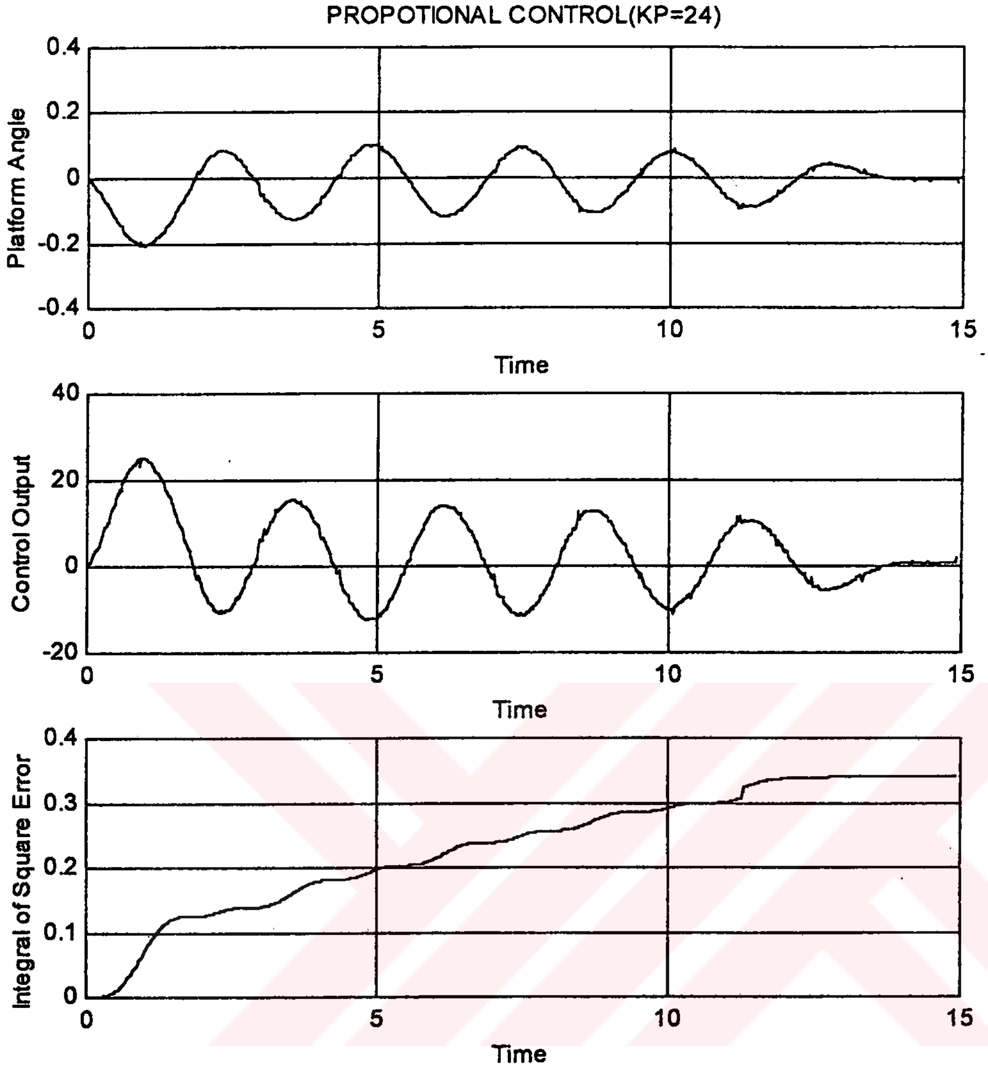
	Value	Time
Maximum	0.0877	2.5056
Minimum	-0.2356	1.0944
Steady State	0.0096	5.8710

Figure 7. 4 Step Input (0.1991Nm) Response of the System (P=18)



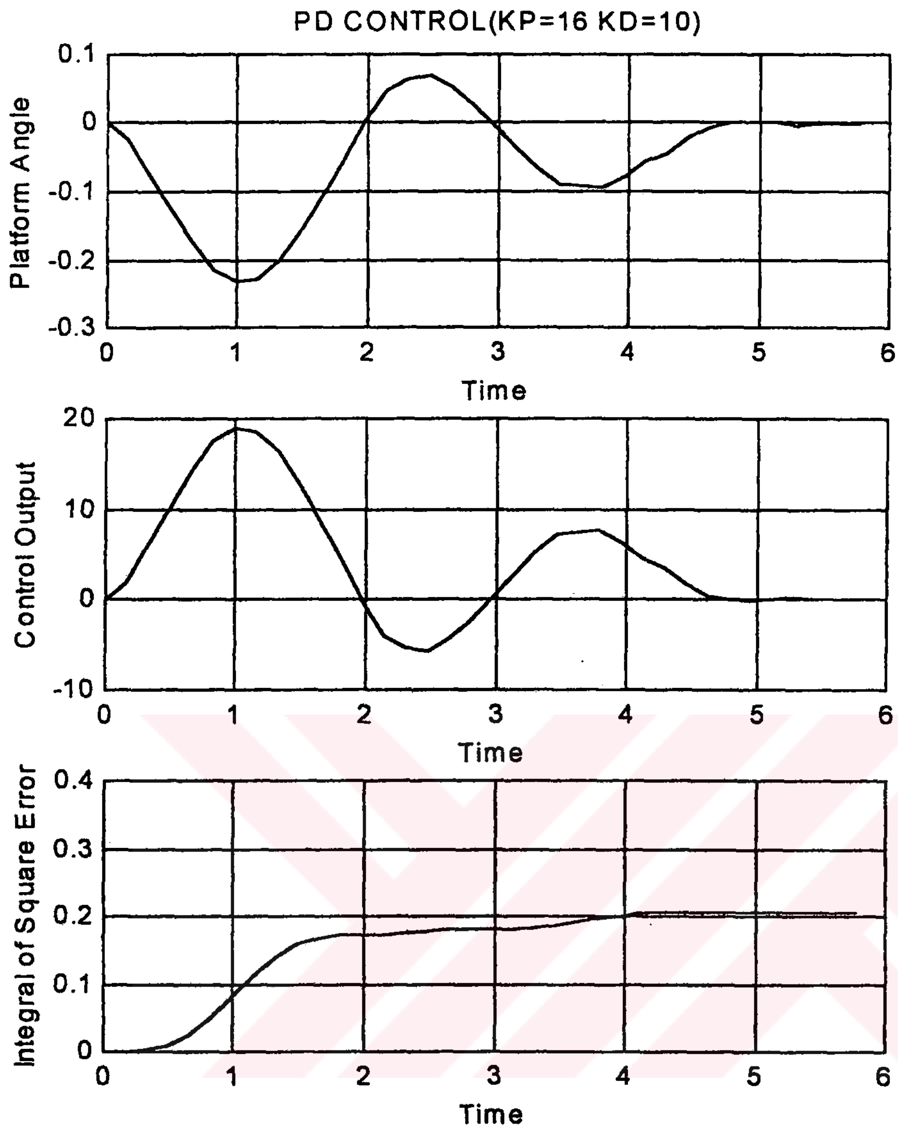
	Value	Time
Maximum	0.0969	5.0256
Minimum	-0.2356	0.9936
Steady State	-0.0068	8.6849

Figure 7. 5 Step Input (0.1991Nm) Response of the System (P=20)



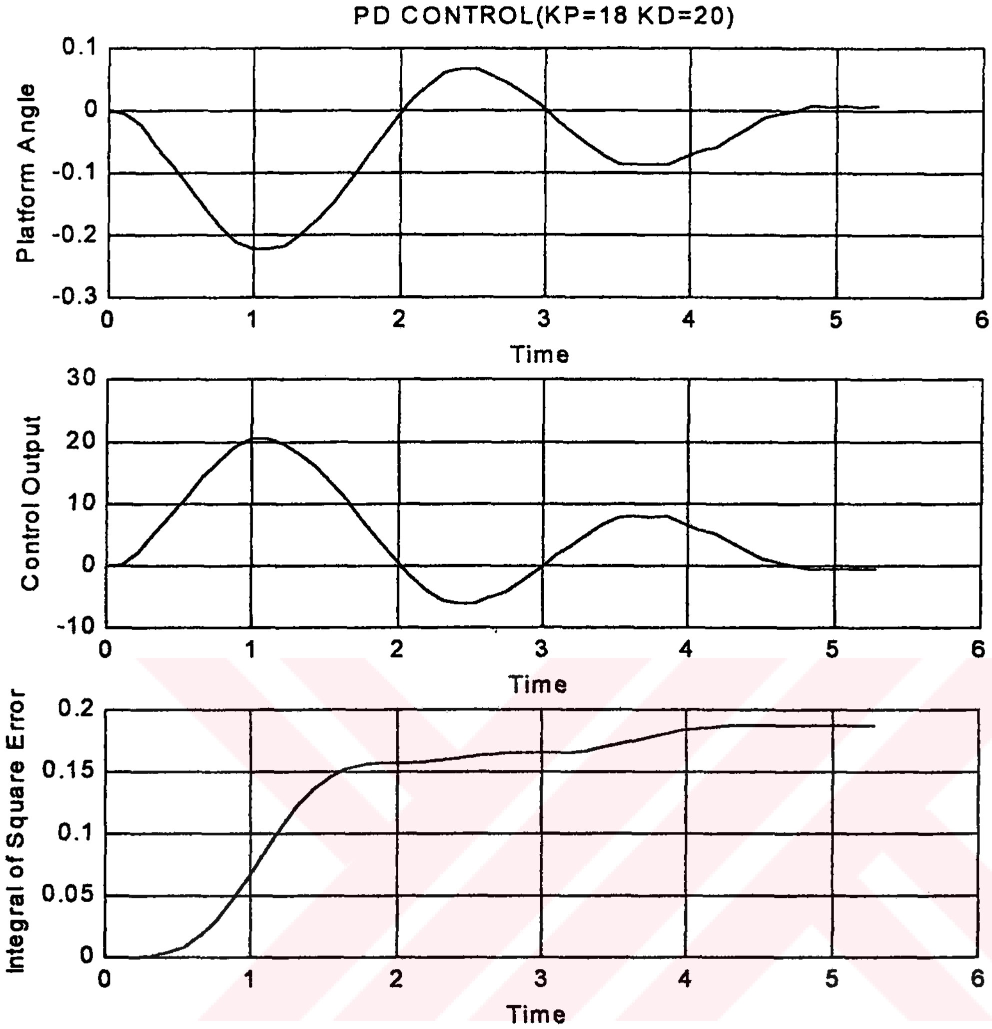
	Value	Time
Maximum	0.1038	2.3256
Minimum	-0.2093	0.9477
Steady State	-0.0079	14.1606

Figure 7. 6 Step Input (0.1991Nm) Response of the System (P=24)



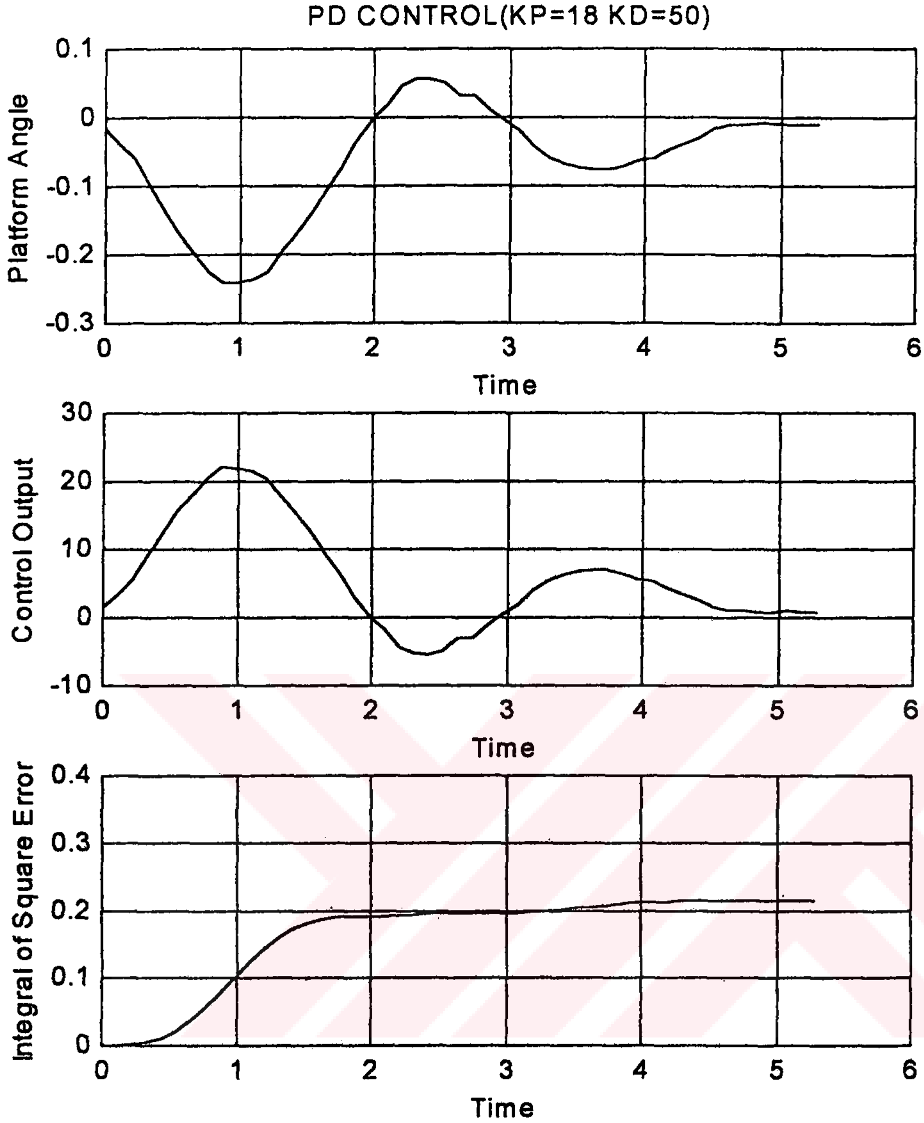
	Value	Time
Maximum	0.0704	2.4750
Minimum	-0.2333	1.1000
Steady State	-0.0044	4.6871

