THE DESIGN AND CONSTRUCTION OF
AN
AUTONOMOUS VEHICLE
WITH
ULTRASONIC SENSORS.

by

Özkal Özsoy

BS. in E.E., İstanbul Technical University, 1993

Submitted to the Institute for Graduate Studies in

Science and Engineering in partial fulfillment of

Master of Science

in

Electrical and Electronical Engineering

Boğaziçi University

1996

ACKNOWLEDGMENTS

I would like to thank Assoc. Professor, Yaðmur Denizhan and Professor Dr. Ömer Cerid for their supervision and guidance on this thesis work.

I would like to thank to my classmate Setrak Iþýkbay for all his effort and help throughout the realization of this work.

ABSTRACT

THE DESIGN AND CONSTRUCTION OF AN AUTONOMOUS VEHICLE WITH ULTRASONIC SENSORS.

The objective of this project is to implement the design and construction of a mobile robot which is suitable for testing any software written in assembly on ‘Collision avoidance’, ‘path planning’ and ‘motion planning’.

Robot navigation, ultrasonic sensing, ultrasonic ranging, methods for driving & controlling different types of DC and stepper motors are studied. And various hardware and software are implemented and tested for optimum performance. Although all of these methods, hardware and software aren’t used in my robot they’re presented in these thesis in their fully working & tested versions.

For navigation in an unknown environment, the information obtained from an virtual ultrasonic ring is used. That virtual ultrasonic ring is constructed by rotating a pair of ultrasonic sensors that are mounted on the robot by a stepper motor.

ÖZET

ULTRASONİK SES DUYUCULARI KULLANARAK KENDİ KENDİNE GİDEBİLEN ROBOT-TAŞITIN TASARIM VE GERCEKLEŞTİRİLMESİ.

Bu tezin amacı, hareketli robot taşıtlarda 'yön bulma' , 'engel sakınma', 'yol ve rota planlaması' üzerine hazırlanmış algoritma ve programların test edilebilmesi amacı ile kullanılabilecek bir robot taşıtın tasarımı ve gerçekleştirilmesidir.

Robotlarda yön bulma, ultrsonik mesafe ölçümü, robot hareketlendirici elemanların kullanılması, özellikleri, mikroişlemcilerle bağlaştırılma yöntemleri gibi konular üzerinde çalışımıştır. Bu konular üzerinde hazırlanmış yazılım ve donanım bu tez içerisinde anlatılmaktadır. Bunların hepsi tasarlanan robot taşıtta kullanılmak için düşünülmüş olmakla birlikte bazıları daha optimum çözümler bulunabilmiş olması nedeniyle kullanılmamıştır.

Bilinmeyen yabancı bir ortamda yön bulma işlemini gerçekleştirmek için bir adım motoru ile 360° döndürülerek meydana getirilen sanal bir ultrasonik duyucu halkası kullanılmıştır. Adı geçen motor ve duyucular robot üzerine monte edilmiş durumda çalışırlar.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 Overview

Obstacle avoidance is one of the key issues to successful applications of mobile robot systems. All mobile robots feature some kind of collision avoidance. Some primitive algorithms can only detect an obstacle and stop the robot short of it in order to avoid a collision, but sophisticated algorithms enable the robot with the detour of the objects. The latter algorithms not only detect an obstacle, but makes a kind of quantitative measurement concerning the dimensions of the obstacle. Once these have been determined, the obstacle avoidance algorithm steers the robot around the obstacle and the robot then, proceeds towards the original target.

There is a number of collision avoidance algorithms. Nearly all of them are simulated on computers. In these simulations, the robot is usually presented as a dot on the screen. The real mass, magnitude of the robot, and the other things that may effect the robot environmentally are usually neglected. Because of these neglected effects, any algorithm that is practically not true and useless may seem to work on a computer simulation. Without these environmental effects being taken into account, simulations usually lack complete reliability. For a real test of the algorithm, a real robotic hardware needs to be built. For building a reliable robot of any kind (Mobile robot, or a robot arm) a number of things needs to be thought.

1.2 The Scope And Objective

The aim of this thesis is to build a mobile robot, which had been planned to be like a model vehicle which can find its way towards a given target and avoid collisions with the objects on its way. The robot uses two DC motors for rotating two wheels on its sides. It has an additional free rotating wheel at the front.

It is planned to power the robot with model car batteries. The robot is equipped with sensors for measuring speed , distance to objects etc.. Additionally the robot's position according to its target will be found by using ultrasonic pulses which are transmitted by the target point.

1.3 Outline Of The Study

1.3.1 The Base Unit Of The Robot

We are free to build our robot to any size, form power that fits our needs and imagination (see Figure 1.1). And the robot can be designed to use almost any available motor or batteries.

However , that is not to say that the mechanical components can be chosen without care. The components we use will strongly affect the success of our project.

1.3.2 Motor Selection

The first step is to select the motors. Since the motors usually are the most expensive part of the mechanical system, they likely determine the configuration of the remainder of the system.

When choosing the motors, a key parameter will be the voltage rating. Motors rated at less than 12 V. DC are rare in the power range we will need. Even if such motors are located, they would severely tax the current-carrying capacity of our motor-controller card and hookup wires. Motors rated at greater than 48 V DC could be used, but the battery-pack and the motor control transistors must be selected to withstand that voltage level. The result is unnecessary expense and increase in weight.

Probably the best choice is a brush-type, permanent-magnet torque motor, rated at 12 to 36 volts DC. Those motors are commonly used in automotive applications as starter motors, wind-shield-wiper motors, or electric window actuators.

The power output of the motors is an other key specification. The output is expressed in either *horsepower* (hp) or watts, with one horsepower being equal to 746 watts. The motor's nameplate usually will indicate the power input, such as 12 volts at 10 amps, which translates to a power input of 120 watts. Assuming for the moment, a 100% conversion efficiency, that yields approximately a 1/6-hp power output (120/746) .

It should be noticed that as the motor voltage increases, the current decreases for a given power output. That factor will determine the gauge of the wires used to connect the motors to the controller. If more than 10 amps of current are drawn by the motors, large-gauge wire will be required.

1.3.3 Speed Calculations

After selecting the motors for our robot, we must calculate the rotational speed of the tires necessary to achieve a desired speed. In my thesis the robot is supposed to make several measurements of distance, and detect the goal around itself periodically. To achieve a better reliability in these measurements, the time between each measurement may need to be increased. Though my robot uses just one pair of ultrasonic sensors (one for transmitting, and an other for receiving signals), it uses a scanning technique to obtain data from several different positions in its environment. Because of the measurement delay times of this scanning process, in each scan of 360 degrees, the robot needs to stop for three seconds. After each stopping period, the wheel motors are energized according to the environment which the robot travels in and the distance between the robot and the target point. As a result, the robot travels with a full speed of approximately 0.7 meter/sec. when it is headed towards the target point and there is no obstacle in front of it. When it encounters with obstacles, the speed reduces to approximately half of full speed heavily depending on the shape and number of the obstacles.

1.3.4 Motive-Power Requirements

Our next task is to calculate the amount of power it will take to achieve the desired speed. Calculations of the motive power required to drive the robot are influenced by many factors, such as the final weight of the robot, additional payload, the type of surface the wheels are on, the state of the batteries, and the type of wheel we use. However , we can still roughly determine the necessary amount of power.

The torque required at the wheels is largely a function of the surface the robot is operating on. Obviously , a hardwood floor and a shug rug will require different amounts of torque. In order to get a rough idea of the force required to move the robot, it may be loaded with the same weight which it is supposed to carry in normal operation and then attached to a spring scale and pulled across the floor. Then the force which is read on the scale will be the approximate value necessary to move the robot unit. It must be remembered to make additional allowances for climbing grades, towing and rough terrain.

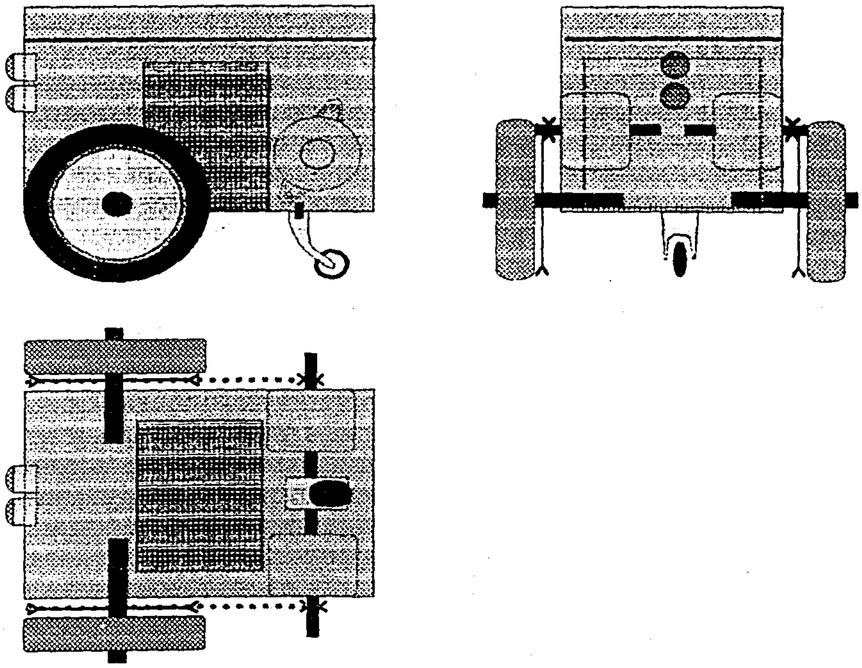


FIGURE 1.1. A sample robot drawing which has two DC motors on each side and a free running wheel at the back.

1.3.5 Torque Motors

The characteristic curves of a DC permanent magnet motor are derived from two basic phenomena:

- Back *emf* (*electromotive Force*) is proportional to motor speed.
- Output torque is proportional to current drain.

If we examine a motor's output speed versus its output torque at a given excitation voltage, the result is a straight line between the no-load speed and the stall torque (See figure 1.2). Basically an excitation voltage causes the motor to rotate, which produces back emf . The difference between the back emf and the excitation voltage causes current to flow in motor's armature, producing the torque that makes the motor to accelerate. The motor's speed increases until the difference between the excitation voltage and the back emf limits the current to an amount sufficient to meet the torque requirements at the shaft. Therefore with no load, the shaft speed is at maximum, and the torque load is only the motor's internal friction. At stall, the back emf is zero, and maximum current Is possible, producing maximum torque.

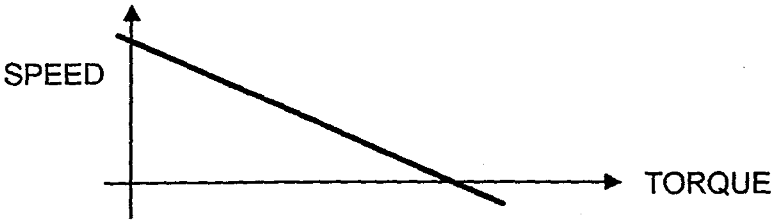


FIGURE 1.2. Characteristic curve for a DC brush type torque motor.

If the motor is delivering no torque (no load) or is not moving (stall), the motor's power output is zero. Maximum power is produced at half maximum speed and half maximum torque. Therefore if we superimposed the power output of our motor on the speed-torque curve, we would get a parabola with maximum power at half speed and half torque.

So, we want to determine whether our motors can deliver the power necessary to drive our robot at the desired speed. Our first task is to determine the stall-torque output of the motor by direct measurement. Most DC motors operate at between 1500 and 2000 rpm, so we have all the information necessary to estimate the power output of our motor.

For example, if a DC motor produces 3 N-m of stall torque and operated at 60 rev/sec with no load. The maximum power point (in this case 45 N-m) is found by multiplying each value by one half. To find the power point in horsepower, the value in N-m is divided by a constant value.

1.3.6 Drive Train

If the wheel is rotating at 2 rev./sec., the motor must be operating at a much higher rate. If we assume a speed of 16 rpm, the motor may be operating at 1000 rpm. Therefore , a reduction mechanism must be used to convert the high-speed, low-torque output of the motor to the low-speed, high-torque requirement of the wheel. The reduction ratio required is equal to the speed of the motor to the speed of the wheel. Which is 16:2 in the above example (or 8:1). There are DC brush motors in the market which are sold with internal gears which may be very suitable for this kind of robotic applications. In my project, we also used this kind of motors which are specially designed to be used in servo controlled voltage regulators. And later I've changed the motors with cheaper motors with gears extracted from used Epson printers. The stepper motor which is used for rotating the "head" is extracted from an old hard disk. I've realized that the more time you spend in the junk-yard, the more money you save.

1.3.7 Battery Selection

As the batteries must power the robot at all times, except during recharging, their selection is an important part of the design process. Two factors must be considered when selecting the batteries: the amount of power that must be supplied, and the length of time that that power must be supplied.

It is important to remember that the robot has two basic modes of operation: moving and non-moving. The change in the power consumption in these two modes must be considered.

1.3.8 Building The Robot

After selecting the batteries and motors, the actual building process seems less complicated. Basically, the process involves creating and then assembling the chassis, and mounting the components on it. Although the operation is quite simple, it requires great attention to innumerable details.

2. DESIGN AND CONSTRUCTION PROCESS

2.1 Robotic Constructions

A robot is a complex system not unlike an automatic instrumentation system. It is a combination of a number of basic units, i.e. the data processor, mechanical superstructure, electronic signal processors, transducers and so on, the aim is to produce a machine that is general purpose in nature and capable of performing physical tasks in the same way that computers perform mental tasks. Since one can not perform any physical task without some form of mental effort, it follows that any good robot will need to incorporate a computing ability. A block diagram of a physical robot is shown in figure 2.1. Of course , not all robots will take this form-not all will possess a limb.