Figure 7. 7 Step Input (0.1991Nm) Response of the System (P=16 D=10)



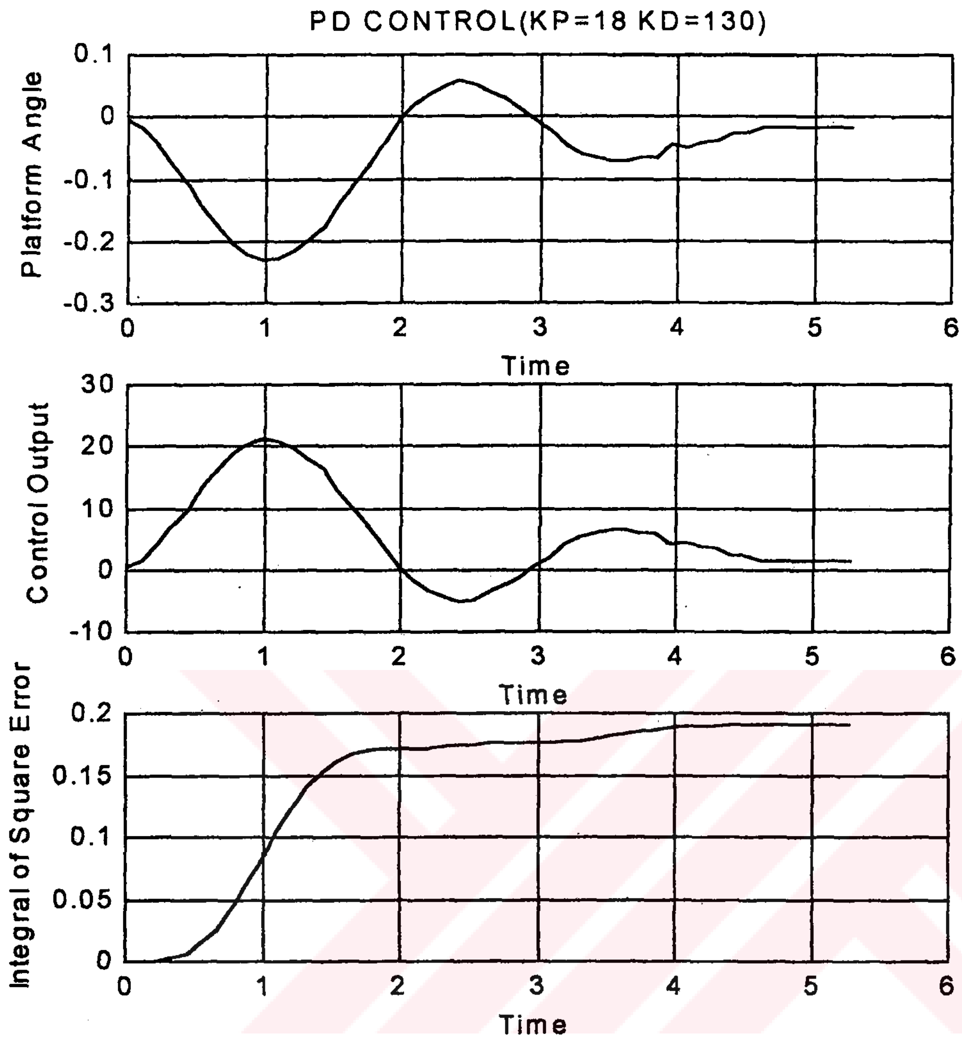
	Value	Time
Maximum	0.0692	2.4750
Minimum	-0.2241	1.0450
Steady State	-0.0036	4.8138

Figure 7. 8 Step Input (0.1991Nm) Response of the System (P=18 D=20)



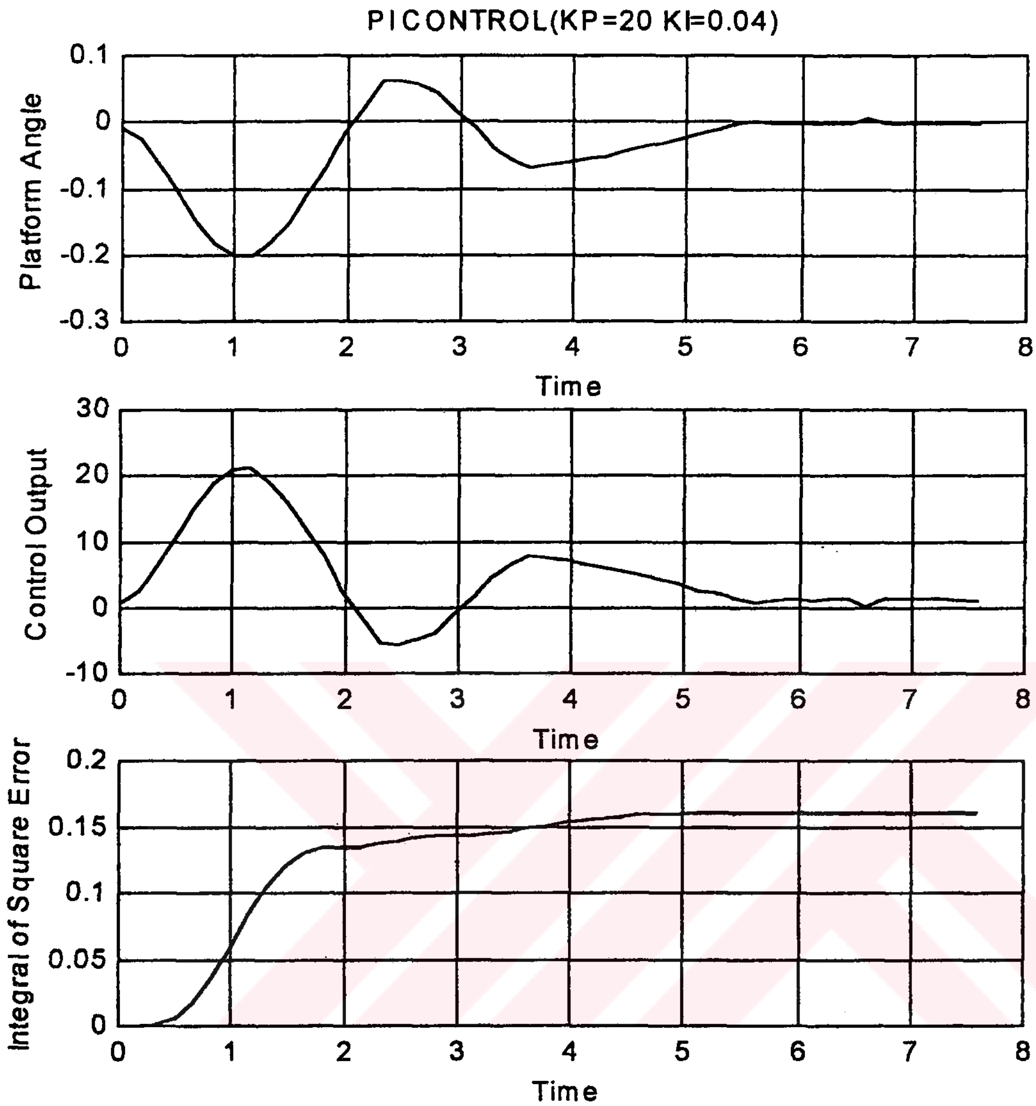
	Value	Time
Maximum	0.0577	2.4200
Minimum	-0.2414	0.8800
Steady State	-0.0128	4.6963

Figure 7. 9 Step Input (0.1991Nm) Response of the System (P=18 D=50)



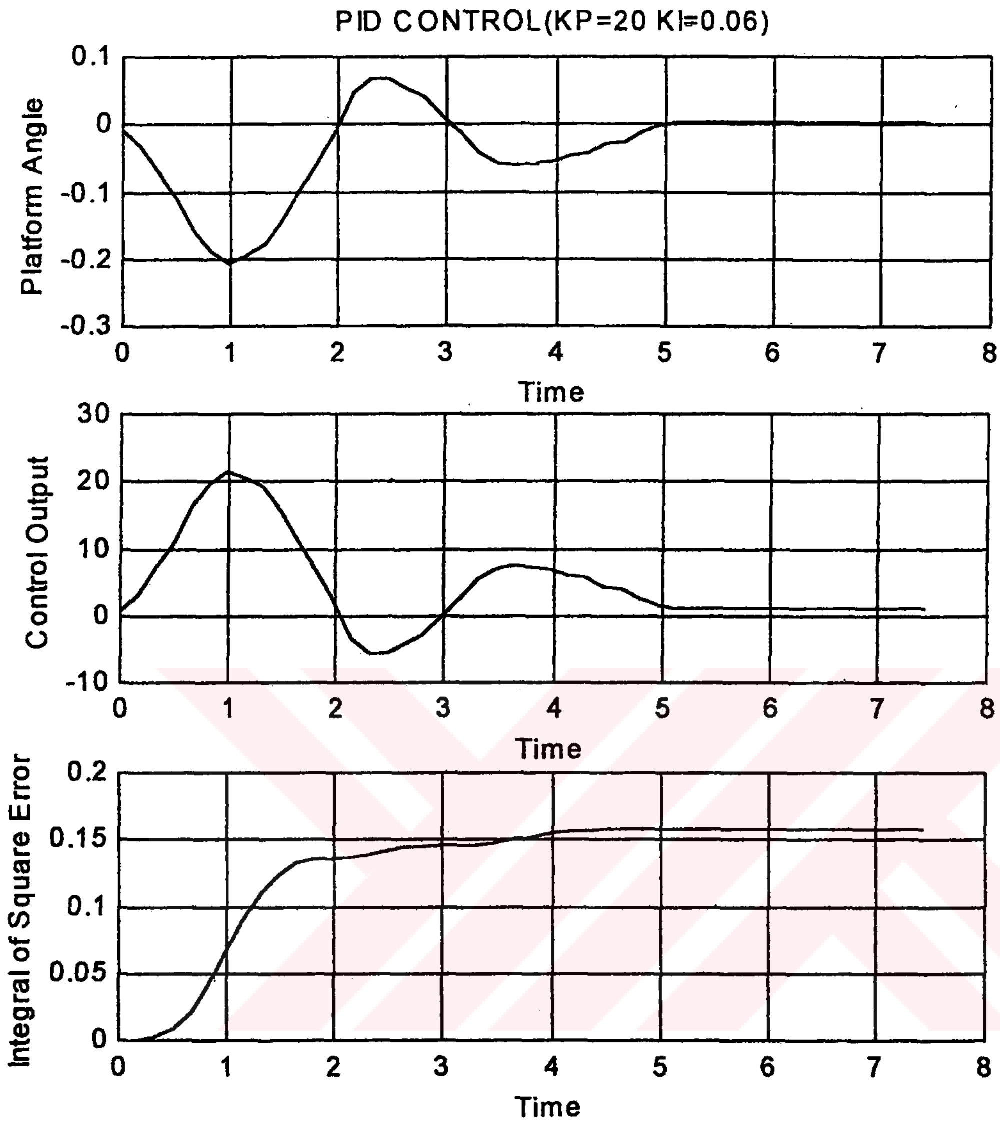
	Value	Time
Maximum	0.0566	2.3650
Minimum	-0.2322	1.0450
Steady State	-0.0194	4.6130

Figure 7. 10 Step Input (0.1991Nm) Response of the System (P=18 D=130)



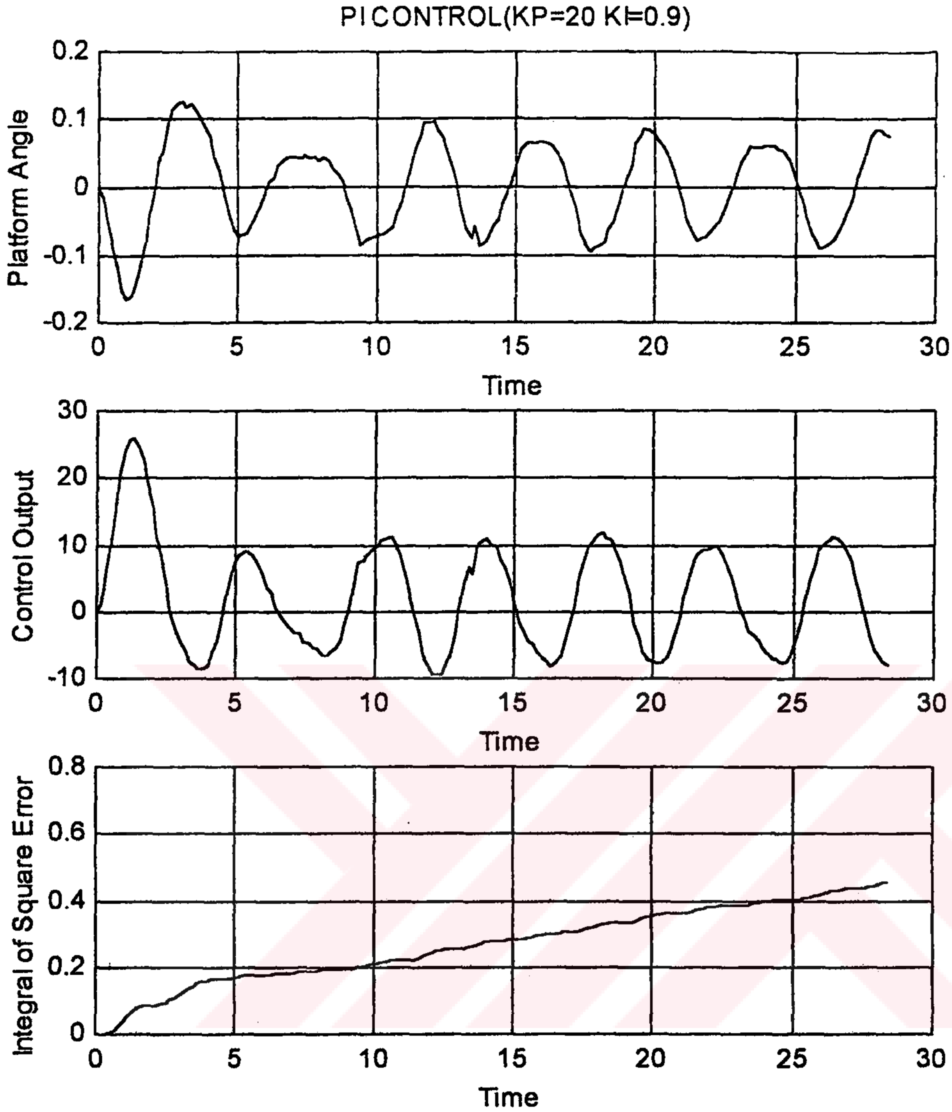
	Value	Time
Maximum	0.0681	2.5300
Minimum	-0.2046	1.0450
Steady State	-0.0043	5.5161

Figure 7.11 Step Input (0.1991Nm) Response of the System (P=20 I=0.04)



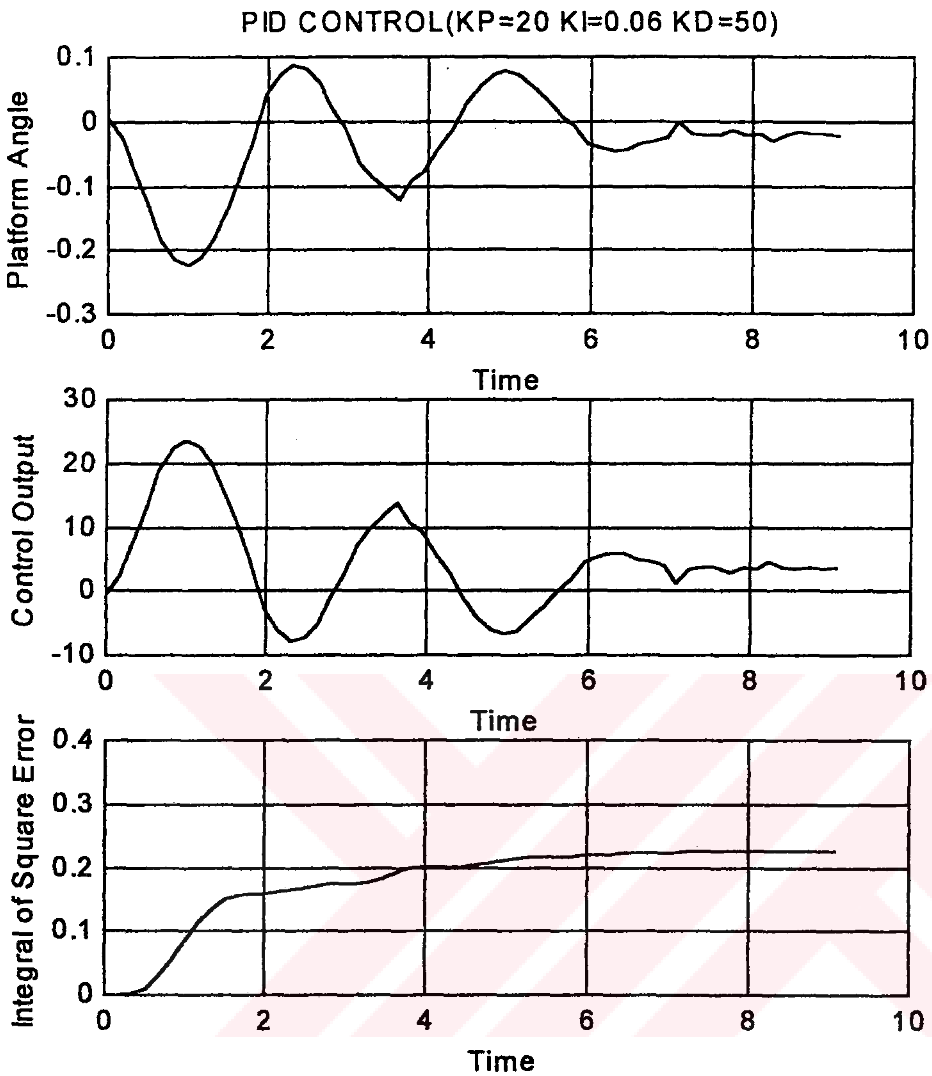
	Value	Time
Maximum	0.0669	2.3100
Minimum	-0.2057	0.9900
Steady State	-0.0014	5.0364

Figure 7. 12 Step Input (0.1991Nm) Response of the System (P=20 I=0.06)



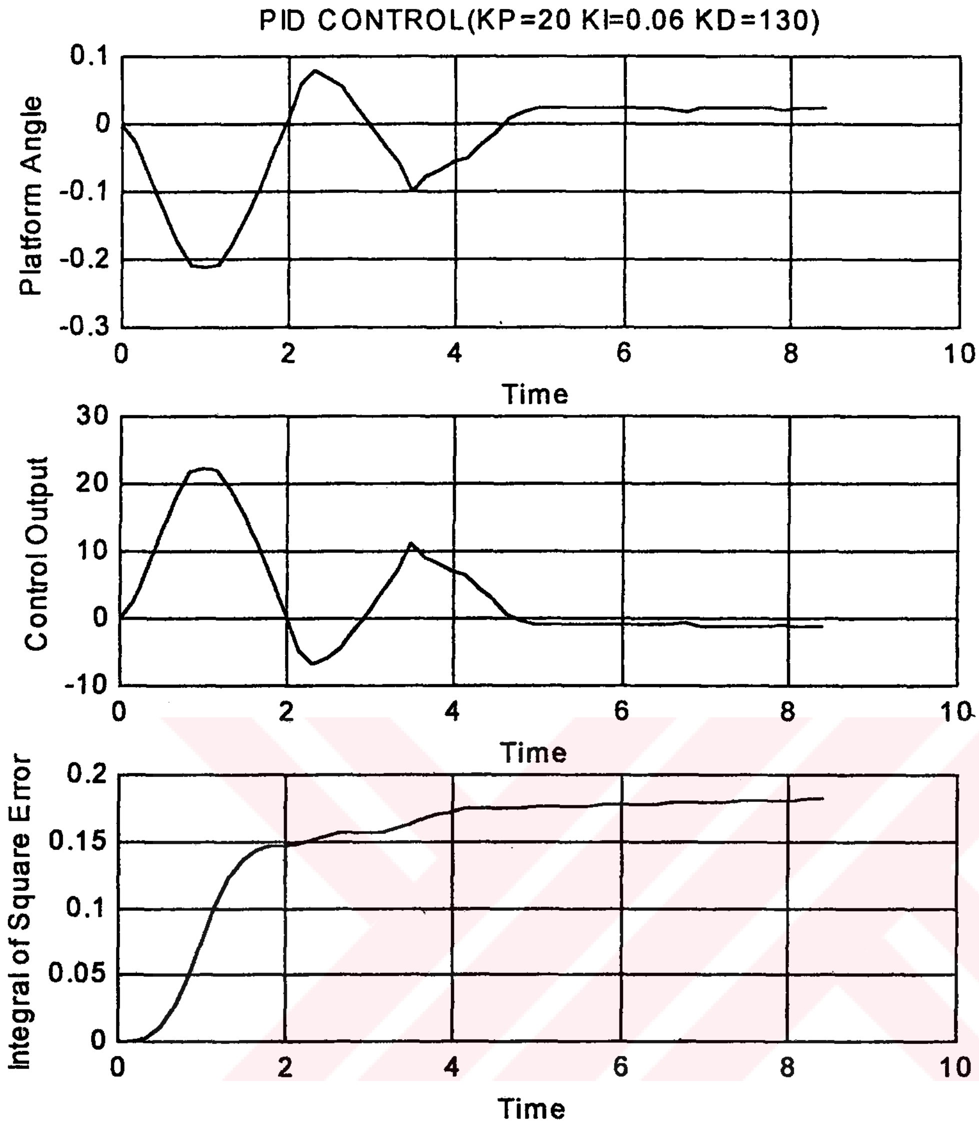
	Value	Time
Maximum	0.1268	2.9700
Minimum	-0.1655	0.9900
Steady State	-	-

Figure 7. 13 Step Input (0.1991Nm) Response of the System (P=20 I=0.9)



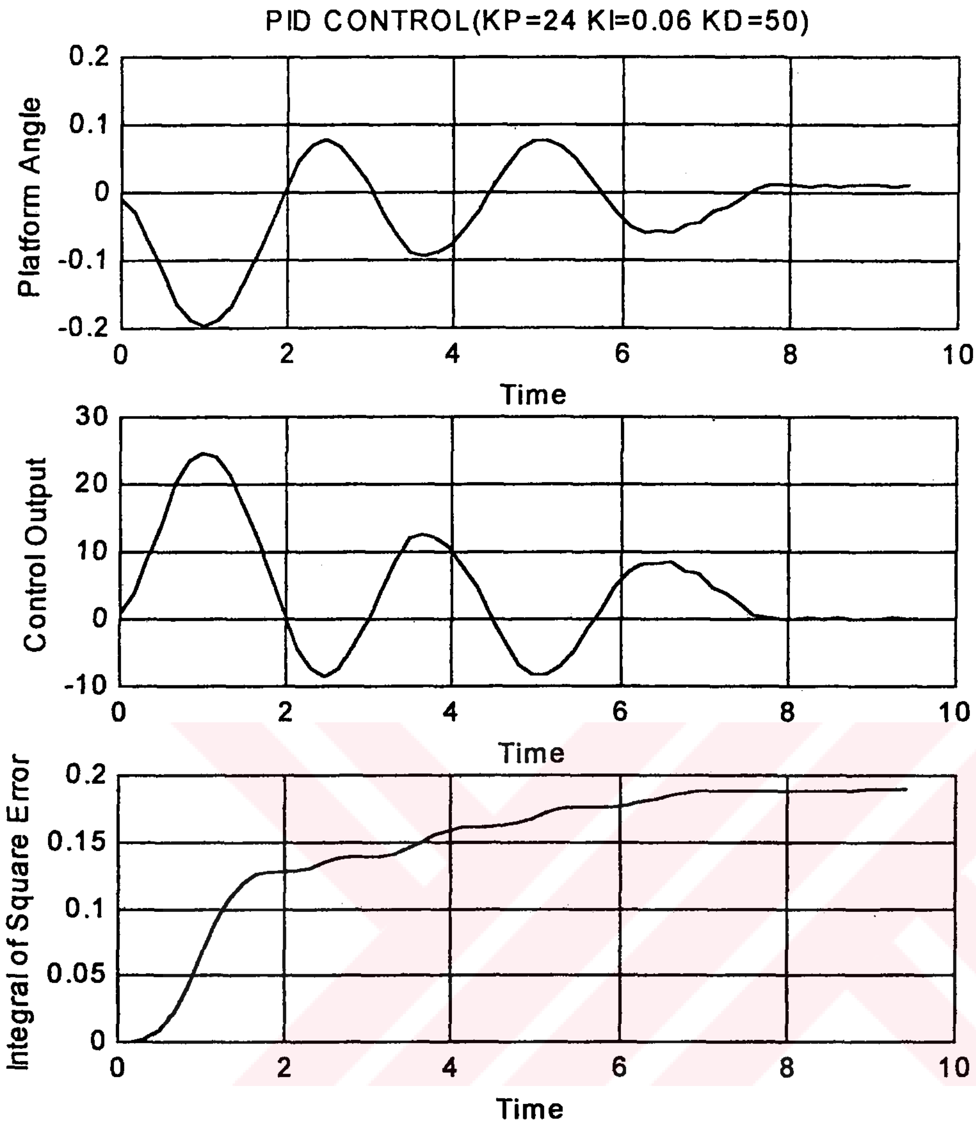
	Value	Time
Maximum	0.0877	2.3100
Minimum	-0.2414	0.9350
Steady State	-0.0221	7.2553

Figure 7. 14 Step Input (0.1991Nm) Response of the System (P=20 I=0.06 D=50)



	Value	Time
Maximum	0.0796	2.3100
Minimum	-0.2138	0.9350
Steady State	0.0217	5.0000

Figure 7. 15 Step Input (0.1991Nm) Response of the System (P=20 I=0.06 D=130)



	Value	Time
Maximum	0.0831	2.3650
Minimum	-0.2011	1.0450
Steady State	0.0102	7.7283

Figure 7. 16 Step Input (0.1991Nm) Response of the System (P=24 I=0.06 D=50)

To have a general view of the response, three important values and occurring times are used and evaluated. The maximum value is analogous to overshoot value, the minimum value shows starting dynamics of the controller and the minimum time is similar to rise time. The last parameters are steady state time and error and also integral of the square error is used as performance criteria to compare the results.

Proportional controller results are similar to expectations. Increasing gain also increases oscillation and maximum value. Better starting dynamics are observed for K_p=15 and K_p=24. Evidently K_p=24 has better minimum time as advantage. But for steady state time, it is clear that K_p=24 is quite long. For K_p=24 and K_p=20 ISE is bigger than the other, that means, indirectly, they used much power to reach reference. K_p= 18 has better steady state error and average maximum value and ISE but fast reaction than K_p=15. K_p=18 seems to be a reasonable choice for P controller.