2.1.1 Mechanical Superstructure

This is required to support all of the internal components, such as motors, batteries, electronic circuit boards, and so on. It is responsible for holding the robot together. It also serves the often ignored function as a 'bed' or base for mounting the sensors and , as such, the geometry of the mechanical superstructure is very important, since the software used in the data processor will also depend on this. Great attention should be paid to the design of the mechanics since unlike an electronic circuit, modifications are not so easily carried out.

I have found during my involvement with robots that, the guiding principle in such a project is efficiency. Because the robot will have to depend its own power supply, it is logical to optimize things so that it will obtain the maximum operating time from its batteries and the factors affecting this parameter are mostly mechanical in nature. For example, the robot's center of gravity should be designed to lie directly over the driving wheel axis, since wheel slip will thus be reduced.

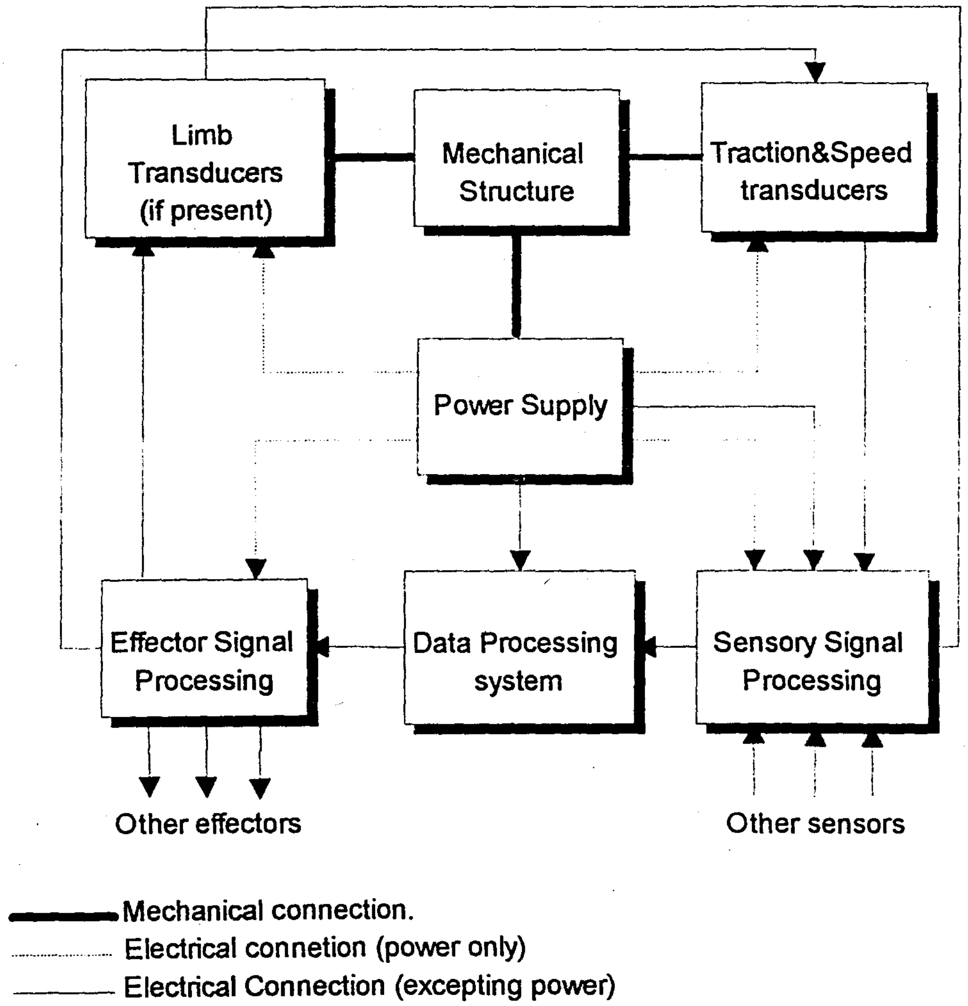


FIGURE 2.1. (a) Block diagram of a typical robot

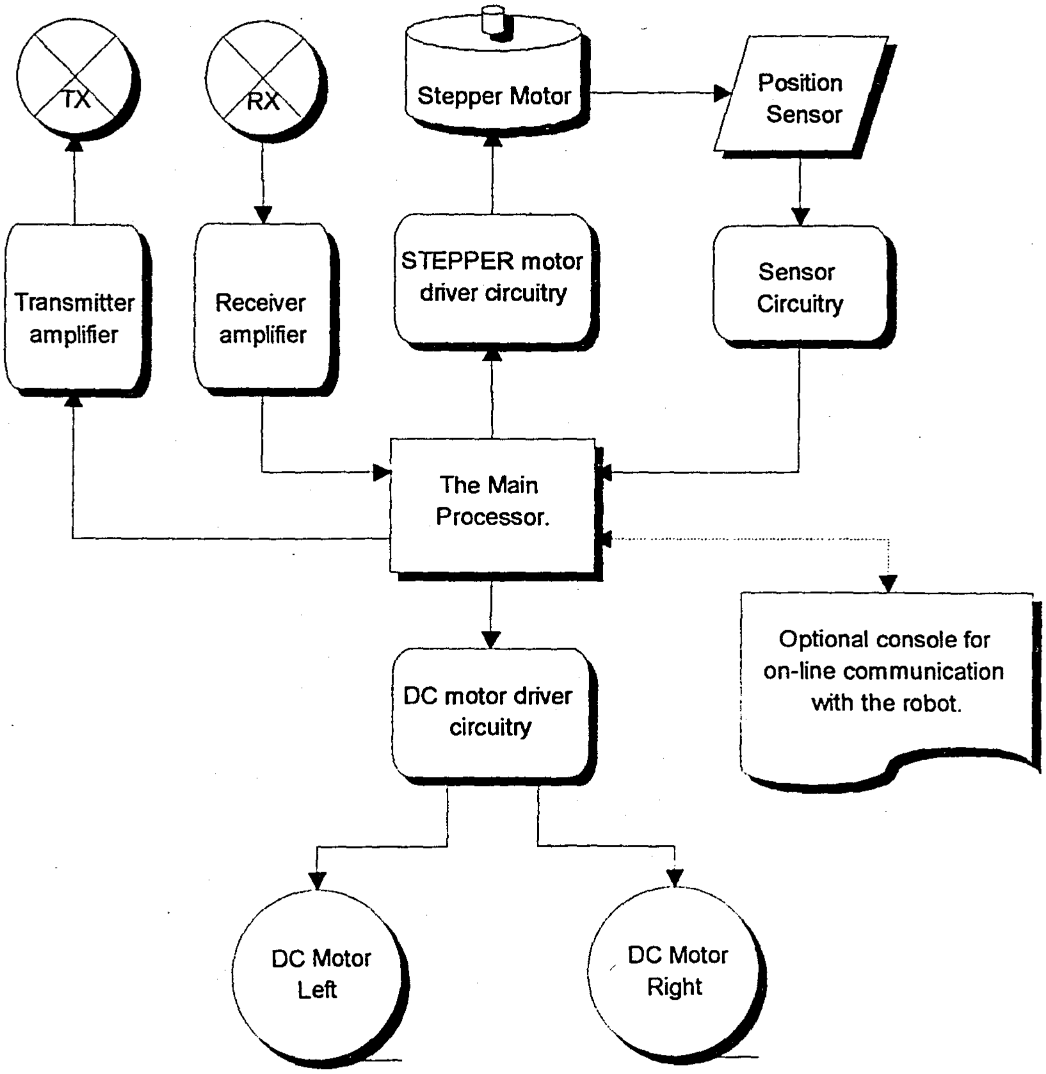


FIGURE 2.1. (b) Block diagram of the robot presented in this thesis

2.1.2 Power Supply

An ideal robot would contain two supplies. One for the electronic units and an other for the electro-mechanical units. This is because the electromechanical parts apply a greater load to the power supply and the voltage drop introduced into the electromechanical supply could drastically affect the performance of the circuitry. And the effects of noise caused by the electromechanical parts may even cause malfunction in the microprocessor based parts. These malfunctions are usually unrecoverable and may put the processors of the robot in endless-loop type unwanted conditions.

The type of supply used will be some form of battery, although in a number of cases an umbilical cable could be used. Lead-acid batteries are favored, due to their good power-to-weight ratio. Although ni-cad cells can be employed for units which require less power since these units require less maintenance.

In non-ideal it is very common to use different power sources. In applications which several sensors which need higher voltages are used, using more power supplies might be necessary.

In my robotic design, there are more than two power sources. One for the microprocessors and the interface circuitry. One for the electromechanical parts namely the two DC motors and the step motor. A third source is used to power the ultrasonic transmitter which needs to be supplied with greater voltage to operate within greater distances. In my robot, the DC motors are supplied with 12 volts, the step motor with 6 volts and the sensors with 18 volts. Two small 9V batteries are used to power the transmitter. Though the power need of the transmitter is less, these cells are quite adequate for an operation time of several hours continuously.

2.1.3 Sensory Signal Processing

This is one of the most interesting aspects of robot design: its purpose is to provide the conditioning of the signals from the multitude of transducers on the robot, i.e. analog-to-digital conversion(ADC). It will also consist of a switching pack , to enable the data processor to access any one of the sensors. Many types of sensors can be added to a robot, e.g. ultrasonic ranging, temperature sensing, and force sensing, each supplying data to the data processor.

For my robot, there are several sensory signal processors which are used to amplify and shape the signals which come from the ultrasonic receiver, to amplify the signals to be transmitted through the ultrasonic transmitter. Without these circuitry, the interfacing the microcontrollers to the high-power motor drivers, and also to the low power- noise sensitive ultrasonics may be very problematic.

2.1.4 Effector Signal Processing

The function of this particular unit is to convert the digital data from the data processor into signals capable of operating the robot's effectors, i.e. its DC motors and stepper motor for turning the sensors of the ultrasonic ranging system.

One must be aware that the robot must process more sensors than the effectors since that is an unwritten law that much more data goes into a system than comes out of it. Effector signal processing covers such things as digital analog conversion and motor speed control.

The speed control is basically accomplished by applying pulse-width modulation to the driver circuitry when it is thought to be necessary to go slower. The pulse width modulation, according to the duty cycle of it changes the rms voltage applied to the motors. Thus reducing the speed of the motors without reducing he torque of them. Also a special driver IC is used as a H bridge for driving both the DC and Step motors. This IC is very useful in controlling DC and step motors. It makes direct controlling with microcontrollers possible and requires a few external components.

2.1.5 Data Processing

A better title for this device would probably be 'data converter', since that is its primary function in a robot. The object is that for every type of data entering the system, there should be corresponding data coming out. The relationship between data in and data out is complex, and is wholly determined by the software contained in the data processor. It's a good idea to develop the software for your robot simultaneously with the hardware, since the deficiencies in one aspect can be taken up in the other.

2.1.6 Robot Sensors

A robot needs to have a lot of information entering it per unit time in order to make useful deductions about its environment. It is for this reason that It's been decided to make an accurate ultrasonic ranging system.

2.1.7 Motor Controls

A major problem is a robot's inability to move in straight lines. This is brought about by the robot's drive wheels rotating at different speeds due to things like differences in instantaneous wheel load and friction. The combination of all these apparently minor effects, is to cause the wheels to rotate at unrelated speeds, a feature known to be highly undesirably.

There are two approaches for solving this problem; software intervention or use of a hardware lock circuit. Both of these ideas are worked on paper and the software intervention method is preferred, because it is more accurate and uses a processor rather than the hardware.

A block diagram of the computer intervention method of wheel control is shown in figure 2.2. It is a kind of digital control where the speed of the motor is measured by the voltage produced by the motor itself. While measuring the speed of the motor from its voltage generating facility, the rate of pulse width modulation which is necessary to be applied to achieve the desired speed is calculated by the processor. Then this pulse width modulation is applied to the motors by using special driver circuitry.

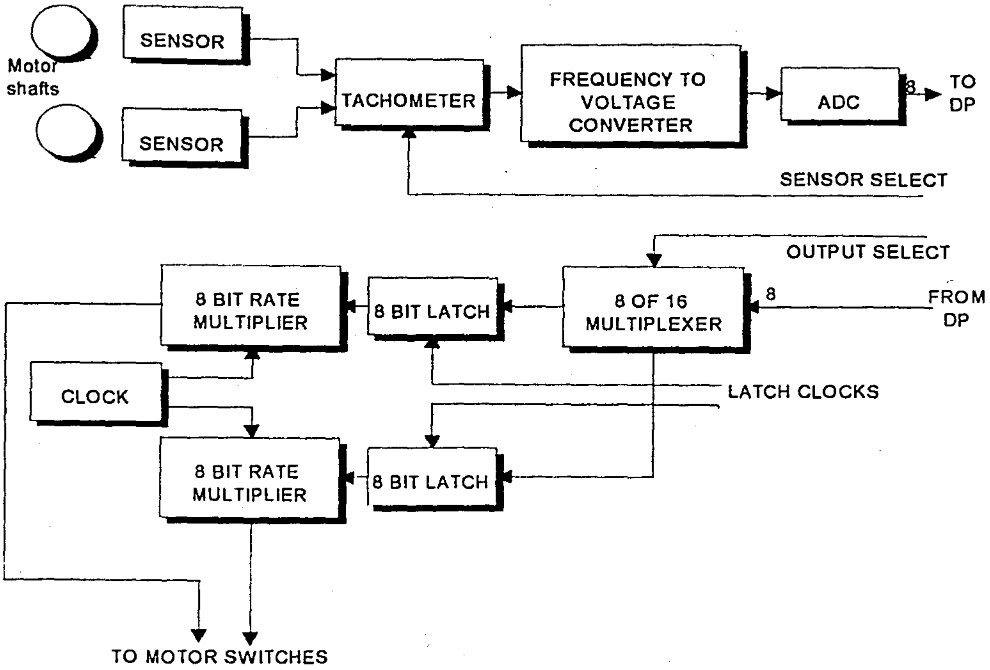


FIGURE 2.2. Block diagram of a sample motor regulation system

2.1.8 Mechanical Working

The robot was first thought to be like the one in the figure below, but after working on the shape of the robot a time, the shape changed to the one in figure 2.7.