PD controllers also have satisfying results. Because of the effect of the derivation part maximum and minimum values get lower also reaction and steady state times are shorter compared to other controllers. Then it is clear to choose highest "K_d " value is better for characteristics but in contrast higher the K_d becomes, more sensitive to noise it is. This is clearly seen in error curve of the K_d =50 and K_d =130 for K_p =18. For ensured stability K_d =50 and K_p =18 are selected as PD parameters but K_d =130 can be also a good choice.

PI controllers have more unpredictable reaction curves for this system calculated value of 0.4 is near to continuos oscillation limit. This situation is supposed to be related in mechanical problem caused by mountings and friction, as previously mentioned oscillation around "0" radian with small amplitudes are not possible. But for low K_i values system has proper responses. As expected, increase in K_i causes increase in time and values of the maximum and minimum. Because K_p =20 and K_i =0.04 has better results, they are supposed to be coefficients for PI controller.

For PID controller system acts better for higher K_d values. K_d =130 skips oscillation effect of the integral coefficient better than the others. System becomes faster maximum and minimum times are shorter. But maximum and minimum values are slightly larger than PI controller. Though controller with gains K_p=20 K_i =0.06 and K_d=130 has shorter steady state time, its steady state error is relatively large.

Comparing all controllers, the fastest controller is PD controller with K_d =18 K_d =130 and PI controller with K_p =20 K_d =0.04 has the best ISE value and the lowest steady state error. But PD controller , K_p =18 K_d =50, is more reliable than the two controller short steady state time, which also reduces the spent power to steady state, and less noise sensitivity.

CHAPTER EIGHT

FUZZY CONTROL THEORY

8.1 Introduction

The Fuzzy controllers are used to control consumer products, such as washing machines, video cameras, and rice cookers, as well as industrial processes, such as cement kilns, underground trains, and robots. Fuzzy control is a control method based on fuzzy logic. Just as fuzzy logic can be described simply as ''computing with words rather than numbers'', fuzzy control can be described simply as ''control with sentences rather than equations''.

A fuzzy controller can include empirical rules, and that is especially useful in operator controlled plants.

For instance a typical fuzzy controller

1. If error is Neg and change in error is Neg then output is NB
2. If error is Neg and change in error is Zero then output is NM

The collection of rules is called a rule base. The rules are in the familiar if-then format, and formally the if-side is called the condition and the then-side is called the conclusion (more often, perhaps, the pair is called antecedent-consequent or premise - conclusion). The input value "Neg'' is a linguistic term short for the word Negative the output value "NB'' stands for Negative big and ''NM'' for Negative Medium. The computer is able to execute the rules and compute a control signal depending on the measured inputs error and change in errors. In a rule based controller the control strategy is stored in a more or less natural language.

The control strategy is isolated in a rule base opposed to an equation based description. A rule based controller is easy to understand and easy to maintain for a non-specialist end-user.

An equivalent controller could be implemented using conventional techniques in fact, any rule based controller could be emulated in, say, C++ it is just that it is convenient. to isolate the control strategy in a rule base for operator controlled systems.

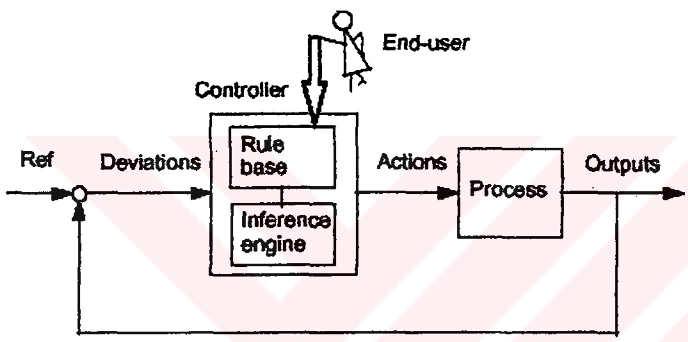


Figure 8.1 Direct Control

Fuzzy controllers are being used in various control schemes. The most obvious one is direct control, where the fuzzy controller is in the forward path in a feedback control system (Fig. 8.1). The process output is compared with a reference, and if there is a deviation, the controller takes action according to the control strategy. In the figure, the arrows may be understood as hyper-arrows containing several signals at a time for multi-loop control. The sub-components in the figure will be explained shortly. The controller is here a fuzzy controller, and it replaces a conventional controller.

In feed forward control a measurable disturbance is being compensated. It requires a good model, but if a mathematical model is difficult or expensive to obtain, a fuzzy model may be useful. Figure 8.2 shows a controller and the fuzzy compensator, the

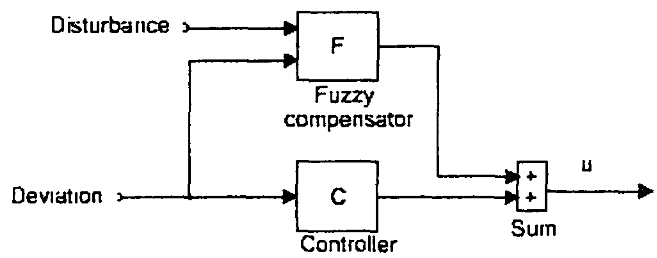


Figure 8.2 Feedforward Fuzzy Controller

process and the feedback loop are omitted for clarity. The scheme, disregarding the disturbance input, can be viewed as a collaboration of linear and nonlinear control actions; the controller C may be a linear PID controller, while the fuzzy controller F is a supplementary nonlinear controller

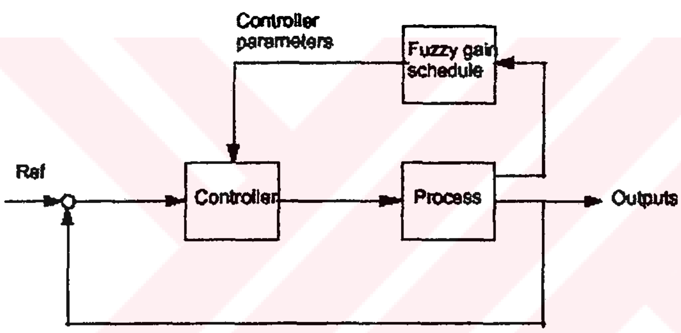


Figure 8.3 Fuzzy parameter adaptive control

Fuzzy rules are also used to correct tuning parameters in parameter adaptive control schemes (Figure 8.3). If a nonlinear plant changes operating point, it may be possible to change the parameters of the controller according to each operating point. This is called "Gain Scheduling" since it was originally used to change process gains. A gain scheduling controller contains a linear controller whose parameters are changed as a function of the operating point in a preprogrammed way. It requires thorough knowledge of the plant, but it is often a good way to compensate for non-linearity and parameter variations. Sensor measurements are used as scheduling variables that govern the change of the controller parameters, often by means of a table look-up.

Whether a fuzzy control design will be stable is a somewhat open question. Stability concerns the system's ability to converge or stay close to an equilibrium. A stable linear system will converge to the equilibrium asymptotically no matter where the system state variables start from. It is relatively straight forward to check for stability in linear systems, for example by checking that all eigenvalues are in the left half of the complex plane. For nonlinear systems, and fuzzy systems are most often nonlinear, the stability concept is more complex. A nonlinear system is said to be asymptotically stable if, when it starts close to an equilibrium, it will converge to it. Even if it just stays close to the equilibrium, without converging to it, it is said to be stable (in the sense of Lyapunov). To check conditions for stability is much more difficult with nonlinear systems, partly because the system behavior is also influenced by the signal amplitudes apart from the frequencies. The literature report on four methods (Lyapunov functions, Popov, circle, and conicity), and they give several references to scientific papers. It is characteristic, however, that the methods give rather conservative results, which translate into unrealistically small magnitudes of the gain factors in order to guarantee stability. Another possibility is to approximate the fuzzy controller with a linear controller, and then apply the conventional linear analysis and design procedures on the approximation. It seems likely that the stability margins of the nonlinear system would be close in some sense to the stability margins of the linear approximation depending on how close the approximation is.

There are at least four main sources for finding control rules;

- Expert experience and Control engineering knowledge: The most common approach to establishing such a collection of rules of thumb, is to question experts or operators using a carefully organized questionnaire.
- Based On Operator's Control Actions: Fuzzy if-then rules can be deduced from observations of an operator's control actions or a log book. The rules express input-output relationships.
- Based On Fuzzy Model of The Process: A linguistic rule base may be viewed as an inverse model of the controlled process. Thus the fuzzy control rules might be obtained by inverting a fuzzy model of the process. This

method is restricted to relatively low order systems, but it provides an explicit solution assuming that fuzzy models of the open and closed loop systems are available. Another approach is fuzzy identification or fuzzy model-based control.

- Based on Learning The self-organising controller is an example of a controller that finds the rules itself. Neural networks is another possibility.

There is no design procedure in fuzzy control such as root-locus design, frequency response design, pole placement design, or stability margins, because the rules are often nonlinear.

8.2 Fuzzy Logic

Conventional Boolean logic is limited in that it can be used only to describe attributes that are either completely true or completely false. Although it is useful for modeling many situations, conventional Boolean logic does not provide adequate means to model the many imprecise concepts used within human reasoning. Imprecise concepts are attributes of which people generally have a cognitive perception, yet are impossible to define precisely.

The human characteristic of tallness is defined as, "Having more than average height.". For this example, that the average height of people is 170cm is assumed. Thus, using conventional Boolean logic, tall must be defined as

$$tall(x) = \begin{cases} 0 & x \geq 170 \\ 1 & x < 170 \end{cases} \tag{8.1}$$

It is the strict separation between tall and not tall required by conventional Boolean logic that renders it inadequate for describing imprecise concepts like tallness. Because 170 cm is considered as certainly tall and 169 cm is certainly not tall.

Fuzzy logic is a superset of conventional Boolean logic that has been extended to handle the concept of partial truth. Rather than limiting degrees of membership for a particular attribute to the discrete values 0 or 1 as in conventional Boolean logic, fuzzy logic allows degrees of membership to have values within the continuous range 0 to 1. In this way, it is possible to assign a degree of membership for a particular attribute that is neither completely true nor completely false, but somewhere in between. Many of the same logical concepts still apply (such as AND, OR & NOT), but carry slightly different meanings.

In 1973, Professor Lotfi Zadeh proposed the concept of linguistic or "fuzzy" variables. Think of them as linguistic objects or words, rather than numbers. The sensor input is a noun, e.g. "temperature", "displacement", "velocity", "flow", "pressure", etc. Since error is just the difference, it can be thought of the same way. The fuzzy variables themselves are adjectives that modify the variable (e.g. "large positive" error, "small positive" error ,"zero" error, "small negative" error, and "large negative" error). As a minimum, one could simply have "positive", "zero", and "negative" variables for each of the parameters. Additional ranges such as "very large" and "very small" could also be added to extend the responsiveness to exceptional or very nonlinear conditions, but aren't necessary in a basic system.

Since fuzzy logic is an extension of conventional Boolean logic, it is not surprising that the two theories share many of the same concepts. Just as there is a strong relationship between Boolean logic and the concept of a subset, there is a similar strong relationship between fuzzy logic and fuzzy subset theory

8.2.1 Fuzzy Subsets

For any fuzzy logic implementation, it must be understand the way how entities are assigned a degree of membership for a particular attribute. This is done using fuzzy subsets. Let S be a non-empty set. A fuzzy subset μ in S is characterized by its membership function:

$$\mu : S \rightarrow [0,1] \tag{8.2}$$

and μ (x) is interpreted as the degree of membership of element x in fuzzy set μ for each x in S."1" is used to represent complete membership,"0" is used to represent complete non membership. All values in between are used to represent intermediate degrees of membership. S is referred to as the universe of discourse. The mapping μ is also called the member-ship function of the fuzzy set.

The transformation of an attribute into a fuzzy concept is called fuzzification. The fuzzification of the attribute "tall" may be defined as in a very basic form:

$$tall(x) = \begin{cases} 0 & x < 160 \\ \dfrac{x-160}{20} & 160 \le x \le 180 \\ 1 & x > 180 \end{cases} \tag{8.3}$$

8.2.2 Logical Operations on Fuzzy Subsets

Although the logical operations NOT, AND, & OR from conventional Boolean logic also apply to fuzzy logic, the interpretations of these operations are slightly different. Since fuzzy logic researchers have explored the use of several different interpretations of the logical AND and OR operations, the most-widely accepted interpretation of each is represented in the following.

8.2.2.1 Complement

Let S be a non-empty set. The complement (NOT) of a fuzzy subset F in S is defined as

$$\neg\mu(x) = 1 - \mu(x) \tag{8.4}$$

8.2.2.2 Intersection

Let S be a non-empty set. The intersection (AND) of fuzzy subsets μ_1 and μ_2 in S is defined as

$$(\mu_1 \cap \mu_2)(x) = \min(\mu_1(x), \mu_2(x)) \tag{8.5}$$

8.2.2.3 Union

Let S be a non-empty set. The union (OR) of fuzzy subsets F and G in S is defined as

$$(\mu_1 \cup \mu_2)(x) = \max(\mu_1(x), \mu_2(x)) \tag{8.6}$$

When the only degrees of membership used are 0 and 1, the same truth tables as you would expect from conventional Boolean logic is obtained. This is phenomenon is known as the extension principle, which states that the traditional results of Boolean logic are recovered from fuzzy logic operations when all degrees of membership are restricted to the traditional set {0, 1}. Since this ensures that there are no conflicts between fuzzy logic and traditional Boolean logic, the extension principle effectively establishes and fuzzy logic as a true generalization of traditional Boolean logic.

8.3 Structure of Fuzzy Controller

There are specific components characteristic of a fuzzy controller to support a design procedure. In the block diagram in figure 8.4, the controller is between a preprocessing block and a post-processing block. The following explains the diagram block by block.

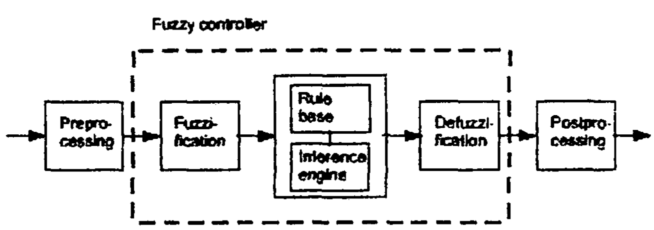


Figure 8.4 Blocks of Fuzzy Controller

8.3.1 Preprocessing

The inputs are most often hard or crisp measurements from some measuring equipment, rather than linguistic. A preprocessor, the first block in Fig. 4, conditions the measurements before they enter the controller. Examples of preprocessing are:

- Quantisation in connection with sampling or rounding to integers;
- Normalization or scaling onto a particular, standard range;
- Filtering in order to remove noise;
- A combination of several measurements to obtain key indicators; and
- Discrete equivalencies of differentiation and integration.

A quantiser is necessary to convert the incoming values in order to find the best level in a discrete universe. Assume, for instance, that the variable error has the value 0.5323, but the universe is u={-1.000,-0.950,......,0,950,1.000} The quantiser rounds 0.532 to fit it to the nearest level. Quantisation is a means to reduce data, but if the quantisation is too coarse the controller may oscillate around the reference or even become unstable.

When the input to the controller is error, the control strategy is a static mapping between input and control signal. A dynamic controller would have additional inputs, for example derivatives, integrals, or previous values of measurements backwards in time. These are created in the preprocessor thus making the controller multi-dimensional, which requires many rules and makes it more difficult to design. The preprocessor then passes the data on to the controller.

8.3.2 Fuzzification

The first block inside the controller is fuzzification, which converts each piece of input data to degrees of membership by a lookup in one or several membership functions. The fuzzification block thus matches the input data with the conditions of the rules to determine how well the condition of each rule matches that particular input instance. There is a degree of membership for each linguistic term that applies to that input variable.

8.3.3 Rule Base

The rules may use several variables both in the condition and the conclusion of the rules. The controllers can therefore be applied to both multi-input-multi-output (MIMO)problems and single-input-single-output (SISO) problems. The typical SISO problem is to regulate a control signal based on an error signal. The controller may actually need both the error, the change in error ,and the accumulated error as inputs, anyway it will called single-loop control, because in principle all three are formed from the error measurement. This section assumes that the control objective is to regulate some process output around a prescribed set point or reference. The presentation is thus limited to single-loop control according to our model.

Rule formats, basically a linguistic controller contains rules in the if-then format, but they can be presented in different formats. In many systems, the rules are presented as following:

"If error is Neg and change in error is Neg then output is NB"

or in the tabular linguistic format:

Table 8.1 Tabular linguistic format

		Change in error		
		Neg	Zero	Pos
	Neg	NB	NM	Zero
Error	Zero	NM	Zero	PM
	Pos	Zero	PM	PB

Connectivities: In mathematics, sentences are connected with the words and, or, if-then or modifications with the word not. These three are called connectivities. It also makes a difference how the connectives are implemented. The most prominent is probably multiplication for fuzzy "and" instead of minimum.

The connectives "and" and "or" are always defined in pairs, for example,

$$a \text{ and } b = \min(a,b) \qquad \text{or} \qquad a \text{ or } b = \max(a,b)$$

Modifiers: A linguistic modifier, is an operation that modifies the meaning of a term. For example, in the sentence "very close to 0", the word "very" modifies close to 0 which is a fuzzy set. A modifier is thus an operation on a fuzzy set. The modifier "very" can be defined as squaring the subsequent membership function, that is very $a=a^2$.

A whole family of modifiers is generated by a^p where p is any power between zero and infinity. With p is the modifier could be named exactly, because it would suppress all memberships lower than 1.0.

Universes: Elements of a fuzzy set are taken from a universe of discourse or just universe. The universe contains all elements that can come into consideration. Before designing the membership functions it is necessary to consider the universes for the inputs and outputs.

Another consideration is whether the input membership functions should be continuous or discrete. A continuous membership function is defined on a continuous universe by means of parameters. A discrete membership function is defined in terms of a vector with a finite number of elements. In the latter case it is necessary to specify the range of the universe and the value at each point. The choice between fine and coarse resolution is a trade off between accuracy, speed and space demands. The quantiser takes time to execute, and if this time is too precious, continuous membership functions will make the quantiser obsolete. Some examples fuzzy universes are given in folowing:

- The basic fuzzy logic controller uses the real number interval [-1,1].
- [-100,100] is the another possible universe that corresponding to percentages of full scale.
- Integer range [0,4095] corresponding to the output from a 12 bit analog to digital converter(ADC).(or [–2047,2048] which is possible setting for ADC)

Any of the possibilities or ones which are more suitable to our input and out put scales can be used.

Membership functions: Every element in the universe of discourse is a member of a fuzzy set to some grade, maybe even zero. The grade of membership for all its members describes a fuzzy set. In fuzzy sets elements are assigned a grade of membership, such that the transition from membership to non-membership is gradual rather than abrupt. The set of elements that have a non-zero membership is called the support of the fuzzy set. The function that ties a number to each element x of the universe is called the membership function $\mu(x)$.

According to fuzzy set theory the choice of the shape and width is subjective, but a few rules are taken in to consideration

- A term set should be sufficiently wide to allow for noise in the measurement.
- A certain amount of overlap is desirable; otherwise the controller may run into poorly defined states, where it does not return a well defined output.

The necessary and sufficient number of sets in a family depends on the width of the sets, and vice versa. The procedure for starting fuzzy controller design is given below

Start with triangular sets. All membership functions for a particular input or output should be symmetrical triangles of the same width. The leftmost and the rightmost should be shouldered ramps. The overlap should be at least 50%.The widths should initially be chosen so that each value of the universe is a member of at least two sets, except possibly for elements at the extreme ends. If, on the other hand, there is a gap between two sets no rules fire for values in the gap. Consequently the controller function is not defined.

Membership functions can be flat on the top, piece-wise linear and triangle shaped, rectangular, or ramps with horizontal shoulders. Fig. 8.5 shows some typical shapes of membership functions.

Strictly speaking, a fuzzy set A is a collection of ordered pairs Item x belongs to the universe and $\mu(x)$ is its grade of membership in A.

$$A=\{(x,\mu(x))\} \tag{8.7}$$

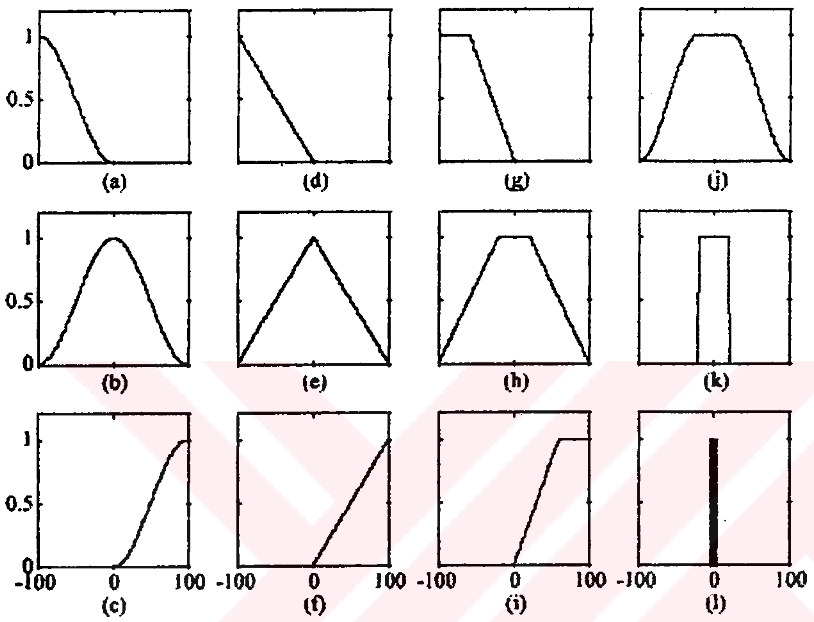


Figure 8.5 Examples of membership functions

8.3.4 Inference Engine

The rules reflect the strategy that the control signal should be a combination of the each input (eg. error ,change in error a fuzzy proportional-derivative controller).The instances of the each input are indicated by conditional pairs of the rules. For each rule, the inference engine looks up the membership values in the condition of the rule.

Aggregation: the aggregation operation is used when calculating the degree of the fulfillment or firing strength α_n of the condition of a rule n. A rule ,say rule 1 (n=1), will generate a fuzzy membership value μ_{ce} coming from error and a membership value μ_{ce1} coming from change in error. The aggregation is their combination,

$$(\mu_{ce} \text{ and } \mu_{ce1}) \tag{8.8}$$

Activation: Activation of a rule is the deduction of the conclusion, possibly reduced by its firing strength. and "min" is used as the activation operator. A rule n can be weighted a priori by a weighting factor ϖ_n : [0,1], which is its degree of confidence. In that case the firing strength is modified to

$$\alpha_n = \varpi_n \cdot \alpha_n \tag{8.9}$$

The degree of confidence is determined by the designer, or a learning program trying to adapt the rules to some input-output relationship.

Accumulation: all activated conclusions are accumulated, using the max operation. All these steps are shown in figure 8.6

$$(\alpha_n \text{ or } \alpha_{n-1}) \tag{8.10}$$

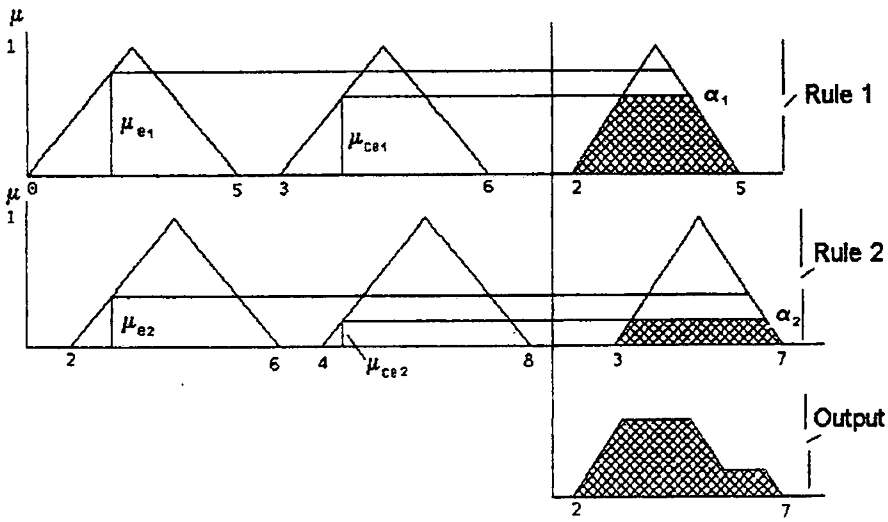


Figure 8.6 Fuzzy inference engine (aggregation, activation, accumulation)

8.3.5 Defuzzification

The resulting fuzzy set (Fig. 6, bottom right) must be converted to a number that can be sent to the process as a control signal. This operation is called "defuzzification". The resulting fuzzy set is thus defuzzified into a crisp control signal. There are several defuzzification methods for this operation.

8.3.5.1 Center of Gravity (COG)

The crisp output value u is the abscissa under the center of gravity of the fuzzy set,

$$u = \frac{\sum_i \mu(x_i).x_i}{\sum_i \mu(x_i)} \tag{8.11}$$

Here x_i is a running point in a discrete universe, and $\mu(x_i)$ is its membership value in the membership function. The expression can be interpreted as the weighted average of the elements in the support set. For the continuous case, replace the summations by integrals. It is a much used method although its computational complexity is relatively high. This method is also called "Centroid Of Area".

8.3.5.2 Bisector Of Area (BOA)

This method picks the abscissa of the vertical line that divides the area under the curve in two equal halves. In the continuous case,

$$u = \left\{ x \mid \int_{Min}^{x} \mu(x).dx = \int_{x}^{Max} \mu(x).dx \right\} \tag{8.12}$$

Here x is the running point in the universe, $\mu(x_i)$ is its membership, Min the leftmost value of the universe, and Max is the rightmost value. Its computational complexity is relatively high, and it can be ambiguous. For example, if the fuzzy set consists of two singletons any point between the two would divide the area in two halves; consequently it is safer to say that in the discrete case, BOA is not defined.

8.3.5.3 Mean Of Maxima (MOM)

An intuitive approach is to choose the point with the strongest possibility, i.e. maximal membership. It may happen, though, that several such points exist, and a common practice is to take the mean of maxima (MOM). This method disregards the shape of the fuzzy set, but the computational complexity is relatively good.

8.3.5.4 Leftmost Maximum (LM) and Rightmost Maximum

Another possibility is to choose the leftmost maximum (LM), or the rightmost maximum (RM). In the case of a robot, for instance, it must choose between left or right to avoid an obstacle in front of it. The defuzzifier must then choose one or the other, not something in between. These methods are indifferent to the shape of the fuzzy set, but the computational complexity is relatively small.

8.3.6 Postprocessing

Output scaling is also relevant. In case the output is defined on a standard universe this must be scaled to engineering units, for instance, volts, meters or tons per hour An example is the scaling from the standard universe [-1,1] to the physical units [-5V,5V] volts. The postprocessing block often contains an output gain that can be tuned, and sometimes also an integrator.