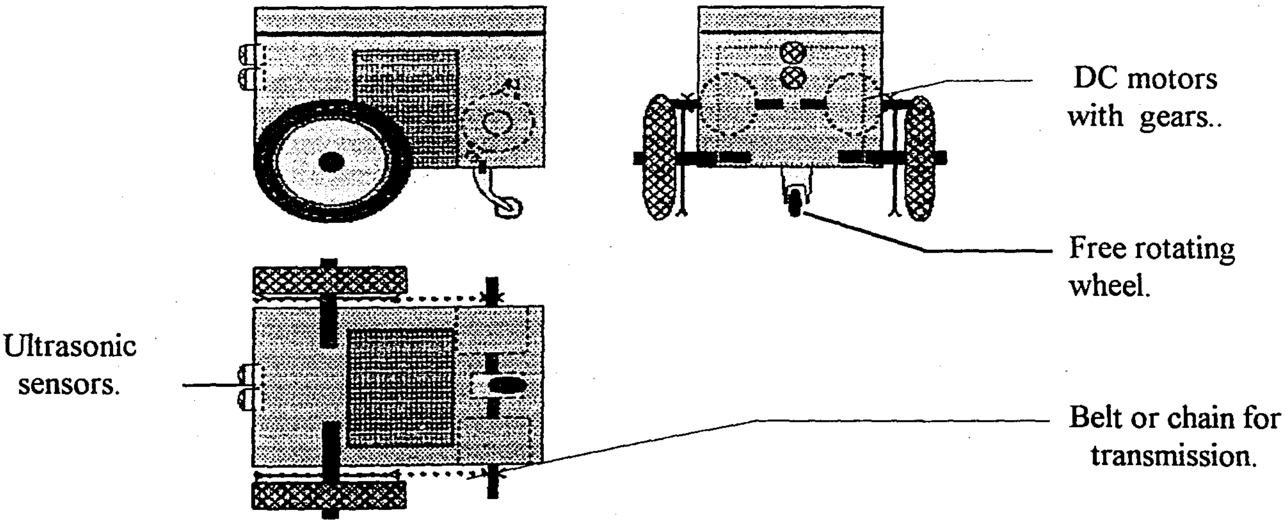


FIGURE 2.3. A sample construction for a mobile robot

Again after working for a time on this shape, the materials, difficulties of designing, producing and working on it, the ideas changed again in to the

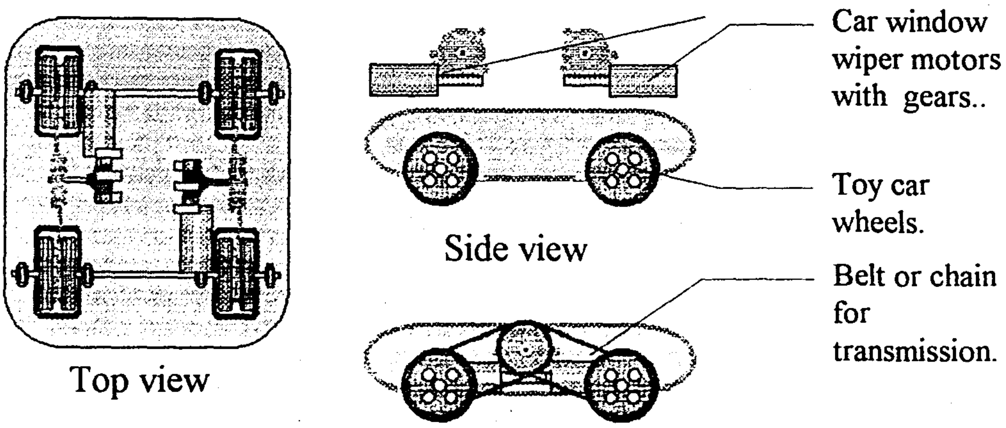


FIGURE 2.4. A sample construction for a mobile robot

shape in the following figures. Namely the shape of a tank. After studying the materials and production methods of such a robot, I've started the construction. The chassis is produced of 3 mm sheet of aluminum and welded at the edges in to the given shape. After the welding is completed the wheels and axes are bought and put together. A rubber belt with 95 cm. circumference is used for the construction of the caterpillar wheels. The power generated by the DC motors was given to the caterpillars by the wheels at the top on both sides. But lots of problems are encountered on power transmission, friction to ground on turnings and other friction between metal parts of the system.

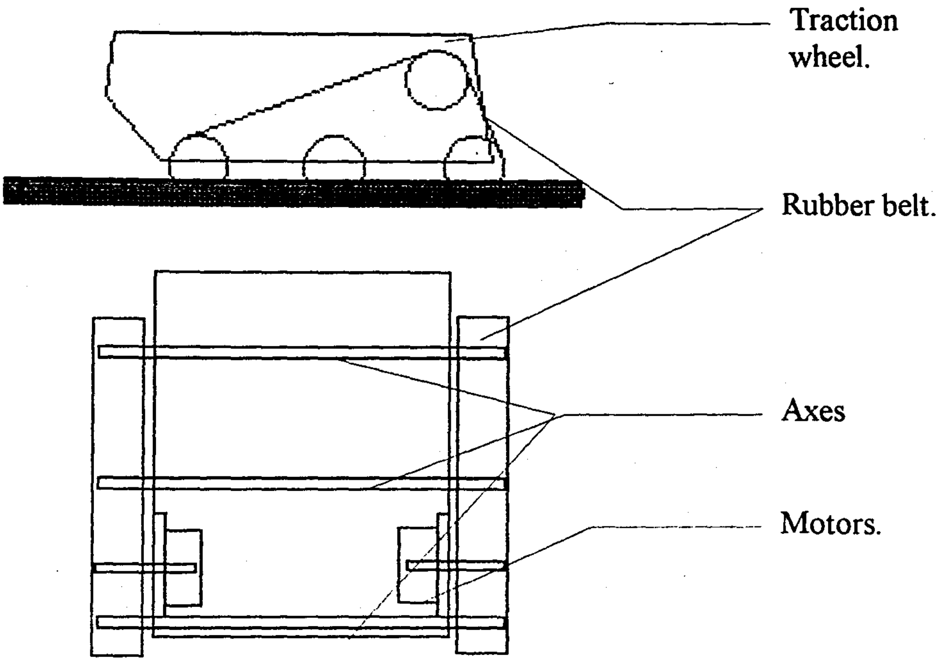


FIGURE 2.5. A sample construction for a mobile robot. This was also the construction which has been attempted to manufacture first

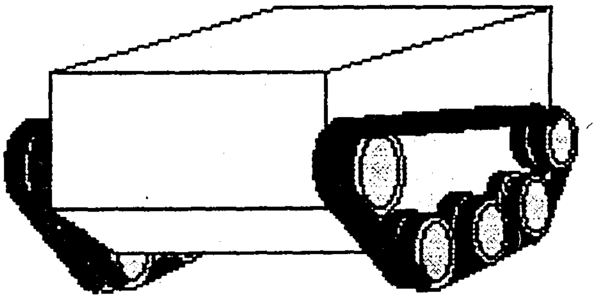


FIGURE 2.6. A sample construction for a mobile robot

Due to the lots of difficulties encountered in the design of the robot shapes in the above figures, I've started to study on the shape of the robot deeper. After a time of research on the toy robots that had been built to date I've found a better and easier shape to construct. And because of the materials used in this design, the robot was quite light and I've seen that building the robot as light as possible is very important though the heavier the robot is, the more power would be necessary to move it. After the study and research on shape of the robot, I ended up with the figure below (figure 2.7). And till the end I 've used the same form.

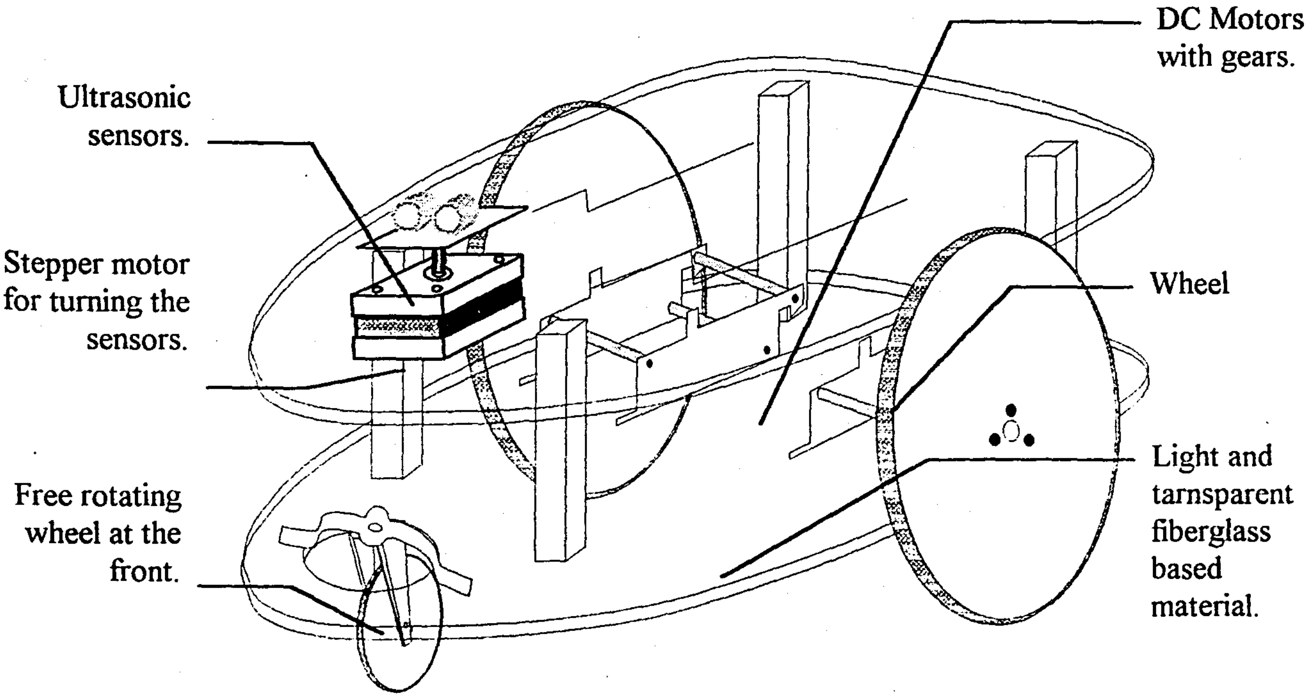


FIGURE 2.7. Final shape of our mobile robot construction ready for the electronical circuitry montage

3. THE MAIN PARTS OF THE ROBOT

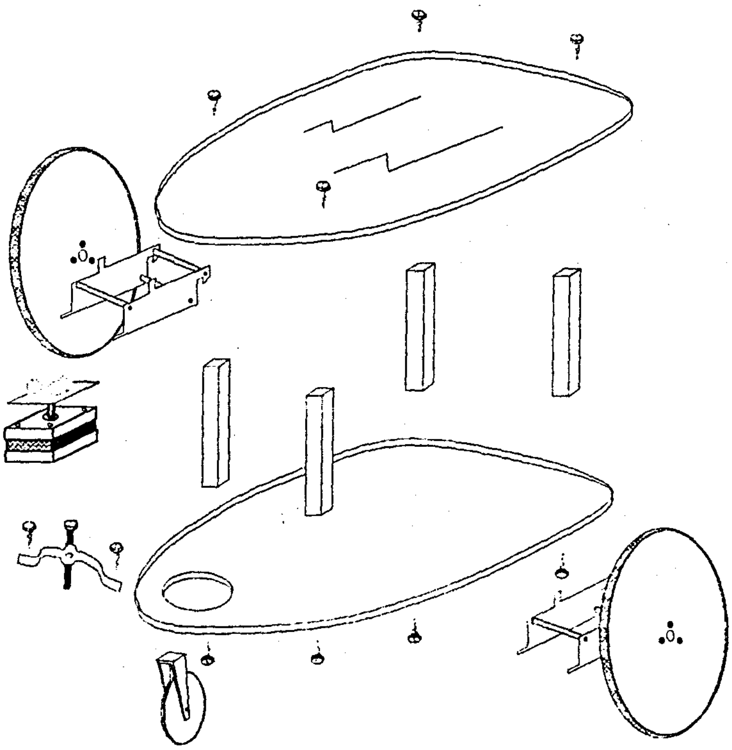


FIGURE 3.1. An expanded view of the mechanics of the robot

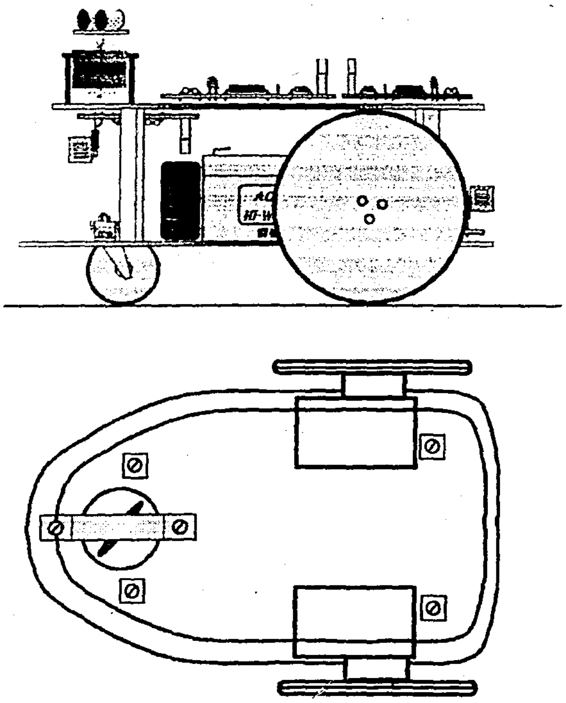


FIGURE 3.2. Technical drawings of the robot

3.1 RANGING AND OBSTACLE DETECTING BY ULTRASONIC SOUND WAVES

After studying the theory of the working principles of such a system, I've started working on the design of a circuit which , in conjunction with the data processing system, would enable the robot to determine the range of obstacles in its path.