CHAPTER NINE

FUZZY CONTROL OF ROTATING PLATFORM

9.1 Design of Reasonable Fuzzy Controller

An expert system is a computer program that simulates the judgement and behavior of a human or an organization that has expert knowledge in a particular field. Expert systems typically consist of a knowledge base containing accumulated experience data, and a set of rules for applying the knowledge base to each situation encountered by the program. A fuzzy expert system is simply an expert system that uses fuzzy logic, rather than traditional Boolean logic, for making decisions using its set of rules. The general process used by fuzzy expert systems to make decisions proceeds in four steps - fuzzification, inference, composition, and defuzzification .

The method, that is used in this application, is direct control, where the fuzzy controller is in the forward path in a feedback control system. The process output is compared with a reference, and if there is a deviation, the controller takes action according to the control strategy. The fuzzy controller replaces the conventional PID controller designed previously. Fuzzy control parameters are normalized error and change in error as inputs and normalized pulse width modulation voltage as output. The rule database is set according this three parameter.

Fuzzy control stability has not been developed sufficiently as mentioned in previous chapter. In this study it is predicted that the fuzzy controller which is replaced with PD controller which converges to an equilibrium, also converges to same point (horizontal position of platform) since error and change in error, analogues to derivative, is chosen as input parameters for fuzzy control similar to PD controller.

Main source for finding control rules is experience that we had while trying to design PID controller and analyzing the time response of the system. Basic requirements for system performance are fast reaction to disturbance and avoid oscillations.

9.1.1 Fuzzy Subsets

For fuzzy logic implementation, membership function for input and output of controller must be precisely determined to design fuzzy subsets for each parameter then optimizing them according to response of the controlled system.

It is easy to find absolute value of maximum of input parameters, experimentally, as the error and change in error range are determined as "0.25" for error and "0.05" for change in error. On the other side output of the controller is normalized PWM voltage, which is multiplied by "5". These values are the first step for determining coefficients for fuzzy controller.

$$ERR_n = \frac{ERR}{ERR_m} \tag{9.1}$$

$$DERR_n = \frac{DERR}{DERR_m} \tag{9.2}$$

$$C = 5 \times C_n \tag{9.3}$$

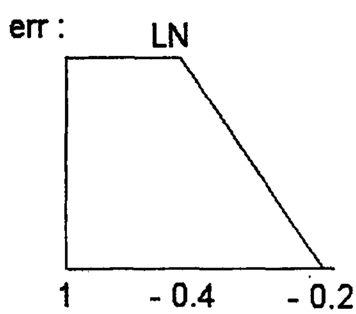


Figure 9. 1 Membership Function of Large Negative Error

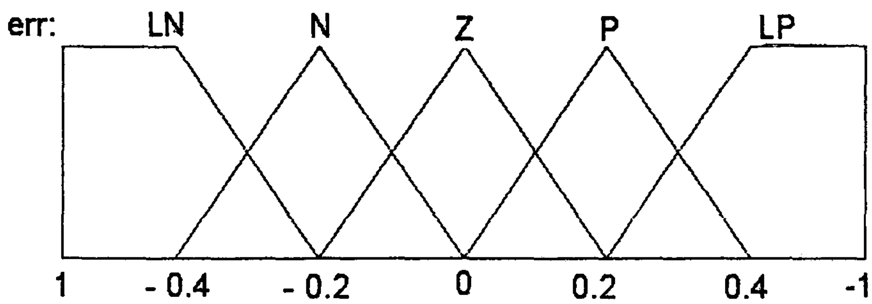
Figure 9.1 represents the large negative error membership function that is used in this study and the mathematical representation of this is also given in equation (9.4),as an example.

Linear shapes are chosen with no conflict to general trend because they have great easiness to calculate both membership values and areas. Shoulders is added to limit values for making them more dominant in output values.

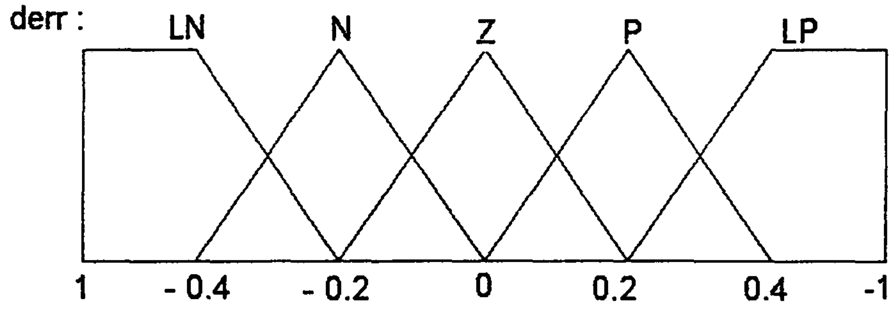
$$\mu(err) = \begin{cases} err < -0.2 & 0 \\ 0.4 < err < -0.2 & \dfrac{-0.2 - err}{0.1} \\ err > 0.4 & 1 \end{cases} \tag{9.4}$$

Number and width of membership functions is one of the basic design problems of fuzzy expert system design. In this study, at the beginning, general concern that 100 percent is large, 50 percent is normal and 0 percent is zero is applied for each three linguistic variable. But operating the system, it is seen then the response is slow and unable to dismiss error. In further experiments, by tightening the triangles and approaching them around zero. This makes controller faster because the higher input parameters have more effect on output.

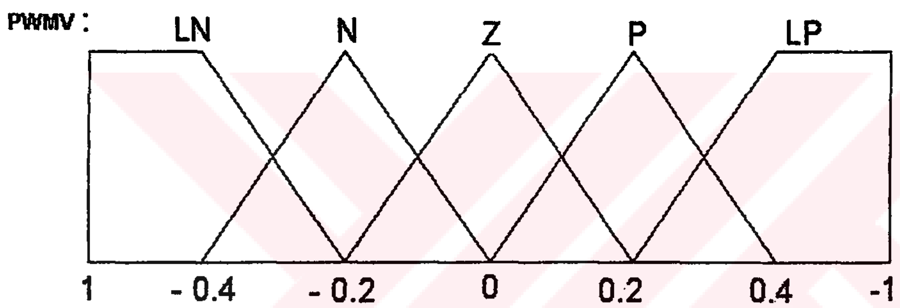
After several tests proper values are determined for fuzzy controller parameters. Determined fuzzy sets are given in following as fuzzy membership functions for error, change in error and PWM voltage.



(a)



(b)



(c)

Figure 9. 2 Membership functions (a) for error (err) (b) for delta error (derr)

Till this point, preprocessing and the fuzzification procedures are defined according to target, balancing the platform. The second important aspect is to build a rule base

9.1.2 Rule Base

A rule base is the most critical concept of fuzzy logic. It is aimed to simulate operator's behavior-means that controller will act like him- which is great expectation for a set of plastic and metal equipment. In anyway, observations helps to determine the draft characteristics of controller and experiments are used for optimizing.

The fuzzy parameters of error (ERR) and delta error (DERR) were modified by the adjectives " large negative (LN)", "negative (N)", "zero (Z)", “positive (P)" and " large positive (LP)". To picture this, imagine the simplest practical implementation, a 5-by-5 matrix. The columns represent “delta error” inputs from left to right. The rows represent "error" input from top to bottom. This planar construct is rule matrix. It has two input conditions, and one output response conclusion (at the intersection of each row and column).Although not absolutely necessary, rule matrices usually have an odd number of rows and columns to accommodate a "zero" center row and column region. Since the "zero" regions correspond to "no change" output responses the lack of this region will cause the system to continually look for "zero". It is also possible to have a different number of rows than columns. This occurs when numerous degrees of inputs are needed. The maximum number of possible rules is simply the product of the number of rows and columns, but definition of all of these rules may not be necessary since some input conditions may never occur in practical operation.

The primary objective of this construct is to map out the universe of possible inputs while keeping the system sufficiently under control. When observing the balance mechanism it is clearly realized that two possible motion mode can occur in relation to system accuracy lets say damping ratio. One is that the load is applied and the balancing mass goes opposite side and platform angle is decreasing first fast then slower and at the point close to zero degree and change is slowest, the output is zero. Other is try to get both platform angle and the movement rate are zero and going clockwise and counter clock wise around zero. The first action what is similar to overdamped response of control systems is desired performance for this mechanism and the rule base are constructed with respect to this.

Linguistic rules describing the control system consist of two parts; an antecedent block (between the IF and THEN) and a consequent block (following THEN). Depending on the system, it may not be necessary to evaluate every possible input combination (for 5-by-5 & up matrices) since some may rarely or never occur. By making this type of evaluation, usually done by an experienced operator, fewer rules can be evaluated, thus simplifying the processing logic and perhaps even improving the FL system performance. After transferring the conclusions from the nine rules to the matrix there is a noticeable symmetry to the matrix. This suggests reasonably well-behaved (linear) system.

Rule base is also constructed according to conditions described above but during experiments some corrections are made according to have better output.

Table 9. 1 Rule Base for Fuzzy Controller

ERR\ DERR	LN	N	Z	P	LP
LN	LP	LP	LP	Z	Z
N	LP	P	P	P	Z
Z	P	Z	Z	Z	N
P	Z	N	N	N	LN
LP	Z	Z	LN	LN	LN

9.2 Inference and Defuzzification

9.2.1 Inference

Under inference, the degree of truth for the premise of each rule is determined using the method AND described in for working with logical operations on fuzzy subsets. This result is then assigned to the conclusion part of each rule. Inference results in one fuzzy value being assigned to the fuzzy subset in the conclusion part of each rule, for each rule. Since the same fuzzy subset may appear in the conclusion part of more than one rule, it is possible that some fuzzy subsets will have more than one fuzzy value assigned to it.

The logical products for each rule must be combined or inferred before being passed on to the defuzzification process for crisp output generation. As previously mentioned several inference methods exist. In this experiment output is inferred by fuzzy maximum. This method combines all outputs of the activated inputs and constitutes a complex area

But for triangular sets controller output was not satisfying. Long operation time and smooth transitions caused insufficient reactions for stabilizing system. More simple and easy to work singletons are selected as output functions.

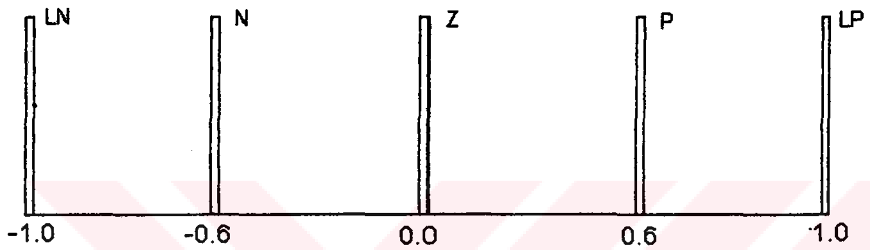


Figure 9. 3 Fuzzy Impuls Output Functions of Fuzzy Controller

9.3 Defuzification

The defuzzification of the data into a crisp output is accomplished by combining the results of the inference process and then computing the centroid of the area. The weighted strengths of each output member function are multiplied by their respective output membership function center points and summed. Finally, this area is divided by the sum of the weighted member function strengths and the result is taken as the crisp output. For singletons, this algorithm works like simple average calculation.

9.4 Implementation Of The Fuzzy Controller

Three different fuzzy controller algorithm is applied to system; Fuzzy controller, determined above, which has error and change in error as input and output as PWM voltage, Fuzzy P controller which determines proportional coefficient for conventional P controller and Fuzzy P+D controller which combines previous controller with a derivative controller.

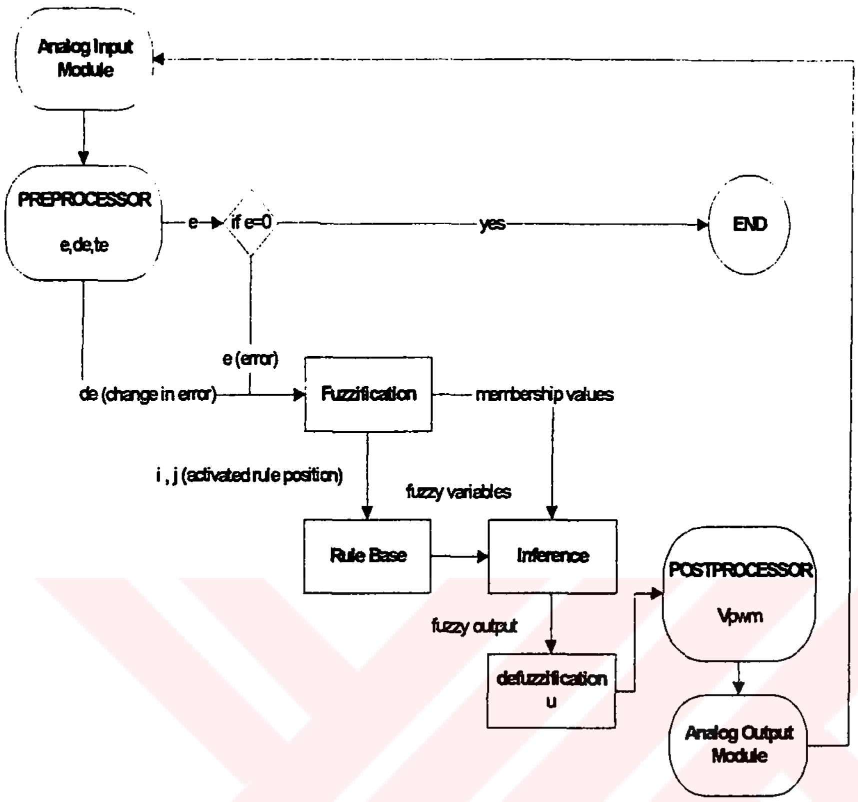


Figure 9. 4 Computer Program for Fuzzy Controller

The first control uses input membership functions as given above. Scaling factors of the inputs and output are ; 12.5 (5/0.4) for error and 100 (5/0.05) for change in error. For the second method only normalized error is selected as input of fuzzy controller and proportional coefficient for P controller is the output of the controller.