After a lot of experimentation and measurements on the abilities and characteristics of ultrasonic sensors, a working system was put together and with it the robot could measure ranges up to a distance of 102 cms., to an accuracy of 1 cm. And also I've designed the circuitry so that it is possible to extend the measuring distance by only changing the voltage applied to the ultrasonic sensor's driver circuitry. Now lets take a closer look first at a general interrupt based ultrasonic ranging system then the system developed for my mobile robot.

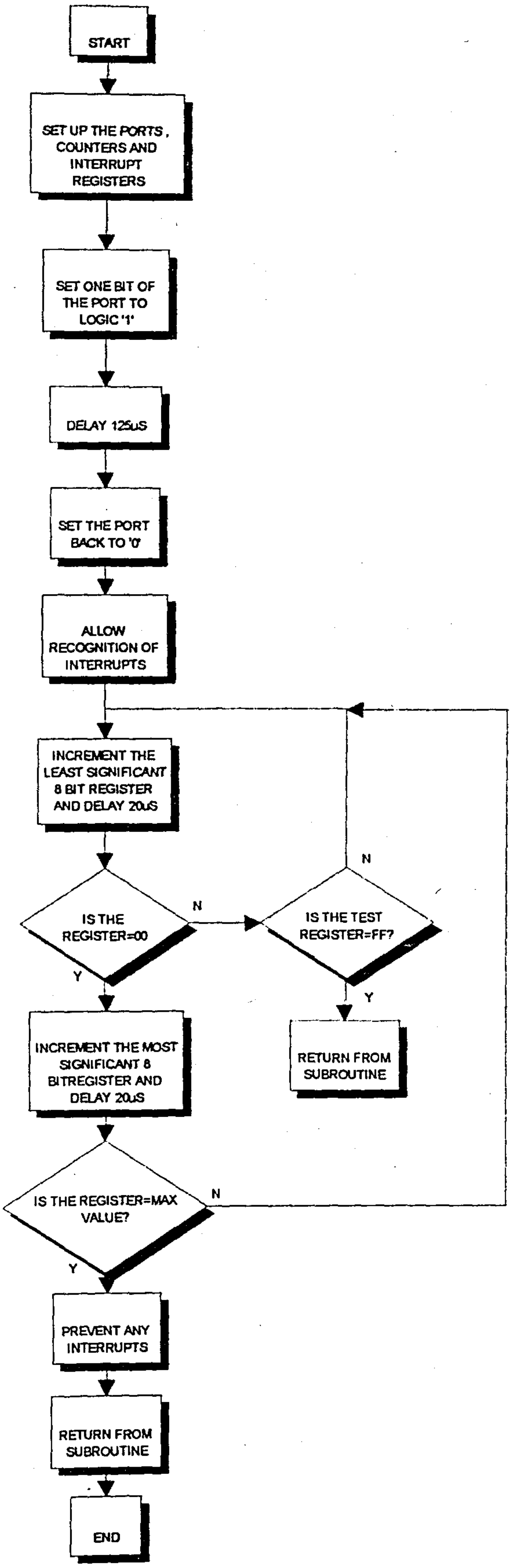


FIGURE 3.3. The flowchart of the software of a general ultrasonic ranging system by using interrupts

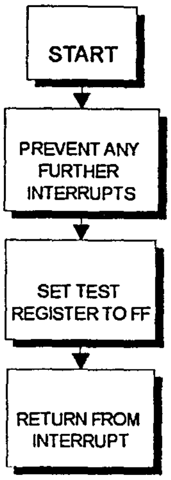


FIGURE 3.4. Flow of the Interrupt service routine of the chart in fig 3.3.

The principle is described in figure 3.5. As can be seen, the ultrasonic signal is produced directly by the microcontroller, and the received ultrasonic input is applied to the non maskable interrupt input (NMI) of the controller. The high gain ultrasonic receiver is a combination of transistors with a special Op-Amp designed to amplify and change the wave shape of the sound into a square wave to make it more suitable for using with the controller's interrupt capability.

The whole assembly is controlled by the data processor which also runs the main algorithm for collision detection and avoidance. Though ultrasonic ranging needs more processor speed, a special controller is dedicated for that job. An SGS Thompson ST6225 microcontroller runs on 8 MHz to accomplish ultrasonic wave generation and calculating the distance by the time elapses till an echo is received. This controller also looks for the ultrasonic pulses generated by the goal. If there is a goal signal coming to the receiver of the controller, it informs the main processor about the goal.

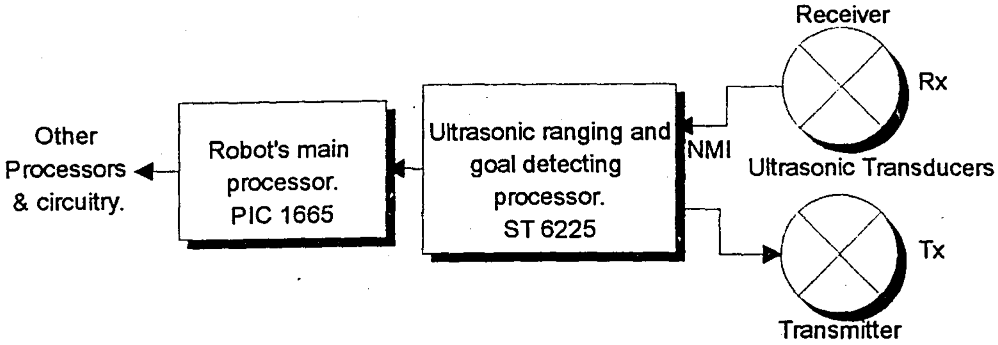


FIGURE 3.5. Block Diagram of the ultrasonic ranging and goal detecting system used in the robot

3.1.1 Interrupt Control

The data processor only activates the IRQ during the 'listening' period immediately switching off the transmitter. Anyway though the ultrasonic sounding and listening jobs are performed sequentially, there is a small amount of time delay between them. And this delay may introduce some range measurement failures while measuring very small distances. But since the distances to be measured in this project are not less than several centimeters, this is thought not to be a point of concern. The ultrasonic ranging and goal detecting processor produces a pulse train which consists of 5 pulses in the ultrasonic frequency range. These pulses in 40 kHz are transmitted through the TX transducer. Then the transmitter is switched off and the processor starts the internal timer and checks the incoming signals from the NMI input. The sound waves travel towards anything that may be in the robot's traveling path, and if there is anything sufficiently close then the reflected sound energy will interrupt the processor. When the processor gets the 5 of the transmitted sound waves, it informs the main processor of the distance between the robot and the obstacle. It gives a data to the main processor to be stored in a RAM location for further calculations. The internal timer is cleared periodically to disable incoming noise to effect the measurements. Care should be taken for the transducers not to be affected from each other. The two transducers must be a suitable pair though the sensitivity and efficiency of the of the sensors vary a lot with frequency and to eliminate unwanted effects they are sold as pairs. Usually a pair is aligned to operate best with each other. So even same sensors sold as different pairs may not function properly together. And the operating frequency is very important. Although some sensors said to work on 40kHz. , they may work better in different frequencies like 38kHz-42kHz. And the difference in efficiency may be quite big to be omitted. The optimum operating frequency must be measured and the sensors must be operating frequency be changed accordingly.

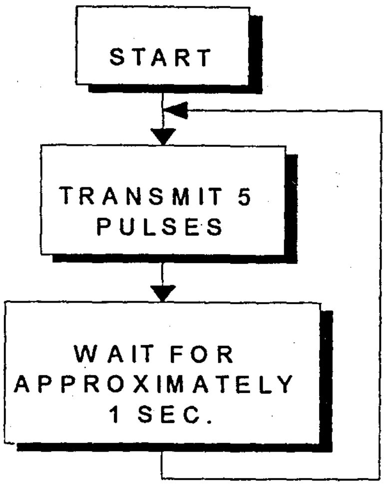


FIGURE 3.6. (a) The algorithm for generating the pulses from the goal

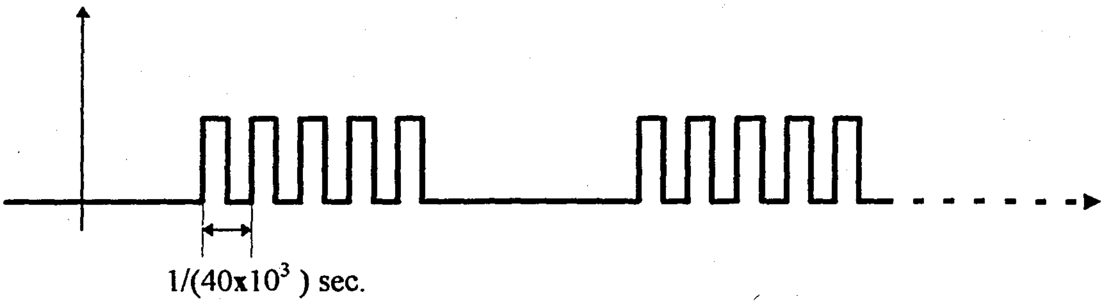


FIGURE 3.6. (b) The signals which are generated and transmitted by the goal

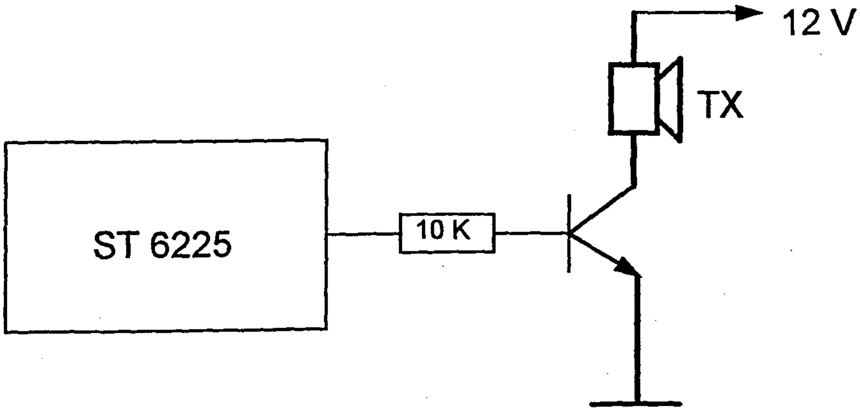


FIGURE 3.7. The goal's microprocessor based ultrasonic sound transmitter system.

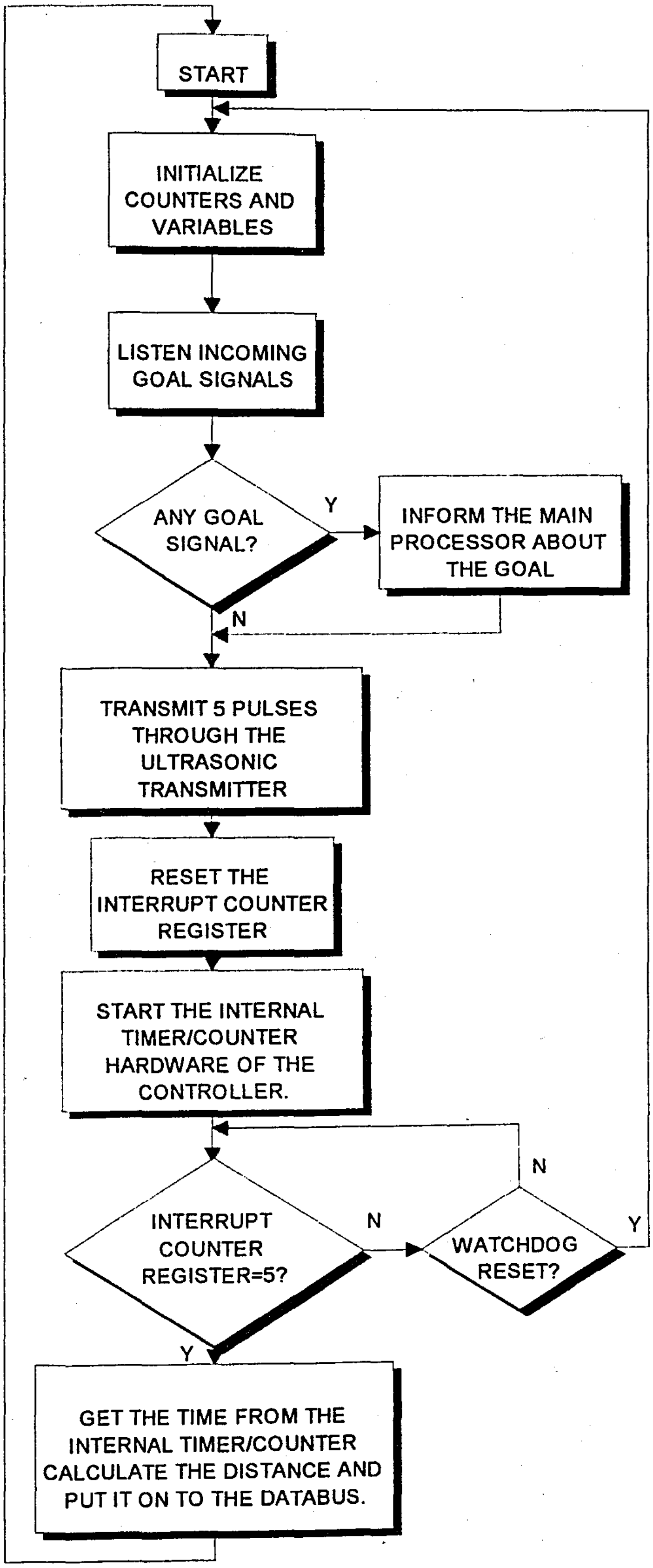


FIGURE 3.8. The ultrasonic ranging and & goal detecting algorithm used in the robot

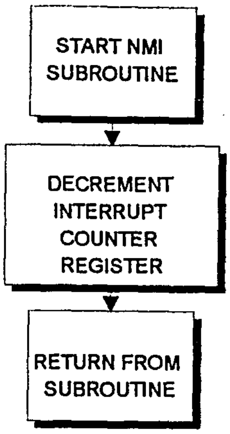


FIGURE 3.9. Non maskable Interrupt service routine, which counts the incoming ultrasonic pulses

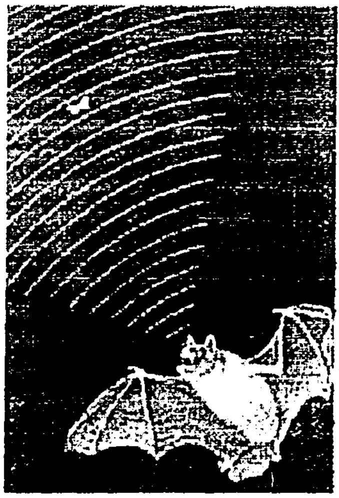


FIGURE 3.10. Bats also use a kind of very complex and reliable ultrasonic ranging system

The ultrasonic ranging processor has the ability to transmit pulses that are in the ultrasonic frequency range and then listen to incoming pulses from its environment. At the beginning, after the first power on it starts with initializing the internal timer & counters. Then it listens the environment for any goal signal . The processor knows the duty cycle and frequency of the goal signals. So it listens just for adequate amount of time. If any goal signal is present, it activates the GOAL (Active zero) output. After receiving the goal signal it waits for the amount of time necessary for the goal signal to die down then sends its own ultrasonic pulses. Just after finishing sending them, the processor starts a timer for measuring the delay between sending and receiving the ultrasonic pulses. From this delay the processor calculates the distance and gives the distance data to the main processor system via the databus. The time which the processor waits for its own pulses to come back can not be longer than the period of the goal pulse trains. Otherwise the two ultrasonic signals interfere with each other and cause wrong measurements.

3.1.2 The Necessity Of Multiple Sensors And The Construction Of The Virtual Sensor Ring By Using A Single Sensor Pair

The robot is designed to operate in a totally unknown environment where there may be unlimited amount of obstacles around it. Due to the shapes of the obstacles, the reflecting signals may come from a lot of directions. The robot must be able to detect all the of the ultrasonic signals which are reflected by the obstacles around it. To achieve this the robot needs to have a number of sensors forwarded to as many directions as possible. The more the number of sensors, the more the amount of data the processor obtains.

To use more sensors is at first quite expensive. Also as the number of sensors increase, data retrieving problems come into account. Also due to the different characteristics of every sensor on the market, the measuring system consisting of multiple sensors might be unreliable. To overcome this problem I've decided to use just one pair of sensors consisting of a transmitter and a receiver, which are specially aligned for each other. To obtain the multiple sensor data, we just rotated the sensor pair in 360 degrees. After working on servo controlled DC motors and step motors for a time, I've decided to use the second one to rotate the sensors around. In fact the both methods are quite reliable but having their own problems. In a servo system for example, the movement of the motor can be controlled with quite high precision. But the settling time of the needed position might be longer. To achieve lesser settling times, special control algorithms need to be used. PID is quite suitable for this purpose. But of course this increases the design time and effort needed to be spent on the project.

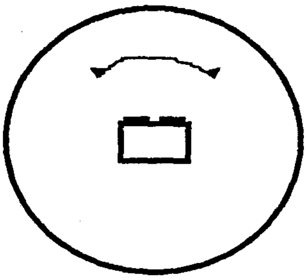


FIGURE 3.11. Sensors rotating with steps of 3.6° to form a "virtual sensor ring"

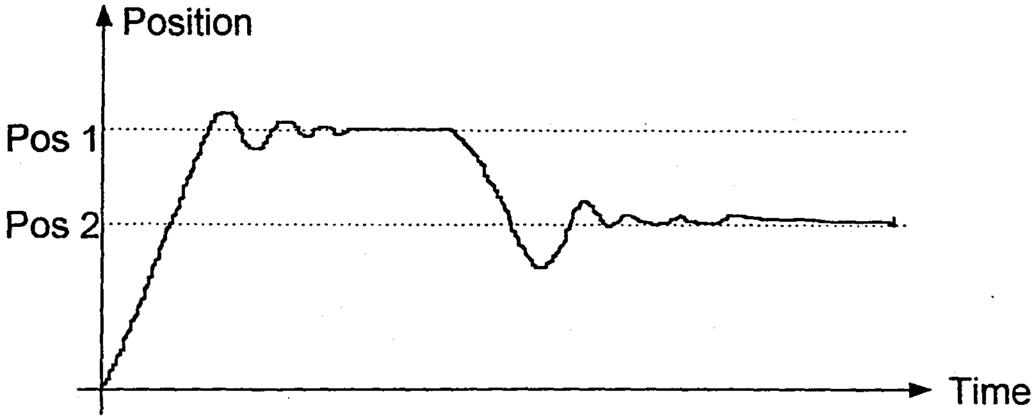


FIGURE 3.12. The approximate settling characteristic of a DC servo system

But in a step motor the settling times are acceptably lower than a DC servo system. The problem of step motors is less precision and higher power requirement.

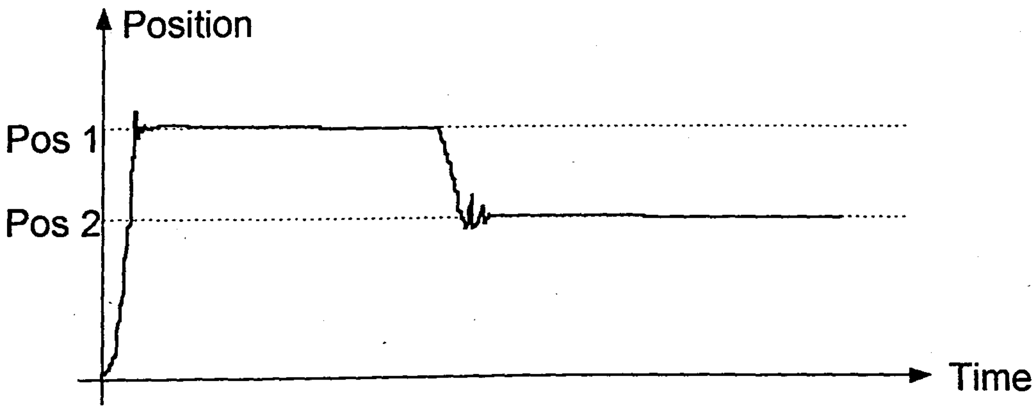


FIGURE 3.13. The approximate settling characteristic of a Stepper motor system

In my project, the preciseness of a step motor was quite adequate, I've worked on the special driving techniques of stepper motors and their interfacing to microcontrollers.

Now lets take a closer look at first to DC motor servo control, DC motors, their interfacing to microcontrollers. And the system, namely the hardware and software we designed for the DC servo system for rotating the 'sensor head' of any mobile robot.

3.1.3 Mechanization Of Control Algorithms For Dc Motor Systems, Using Microprocessors

In recent years, the VLSI technology has witnessed a revolutionary development, and allowed substantial reductions in the size and cost of digital logic circuitry. Computer system building blocks have progressed from the level of discrete components to the level of complex ICs involving many logic circuits in a single chip. It's even possible to find a great number of single chip computers nowadays. The invention and wide application of microprocessors have changed the philosophy of instrumentation, signal processing and control engineering fields. The microprocessor based systems have replaced the conventional ones based on standard analog and digital computing equipment. The application of microprocessors is limited more by the imagination of their users than their technology. As more and more system designers are familiarized with the capabilities of the microprocessors, the number of new microprocessor based applications increased very rapidly.

Vast amount of literature is now available covering important aspects and applications of microprocessors in different areas of control. Obviously the subject is too wide and deep to be covered in length of this thesis. Only some important aspects of the subject will be covered here.

When microprocessors are used for implementation of digital controls, several practical problems involving hardware and software arise. Some of these problems encountered during the realization of my robot and their possible solutions will be discussed thorough this thesis.

3.1.3.1 General Description Of Microcontrollers

A microcontroller is an I/O machine which is easily programmed and includes A/D and D/A converters. Some of the channels are digital and others can be analog. The control algorithm is built by hardware, software or both inside the microcontroller.

Until a generally acceptable μ controller is commercially available, the control engineer has to design his own μ controller by connecting necessary components such as μP, clock, memories, DACs, ADCs and other LSI devices which are usually connected together via a system bus. The function of a μcontroller is to accept analog and logic inputs, so process this information and to provide analog and logic commands as output. Sometimes it may be necessary to reinforce the computing capacity with special arithmetic devices such as fast multipliers or fast dividers. Or several microcontrollers sharing different tasks of the system may be implemented. This is the case in my robot.

A general block diagram of a digital control system is shown in figure 3.14 below;

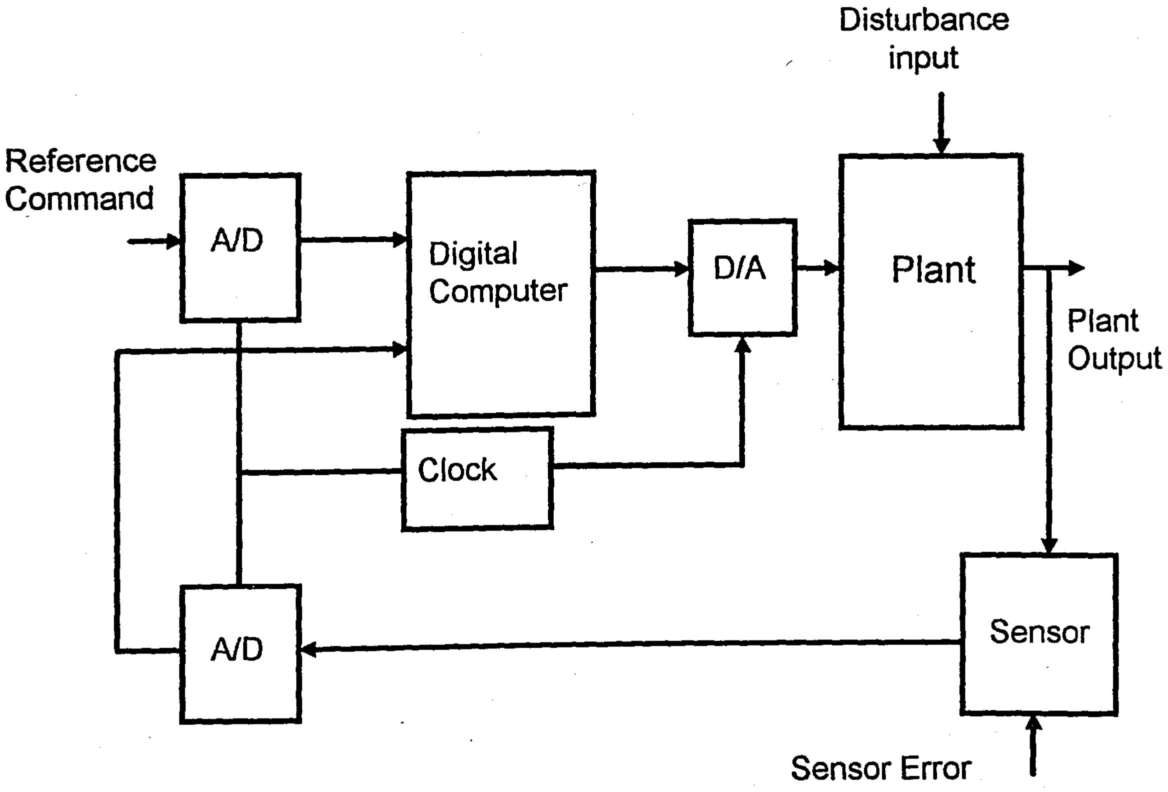


FIGURE 3.14. The general block diagram of a digital control system

If we redraw this block diagram with more details of the implementation of digital controller. In this microprocessor based control system, the μcontroller consists of the μP which is the central processing unit (CPU), a read only memory (ROM), a random access memory(RAM), two input ports and one output port. Additional subsystems needed in μcontroller are A/D and D/A converters, a sample and hold (S/H) unit and a timer, particularly geared to the sampling rate.

The ROM, RAM and I/O ports are connected to the CPU through data/address/control bus. The ROM usually contains the fixed part of the controller program while the RAM contains variable data. In the more recent systems an EEPROM memory is usually added for data which is only changed by the user. This data may be the reference speed , or temperature (or other value depending on the type system). In figure 3.15 , we have a single output port to transmit the calculated control signal and two input ports for feedback signal and command input.

The command input (representing the desired value of the controlled variable) and measured feedback signal are sampled at a sampling frequency and held at a constant value for the duration of the sampling period by the S/H device. Subsequently these signals are converted into binary coded numbers by A/D converters and fed to CPU through input port 1 and 2 respectively. The control signals calculated by the CPU are put out through the output port of the D/A converter which converts them to analog signal (voltage level).

The timer circuit controlled by the CPU is designed to fit the sampling period of the system. The timer determines the start of he next sampling period by transmitting a pulse every T seconds. This pulse is used as follows.

i) It performs an interrupt to the CPU. The currently running routine is stopped. A timer interrupt service routine is started which transmits the calculated control signal, through the output port, to the D/A converter.

ii) It is simultaneously transmitted to S/H devices causing the current input and measured signals to be sampled and held for one sampling period T. The value of the signals is then converted to binary codes by A/D converters. As soon as the A/D converters complete the conversions , they send control signals to the CPU which reads the converted signals through input ports 1 and 2. Within the same sampling period, the next control output is calculated by the CPU.