The output functions are designed to change the proportional coefficient which is always positive. The problem with proportional control is the small output signal in small errors which causes permanent error in steady state. Because of this generally an I controller is added to P controller but it is also problem because it increases the oscillation. In design of fuzzy P controller output functions are formed to increase proportional coefficient while error is getting smaller. It is possible to obtain a Fuzzy P+D controller by inserting Derivative part to controller.

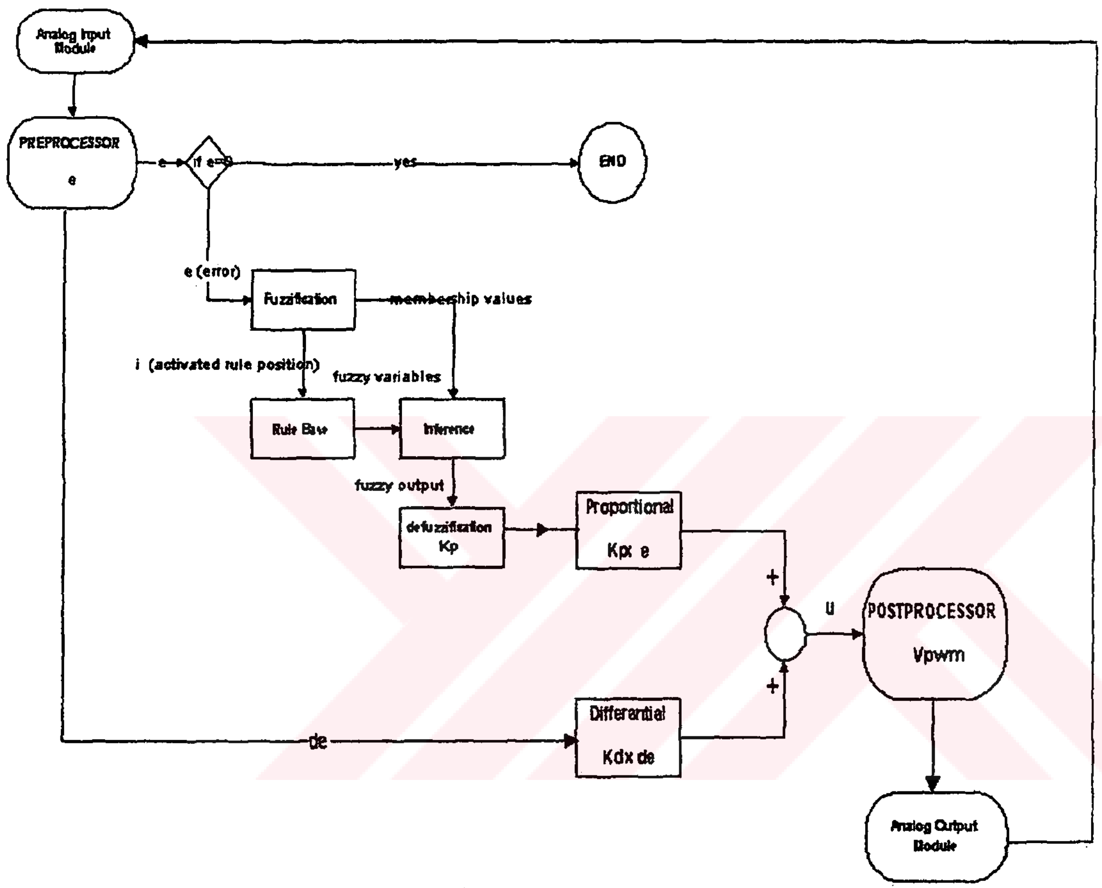
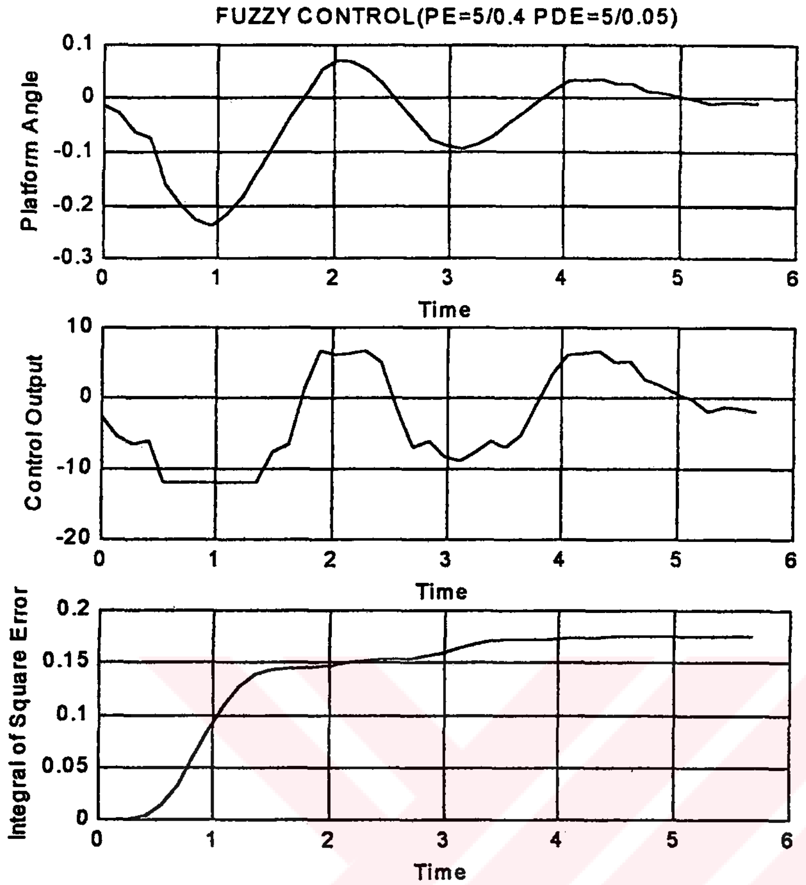
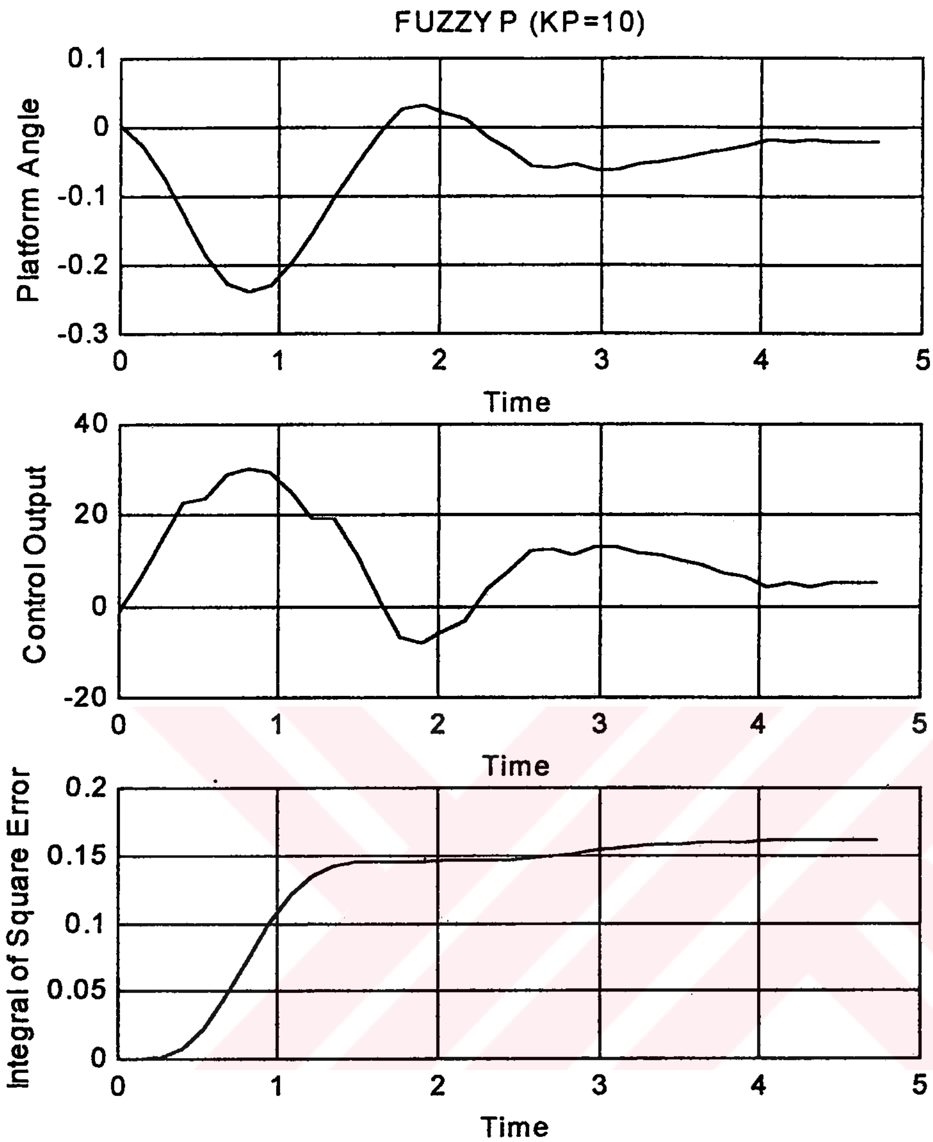


Figure 9. 5 Fuzzy P+D Controller Computer Program



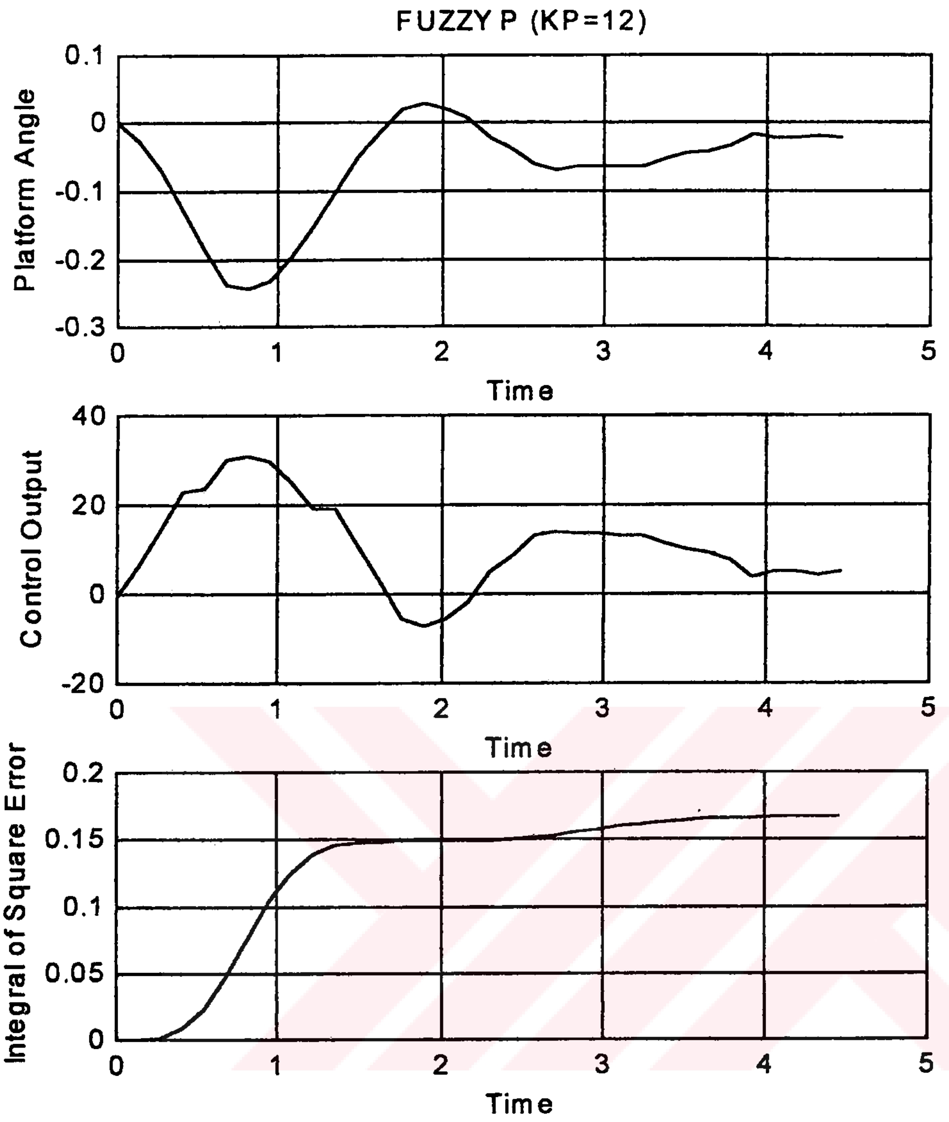
	Value	Time
Maximum	0.0655	2.0272
Minimum	-0.2365	0.9252
Steady State	0.0094	5.3469

Figure 9. 6 Response of the Fuzzy Controller (PE=12.5 PDE=100)