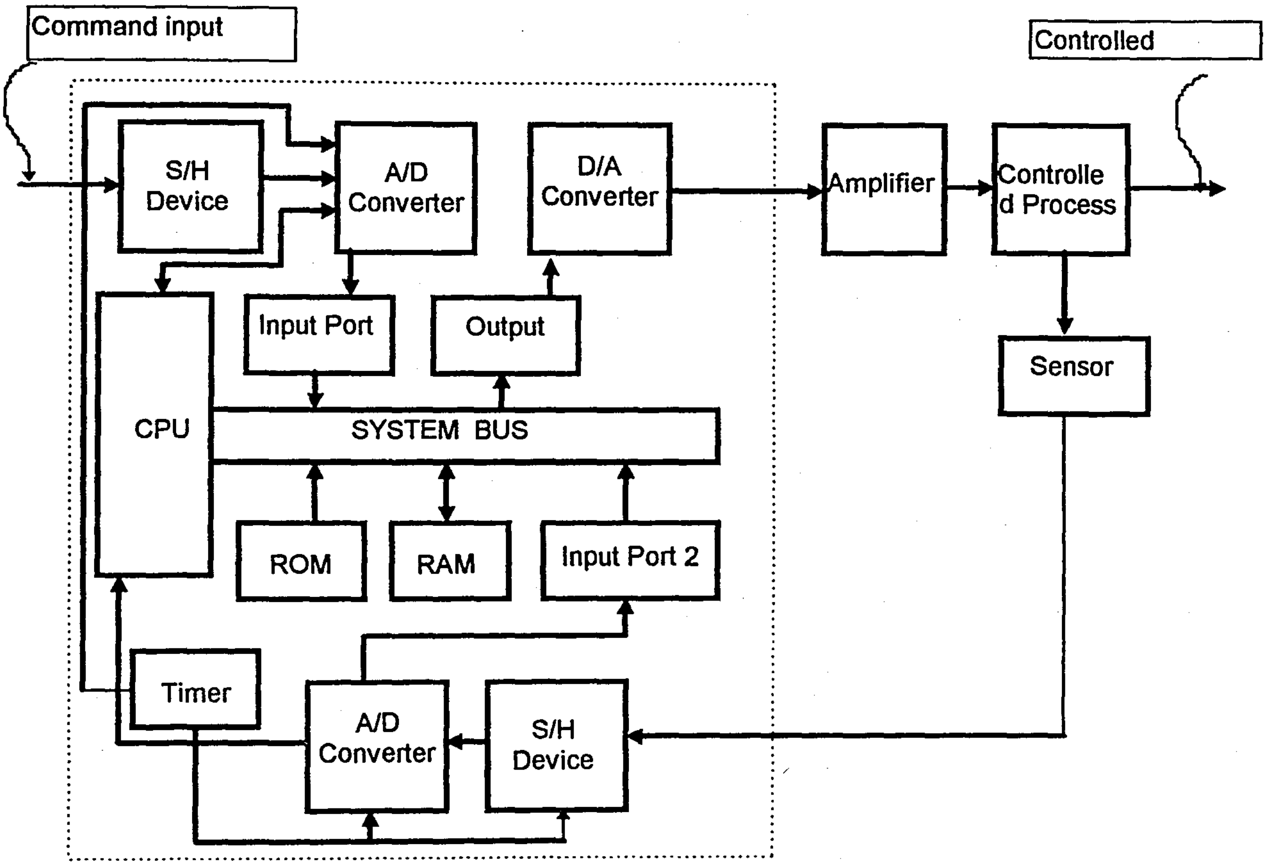


FIGURE 3.15. A µP based control system

An alternative to a hardware implemented timer would be to use timing loops, realized by software. The µ controller in digital control scheme in figure 3.15 uses one ROM chip, one RAM chip and three I/O ports. In general, a number of ROM and RAM chips and I/O ports may be required depending on the system's tasks and implementation. In very simple control schemes, it may suffice to use processors with small amount of on-chip ROM and RAM. Processors with on-chip I/O lines, timer and ADC are very common nowadays.

3.1.3.2 Sampling Rate Selection

The selection of the sampling rate 1/T (or sampling frequency $\omega_s=2\pi/T$) for a control system is carried out during the control loop design. However, a further refinement may be needed during the implementation of the digital controller.

Sampling theorem and some thumb rules for selection of sampling rate during the control loop design are important subjects that are to be thought of seriously. At the implementation stage, the first and the foremost consideration is the detailed analysis of the computing tasks of the microcontroller (μC). Particular attention has to be given to the computing tasks which are to be performed periodically during each sampling interval, such as calculating of the error function and the realization of the digital controller transfer function. Also the conversation times of A/D ad D/A converters and the delay introduced by the S/H circuit should be considered. If the total conversion and computation time is T_c, then the inequality $T_c<T$ must be preserved. Some more trade-off considerations are present. The larger the value of T, the more adverse effects it has on the stability of the feedback system. On the other hand,, if T is large, we can use a slower and therefore less costly μC. Also, for a large T, we can use A/D and D/A converters with a lover conversion speed, which in turn decreases the cost.

The final decision, taking into account, the inequality $T_c<T$ and the trade-offs discussed above, should be made by the designer according to the particular properties, specifications and constraints involved in that system under consideration. A general solution, which would fit all possible cases, can not be offered.

3.1.3.3 Time Delay Effects

Microprocessors are relatively slow digital computing machines. In control applications, the effect of the time taken to execute the control algorithm in a μP is significant. Two immediate problems attributed to time delays are the following:

(i) The control algorithm which requires long computation time can not be used because a reasonable sampling period for the controlled plant is too short to process a time-consuming algorithm during the sampling period. This is particularly true for controlled plants with very short time constants.

(ii) A control algorithm which is saved of situation (I) above, may have time delay sufficiently large to have an adverse effect on the stability of the closed-loop system. For systems, with a large time delay, control algorithms, control algorithms which can account for the input delay time are essential.

It may be noted that, in general, fixed point arithmetics takes shorter time to execute on a µP. This is why in many cases fixed-point arithmetic calculations are preferred in industrial applications.

3.1.3.4 Digital Quantization

In any computer realization, we must concern ourselves with the quantization effects of having finite register lengths, and particularly so with fixed-point arithmetic and small word-length machines(microprocessors).

Fixed-point arithmetic, in general, takes shorter time to execute on a microprocessor. Adverse effects of time delay can thus be reduced by using fixed-point number format. In this format, a binary point (corresponding to a decimal point in decimal numbers) is implicitly or explicitly located in a fixed position. Thus in the integer 0111, an implicit binary point lies immediately to the right of the least significant bit. No significant conceptual change results when we assume that the binary point is in some other position. Relocating the binary point in the number format has the effect of multiplying every number by a constant that is positive or negative power of two. In our discussion, lets assume that an (n+1) bit processor is used for digital implementation of control algorithms. In the 2's complement representation (which is most often used in digital computers), the sign bit of a positive number is 0 and the remaining n bits represent the number's magnitude in standard fashion. Assuming a number format where the binary point is fixed just to the right of the most significant magnitude bit, a real positive number x will be quantized to Q(x):

$$Q(x)=0.b_{n-1}\ldots b_1 b_0\ ;\ b_i \text{ is either 0 or 1 } ;\ 0 \le x < 1 \qquad (1)$$

$$|\,Q(x)\,| = \sum_{I=1}^{n} b_{n-i} * 2^{-i} \qquad (2)$$

For $-1 \le x < 0$, the binary representation is obtained by first forming a binary number for $|x|$ as per (1) and then complementing all bits and adding '1' to the least significant bit.

$$Q(x): 1.a_{n-1} * a_{n-2}* \ldots a_1 a_0\ ;\ a_1 \text{ is either 0 or 1 } ;\ -1 \le x < 0 \qquad (3)$$

It is obvious that

$$0 \le |Q(x)| < 1 \qquad (4)$$

If a number of x, converted to a binary fraction as an infinite series, is truncated at n bits owing to the finite word-length of the processor, the error $e_t = Q(x) - x$ introduced by this truncation is bounded as given below:

$$0 \ge e_t > -2^{-n}\ ;\ e_t = Q_t(x) - x \qquad (5)$$

The quantization characteristic for $Q_t(x)$ is illustrated in figure 3.16. The non-linear transfer characteristic introduces many problems in implementation of digital control algorithms.

A common alternative to truncation is round-off quantizer. The process is same as is common with ordinary base-ten numbers, where 1.115 is rounded to 1.12 but 1.114 becomes 1.11 to two (decimal) places of accuracy. In our fixed-point binary format , the round-off error

$e_r(x)=Q_r(x)-x$ will be greater than $-2^{-2(n+1)}$ if the first bit lost is '0'and it will be less than $2^{-(n+1)}$ if the first bit lost is '1':

$$\frac{2^{-n}}{2} \geq e_r > \frac{-2^{-n}}{2} \qquad ; e_r = Q_r(x)-x \qquad (6)$$

The round-off quantizer transfer function characteristic is shown in figure 3.17.

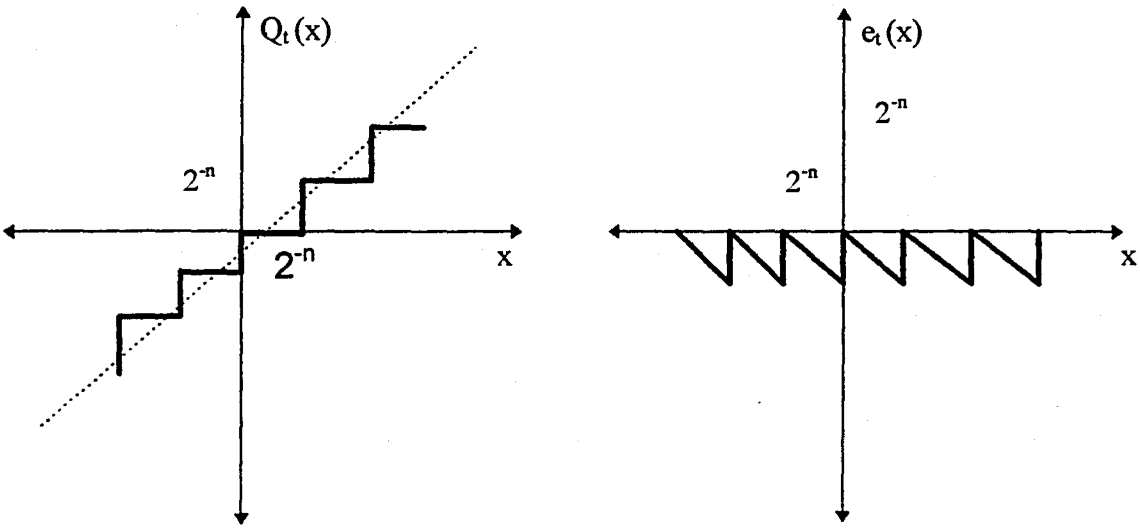


FIGURE 3.16. Transfer caracteristic of truncation quantizer.

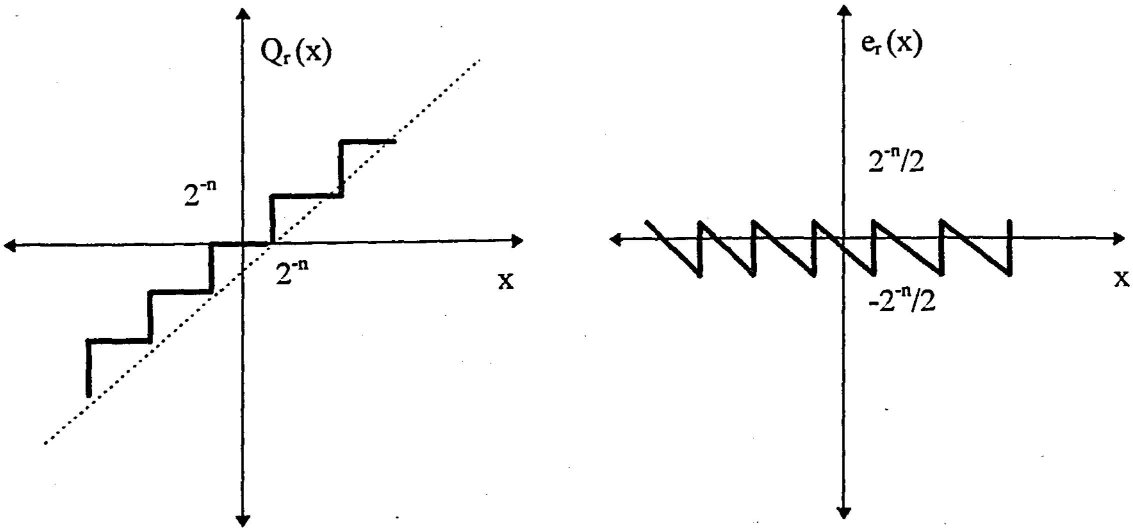


FIGURE 3.17. Transfer caracteristic of round-off quantizer.

Since the error due to truncation equals a constant plus round-off error , round-off quantization is used in all the analysis and assembly language programming throughout this thesis.

Note 1: By performing an A/D conversion, the analog signal is represented by their n-bit binary equivalents. In other words, we quantize the analog signals with a quantization level q , equal to the resolution of the binary representation; $q=2^{-n}$ (refer fig 3.18). Naturally, due to quantization procedure, the value of the discrete signal obtained after A/D conversion is different from the analog one. Thus a quantization error which is the difference between the values of the analog and digital signals is introduced. That was demonstrated in figure 9 that the quantization error is between -q/2 and q/2.