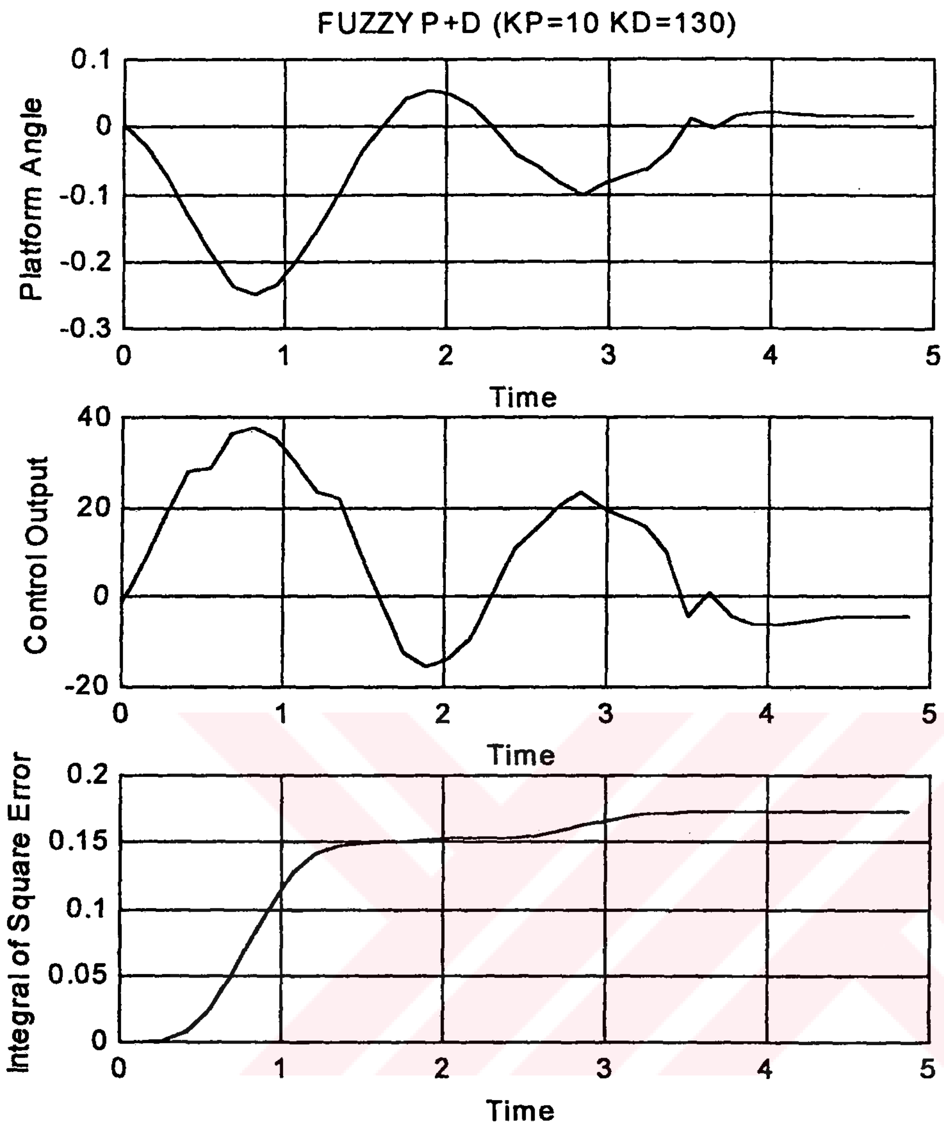
	Value	Time
Maximum	0.0270	1.8532
Minimum	-0.2387	0.8085
Steady State	-0.0197	4.0547

Figure 9. 7 Fuzzy P Control (P=10)



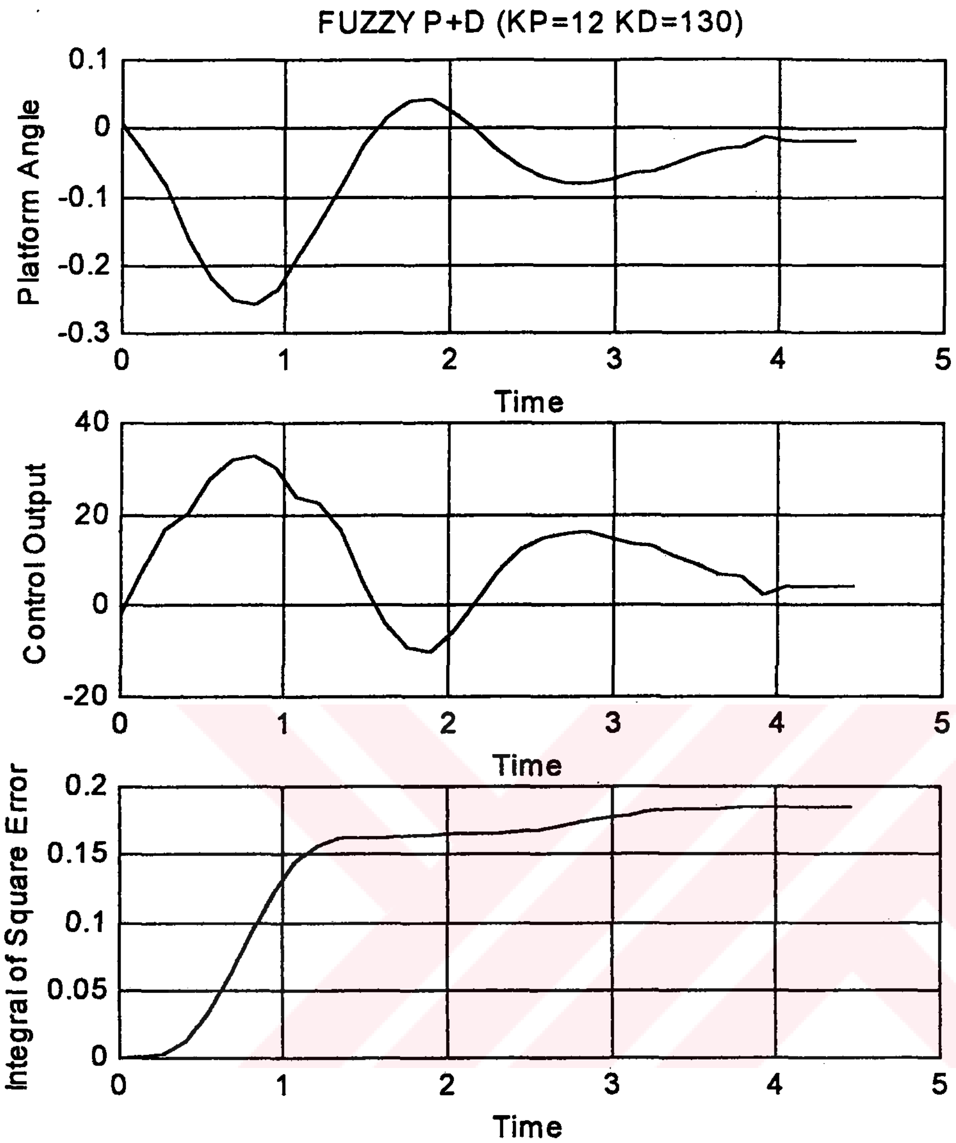
	Value	Time
Maximum	0.0245	1.8750
Minimum	-0.2441	0.8028
Steady State	0.0234	4.0564

Figure 9. 8 Fuzzy P Controller (P=12)



	Value	Time
Maximum	0.0497	1.8128
Minimum	-0.2469	0.7942
Steady State	-0.0133	4.0157

Figure 9. 9 Fuzzy P+D Controller (P=10, D=130)



	Value	Time
Maximum	0.0378	1.8005
Minimum	-0.2571	0.7786
Steady State	-0.0189	3.9659

Figure 9. 10 Fuzzy P+D Controller (P=12 D=130)

Maximum, minimum and steady state values and their occurring times are used to compare system response, and ISE is for performance evaluations of the controllers.

Basic fuzzy controller, which has error and delta error as input, (fuzzyPD) is the first applied algorithm The response of the system has characteristics similar to PD controller as expected. It has low maximum and minimum values and good steady state error. But controller is noise sensitive small changes in error signal causes high changes in control output.

Fuzzy P controllers with P=10 and P=12 also have good response curves.P=12 has lower steady state time and lower minimum time. But for small errors it generates higher control output which causes noise sensitivity and an increase in ISE. P=10 is slightly slower than P=12 but control signal for small errors are smoother and has better steady state value and ISE curve.

Fuzzy P+D controller is supposed to have better characteristics from fuzzyPD controller. Because of the nonlinearity and oscillations in system, to determine rules for delta error or any incremental input is hard and also imperceptible. To use fuzzy P and conventional D algorithm may be a good alternative to PID control. System response of the Fuzzy P+D controller is faster than both fuzzy P and fuzzyPD controller and noise sensitivity is higher than fuzzy P controller but lower then fuzzyPD. Maximum and minimum values are bigger than fuzzy P controller. When ISE curves of the controllers is compared fuzzyP P=10 controller has the best characteristic.

Fuzzy controllers are also capable to cover system performance specifications. But it is relatively hard to design controller because of the logical inference mechanism.

CHAPTER TEN

CONCLUSIONS

10.1 Conclusions

The tuning PID control strategy has been presented for the position control of the rotating platform system. Simulation based design techniques have been addressed. Mathematical model for the system has been founded and the simulation results are used as initial conditions for conventional PID controller. Since there exists a dead zone position control system, it is difficult to achieve high precision using only computer simulation model. Further improvement on PID controller is done by hand tuning methods.

An evaluation of fuzzy logic techniques applied to the control of rotating platform was presented. Experimental results confirmed that the fuzzy logic approach is feasible and can be an interesting alternative to conventional control, even when the system model is known and linear or nonlinear. The implemented fuzzy logic controller presented a slightly superior dynamic performance when compared with a more conventional scheme, namely in terms of insensitivity to changes in model parameters and to speed noise. This can be an important requirement in speed/position control schemes using electrical machines, namely in robotics.

A fuzzy P control scheme that allows for the online replacement and subsequent improvement of existing conventional PID control performance has also been developed and the approach was demonstrated on a physical model and used to improve PID control performance. An explicit control formula and a similar structure to conventional PID will allow its use by personnel unfamiliar with fuzzy logic. Fuzzy gain scheduling will normally show minor set point overshoot since it typically involves a relative increase in control sensitivity near the set point.

The fuzzy P+D controller derived in this paper successfully demonstrated better performance than the conventional PD controller for these mechanical system, particularly for skipping nonlinear effects. The fuzzy P+D controller is also able to tolerate noise problem. Since nonlinear effects will be encountered in many complex systems, such as robotic manipulators, the ability of the fuzzy PID controller to tolerate these unmodeled nonlinear and gain-value variation factors is supposed to be an improvement over conventional linear PD controllers in real-world applications.

Though it is claimed that fuzzy logic controllers are easier to tune than conventional ones, and therefore the development times are shortened. From the experience of the author this statement cannot be supported, at least for this type of application and with the tuning process done manually. Progress in research on machine learning techniques may change this negative point.

From a general point of view fuzzy controllers have better system response characteristics then conventional controllers. In contrast optimization of the fuzzy controller parameters is harder, since it primarily based on specific knowledge of the system. For fuzzy P controllers this problem is ceased particularly. development of artificial intelligence tools and combinations of neural network structures to fuzzy controllers will improve the performance of such systems.

REFERENCES

Alvarez-Ramirez, J., Cervantes, I., Kelly, R. (2000). PID regulation of robot manipulators: stability and performance. Systems & Control Letters 41 73-83

Aström, K. J., Wittenmark, B. (1984) Computer Controlled Systems. New Jersey, Prentice-Hall Inc.

Blanchett, T.P., Kember, G.C., Dubay, R. (2000). PID gain scheduling using fuzzy logic. ISA Transactions 39 317-325

Carvajal, J. Chen G. Ogmen H. (2000). Fuzzy PID controller: Design, performance evaluation, and stability analysis. Information Sciences, 123, 249-270

Dorpf, R.(1992) Modern Control Systems. (6th ed.). İstanbul, Literatür Yayınları.

Emami, M. R., Goldenberg, A., Türkşen, B. (2000), Systematic design and analysis of fuzzy-logic control and application to robotics, Part II. Control. Robotics and Autonomous Systems 33 89–108

Fonseca, J. Afonso, J. L. Martins, J. S. Couto C. (1999). Fuzzy logic speed control of an induction motor. Microprocessors and Microsystems, 22,523–534

Fraser, C., Milne, J. (1994). Integrated Electric and Electronic Engineering for Mechanical Engineers. England, McGraw-Hill

Gorez, R. (1999) Globally stable PID-like control of mechanical systems. Systems & Control Letters, 38, 61-72

Kuo, B.(1987) Automatic Control Systems (5^{th} ed.). New Jersey, Prentice Hall

Li, W. Chang X. (2000). Application of hybrid fuzzy logic proportional plus conventional integral-derivative controller to combustion control of stoker-fired boilers. Fuzzy Sets and Systems, 111, 267-284

McNeil, F.M., Thro, E. (1994) Fuzzy Logic: A Practical Approach. London, Academic Press.

Raven, F. (1982) Automatic Control Engineeering. England, McGraw Hill Co.

Reznik, L., Ghanayem, O., Bourmistrov A. (2000). PID plus fuzzy controller structures as a design base for industrial applications. Engineering Applications of Artificial Intelligence, 19, 419-430

Uyar, E. (1998) Sistem Dinamiği ve Otomatik Kontrol. İzmir, DEÜ MF Yayınları

Woo, Z., Chung, H., Lin J. (2000). A PID type fuzzy controller with self-tuning scaling factors. Fuzzy Sets and Systems, 115, 321-326

Yi, J., Yubazaki, N. (2000). Stabilization fuzzy control of inverted pendulum systems. Artificial Intelligence in Engineering, 14, 153–163