Note 2: During a multiplication of one digital word with another inside the microprocessor, the binary word of length n bits times another binary word of length n bits can generate binary product of length 2n bits. This 2n-bit product is quantized to an n-bit binary word. Every multiplication generates a quantization error source similar to that generated by A/D unit.

Note 3: The coefficients of a digital control algorithm are to be stored in a microcomputer for digital implementation. The parameter values are quantized to the accuracy of the machine. For general purpose microcontrollers of today the resolution of this accuracy is usually 8 bits.

The practical effect of this quantization error is that all of the system variables and coefficients are represented within the digital part of the system, at values which are different (by the quantization error at most) from the actual values.

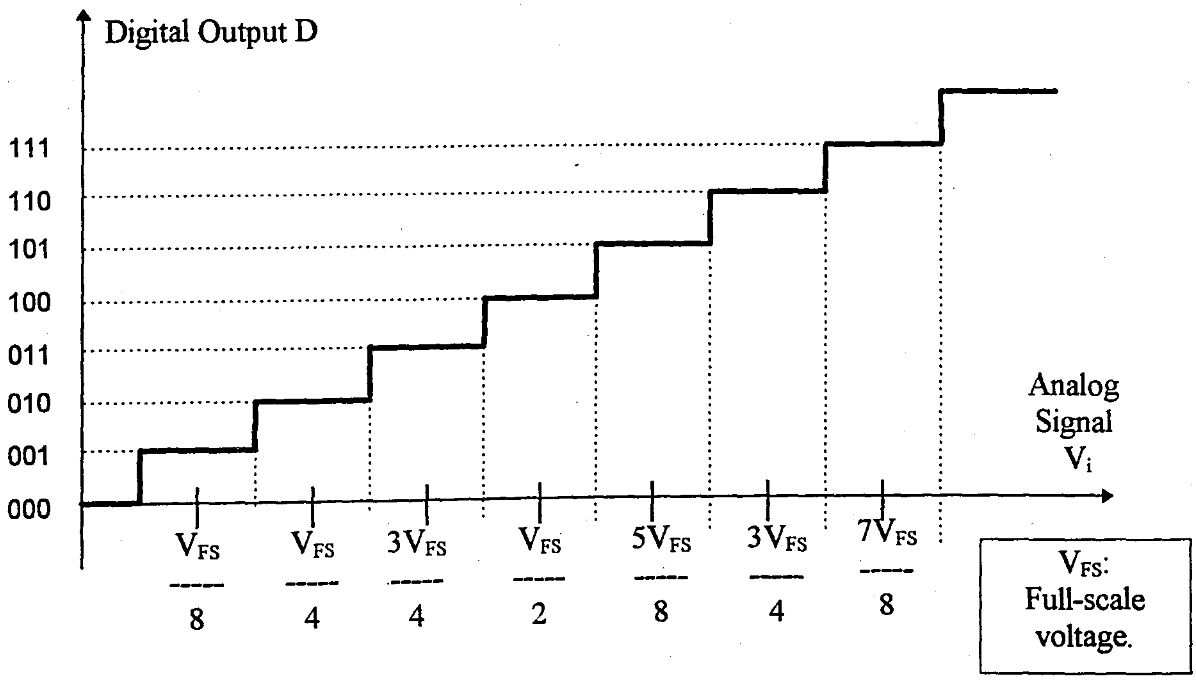


FIGURE 3.18. Relationship between analog and digital signals for a 3 bit A/D converter.

3.1.3.5 A Microprocessor Based Position Control System

This system is especially designed for rotating the sensors of the ultrasonic ranging system of a robot.

This section deals with the design and implementation of a position control system. The mechanical assembly consists of a stepper motor, and an optocoupler for position feedback. A small aluminum construction is used to keep them standing together. The system is designed around the base SGS St62E25 controller. No effort has been made to optimize the design in terms of hardware and software. The objective was to present a working demonstration for this kind of position control and use the design in the construction of the mobile robot, which was the main target of this thesis. The system rotates the stepper motor within 360° degrees and continuously gives the angle data to the databus.

3.1.3.6 Description Of The Control System

For the construction of this position control system, the following units are used:

1- A bipolar stepper motor.

2- An SGS Thompson ST6225 8 bit microcontroller.

3- Stepper motor driver IC (L298).

4- Power supply unit.

5- An optocoupler.

The system is like the one in the figure below. The microcontroller operates on its own and continuously rotates the sensor head of the robot within 360° degrees. This is the scanning process of the sensor head. The microcontroller gives the angle data to the main processor through the system data-bus. The main processor, independent of the working of the stepper-controller takes the angle data from the data-bus with any sampling period, necessary for its proper operation.

3.2 Stepping Motors And Their Interfacing To Microprocessors

A stepping motor (also called step *motor or stepper motor*) is an electromagnetic incremental actuator that converts electrical pulses into mechanical movements. In a rotary stepping motor, the output shaft of the motor rotates in equal increments in response to a train of input pulses. Because modern control systems often have incremental motion of one type or another, stepping motor has become an important actuator in recent years.

Stepping motors come in a variety of types, depending upon the principle of operation. The two most common types are the variable-reluctance (VR) motor and the permanent-magnet (PM) motor. The complete mathematical analysis of these motors is highly complex, since the motor characteristics are quite non-linear (refer Kenjo (1984)). The purpose of this introduction is to acquaint the reader with basic characteristics of Kenjo (1984), the Stepper Motor Handbook of AIRPAX / North American Philips Control Corp., and the Design Engineer`s Guide to DC Stepping Motors of Superior Electric Company, USA, have influenced the write-up of this chapter. Other useful references for in-depth study of the subject are; Kuo (1979), Proceedings of the International Conference of Stepping Motors and Systems, University of Leeds, 1979.

The following sections present the structures and fundamental principles of various types of stepping motors. Technical terms which are used in association with stepping motors are defined. The driving schemes used in my robot are expressed with detail.

3.2.1 Constructional Features Of Stepping Motors

3.2.1.1 Variable -Reluctance Motor

A variable-reluctance (VR) stepping motor has a wound stator and an unexcited rotor. Figure 3.23. illustrates a typical VR motor. This is a tree-phase motor having twelve stator teeth. The figure shows a set of four coils for one phase in each schematic; the coils for other two phases have been omitted for clarity. The rotor has 8 teeth. Both the stator and rotor materials must have high permeability and be capable of allowing a high magnetic flux to pass through them. Silicon steel is commonly used for this purpose.

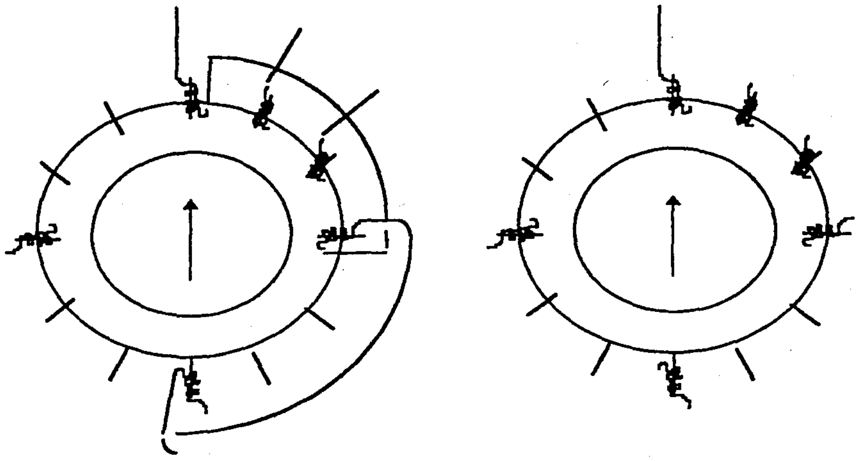


FIGURE 3.23. The Structure of a variable reluctance stepping motor

The operating principle of the VR motor is straightforward. Let any phase of the windings be energized with a dc signal. The magnetomotive force set up will position the rotor such that the teeth of the rotor section under the excited phase are aligned opposite the teeth on the excited phase of the stator. This is the position of minimum reluctance and the motor is in a stable equilibrium .

Figure 3.23 a illustrates the rotor in the position it would assume when phase C is energized. De-energizing phase C and energizing phase B will cause the rotor teeth nearest the phase B windings to be pulled into position directly in line with their magnetic fields as illustrated in Fig. 3.23 b. This causes the rotor to move one step (15° for the motor illustrated) in a clock wise direction.

Energizing phase A winding cause the rotor to step another 15° counter-clockwise (Fig.3.23 c.). However, if phase C windings have been energized instead of phase A windings, the rotor would have stepped in a clockwise direction back to the position shown in Fig. 3.23 a.

Thus the rotor rotates through a fixed angle (15° in this case), which is termed the *step angle* , as one switching operation to de-energize phase C and energize phase B is carried out. Another switching operation to de-energize phase B and energize phase A will make the rotor travel another 15°.The angular position of the rotor can thus be controlled in units of the step angle by a switching process. If the switching process is carried out in a sequence, the rotor will rotate with the stepped motion. The average speed and direction can also be controlled by the switching process.

A block diagram for drive system is shown in Fig. 3.24 Transistors are used as electronic switches for driving a stepping motor and switching signals are generated by digital ICs or a microprocessor. The frequency of pulses applied to the stator controls the speed of rotation, the number of pulses controls the angular displacement of the rotor and the sequence in which the pulses are applied controls the direction of rotation.

The number of steps per revolution can be calculated from a knowledge of the geometry of the motor as follows :

Let us assume that an excitation sequence is started from a phase in an m phase motor. When a cycle of the excitation sequence is completed and the first phase is again excited, having made m steps, the rotor will have rotated one tooth-pitch. Since m pulses are transmitted to rotate the rotor by one tooth pitch, mNr ; Nr = number of rotor teeth, pulses are needed to complete one revolution.

Therefore, number of steps per revolution, S = mNr. In Fig. 3.23 ., Nr=8 and m = 3. This gives S = 8x3 =24 steps/revolution, giving a basic step angle of 360/24 =15°.

The VR stepping motor in Fig. 3.23 is a single-stack motor; the three phases are arranged in a single stack, i.e., in the same plane. Another type of VR stepping motor is the multi-stack type wherein each stack corresponds to a phase and the stator and rotor have the same tooth pitch.

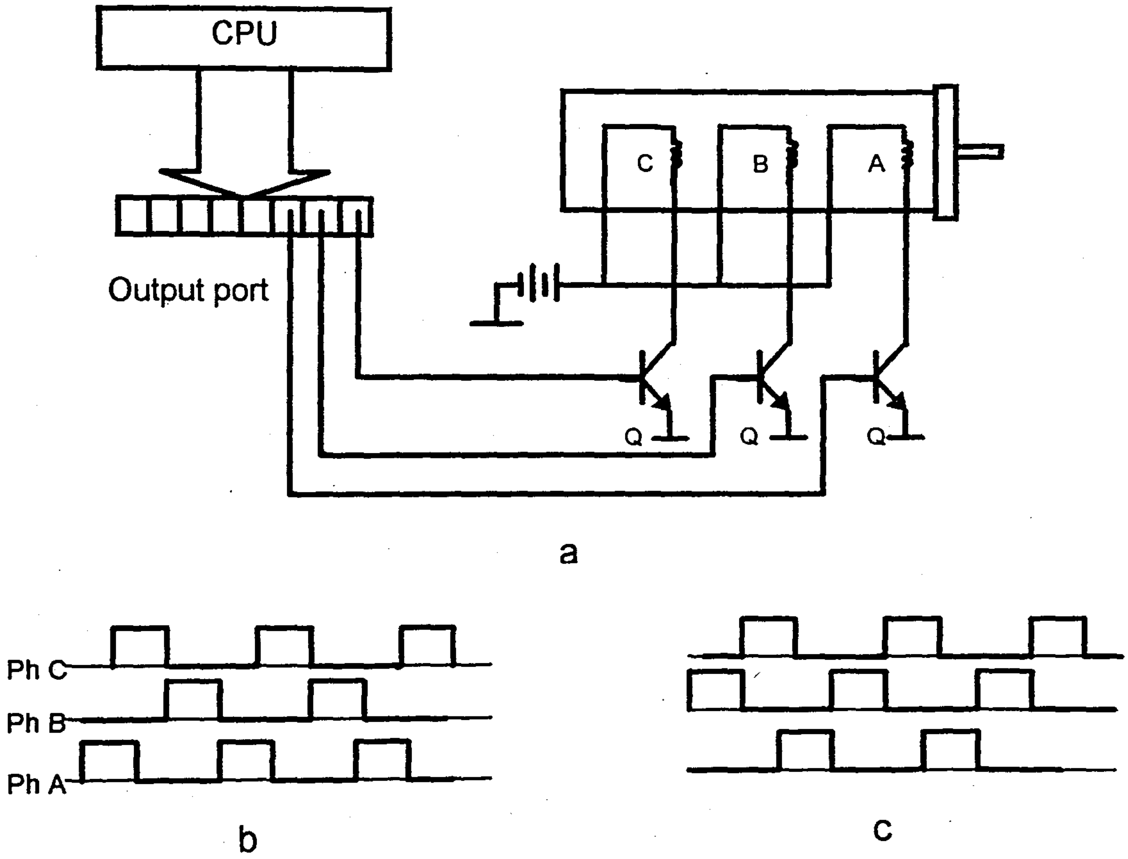


FIGURE 3.24. a) Drive system for VR motor in fig 3.23
b)Excitation sequence for counter-clockwise rotation
c)Excitational sequence for clockwise rotation

Figure 3.25 shows longitudinal cross-sectional view of a typical 3-stack motor. The toothed structure of stator and rotor is shown in the end

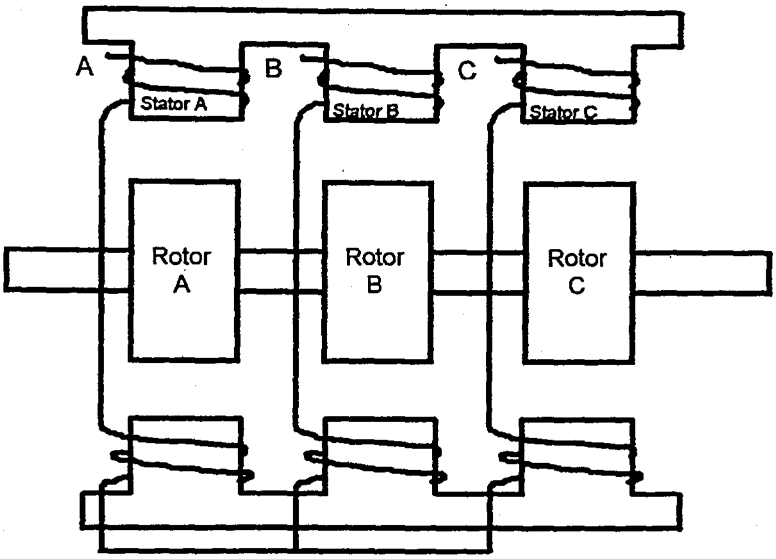


FIGURE 3.25. Longitudinal cross-sectional view of a 3-stack VR motor

iew in Fig. 3.26.; the stator teeth are arranged in groups on poles while the rotor teeth are distributed homogeneously on the periphery. The stator and rotor teeth are of the same size and therefore can be aligned as shown in this figure. The stators are pulse excited and the rotors are unexcited.

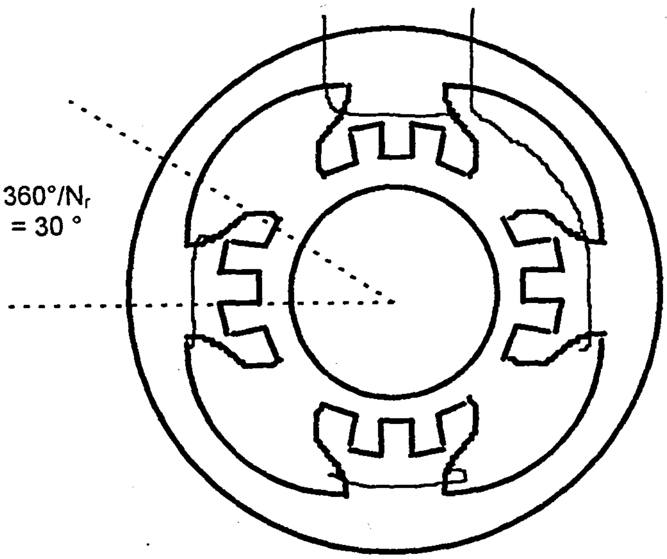


FIGURE 3.26. End view of stator and rotor (12 teeth) of a multi-stack VR motor

While the teeth on all the rotors are perfectly aligned, stator teeth of various stacks differ by an angular displacement of 360° /mNr., where Nr=number of rotor teeth and m=number of phases=number of stacks in a multi-stack motor. For the 3-stack motor shown in Fig. 3.25 and 3.26 with Nr=12, the stator teeth are misaligned by 360° /3 x12 = 10°. Figure 3.27 shows the developed diagram of the motor with rotor in such a position that stack C rotor teeth are aligned with its stator. Now, if phase A stator is excited, the rotor will move by 10°in the direction indicated. If instead, phase B is excited, the rotor will move by 10° opposite to the direction indicated. Pulse train with sequence ABCAB will make the rotor go through incremental motion in the indicated direction while sequence BACBA will make it move in opposite direction. Directional control is possible with three or more phases.

A unique feature of VR motors is the possibility of realizing small step angles. Four-phase motors with 50 rotor teeth giving a step angle of 1.8° are widely used.

3.2.1.2 Permanent-Magnet Motor

A permanent-magnet (PM) stepping motor has a stator with a number of teeth on it which carry the excitation current and a rotor in the form of a permanent magnet having a number of poles equal to the number of stator teeth.

We consider here a typical 2-stack PM motor (Figure 3.28). The two stator cups have two stator windings each of which produces a number of poles which are displaced circumferentially from one another by half a pole pitch.

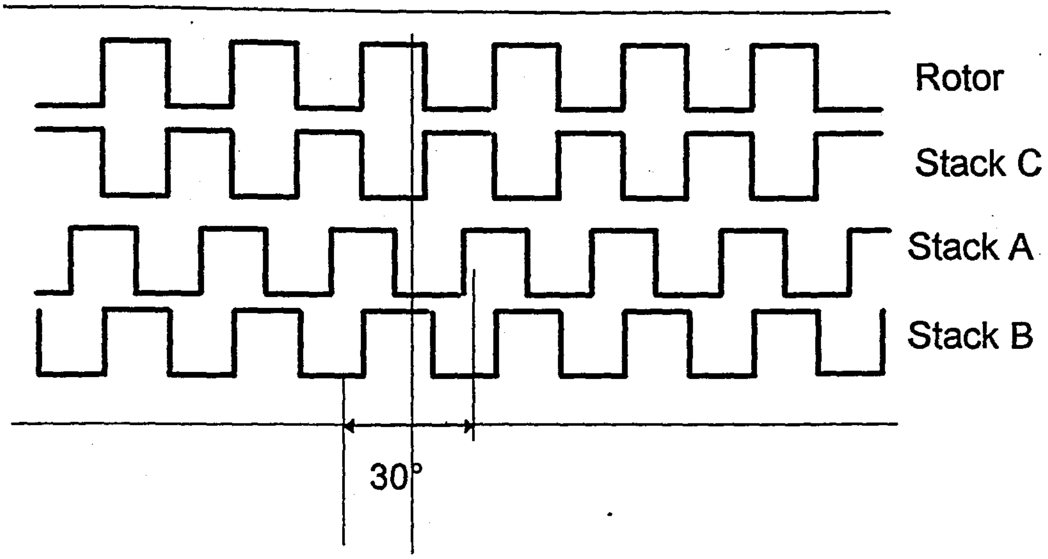


FIGURE 3.27. Developed rotor of and stator stacks of a 3-stack VR motor

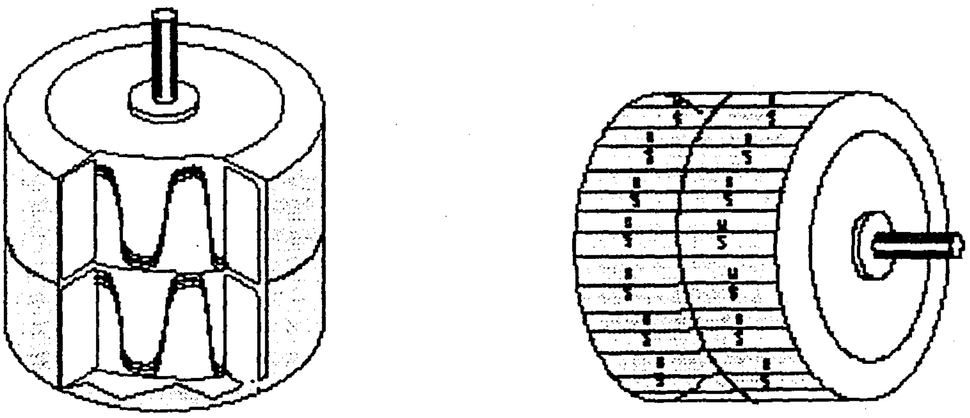


FIGURE 3.28. a) Cut-away diagram of a 2-phase PM motor.
b) Rotor magnetization

Between the two stator-cup pairs, the displacement is 1/4 of a pole pitch. Figure 3.29 shows the split-and-unrolled model of the 2 phase PM motor.

Let us examine the positional relation of the rotor's magnetic poles and the stator teeth in stack A. The rotor is first placed in position of state (a) and phase A is excited with positive current so as to produce magnetic poles in the pattern shown in Fig. 3.29. As is obvious, the interaction between the rotor and the stator (opposite poles attracting and the likes repelling) causes the rotor to move right. State (b) is an equilibrium position with phase A excited in the positive polarity. Next, if phase A is turned off and phase B is excited with positive current, the rotor will be driven further in the same direction, because the stator teeth in stack B are misaligned by a quarter tooth pitch to the right with respect to the teeth in stack A. To advance the rotor

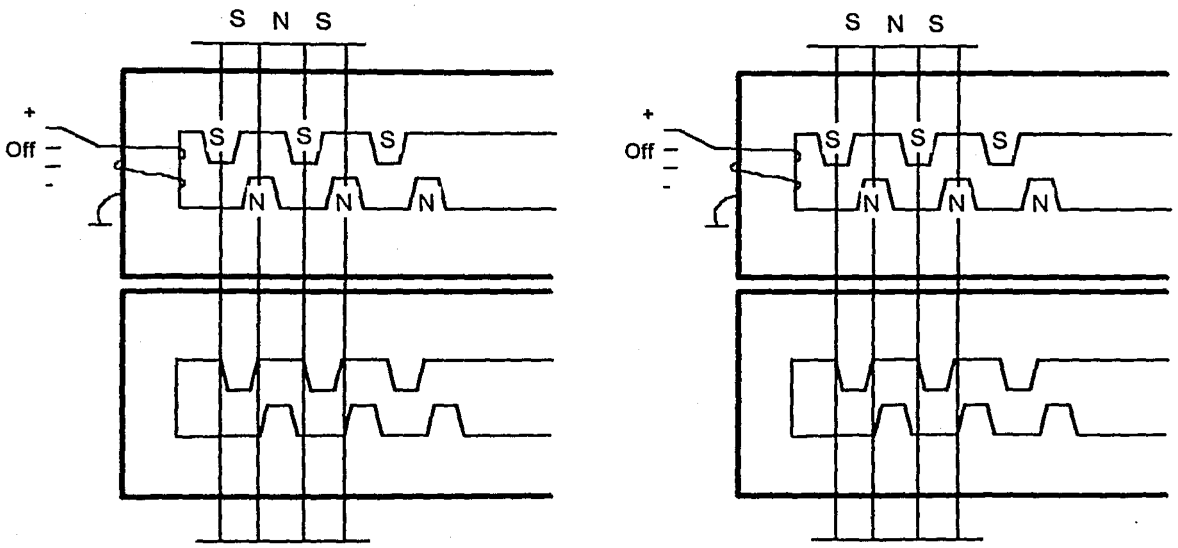


FIGURE 3.29. Layout diagram of a 2-phase PM motor

further , phase B is de-energized and phase A is excited with negative polarity. The switching sequence is summarized in Fig. 3.30. The two-phase motor with 12 pole-pairs per stator cup moves 48 steps per revolution or 7.5° per step.

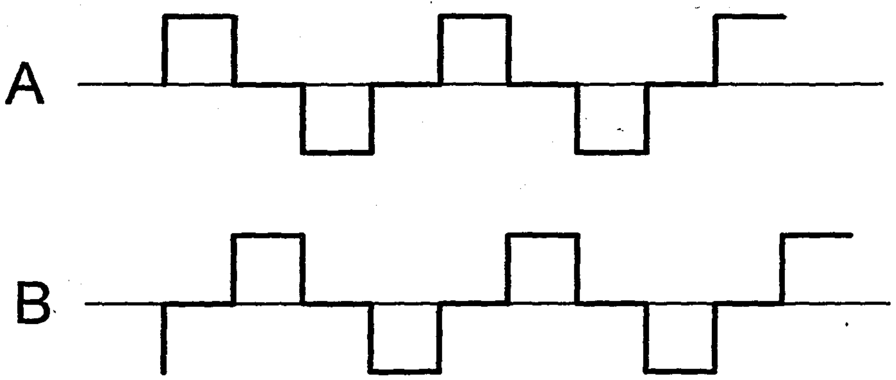


FIGURE 3.30. Excitation sequence for the 2-phase PM motor of the Figure 3.29.

A feature of the PM motor is that, even when the motor is unpowered, a torque has to be applied to the shaft to displace it from the rest position. This is known as the *detent torque* of the motor, and is due to the attraction between the permanent magnet rotor poles and the residual magnetic poles on the stator. The VR motors do not exhibit this `detent' ability with all windings de-energized.

In a PM motor, to decrease the step angle, the number of rotor poles and stator teeth must be increased. However, there is a limit to both teeth and magnetic poles. As an alternative, the *hybrid structure* is widely employed in PM motors having a small step angle. The hybrid type motor is a combination of the PM type and the VR type. A typical form of construction for a 2-phase hybrid motor has as a stator core structure resembling that of a VR motor -a stator -stack has a number of salient poles; the tip of each pole being subdivided into a number of teeth by means of slots. The rotor comprises a cylindrical shaped magnet which is magnetized lengthwise. Each pole of the magnet is covered with uniformly toothed soft steel arranged in two sections. The teeth on the two sections are misaligned with respect to each other by half tooth pitch. (In some motors, the rotor teeth are aligned with each other but the stacks of stator core have a misalignment).

The step angle depends on the number of teeth on the rotor and on the stator and typically lies in the range 0.9° to 5°. The most popular hybrid motor is the four-phase step motor, with step angle 1.8°. The motor that i've used for the rotation of the sensors in my robot is also one with a step angle of 1.8°. The motor is extracted from a junked computer hard drive. That was probably the cheapest method for obtaining a stepper motor. The motor was in perfect condition and all I did was to do some testing work to understand which kind of motor I have and special properties of the motor if there exists any.

3.2.1.3 Important Parameters Of Stepping Motors

In the following we describe important parameters of stepping motors. Some of these parameters have already appeared in the earlier section.

Step angle : A stepping motor is driven by sequences of excitation and by each such sequence, the motor moves through a fixed angle. The rated value of this angle, in degrees, is commonly referred to as the step angle θg. Step angles of some of the modern motors are 0.72°, 1.8°, 2.0°, 2.5°, 5°, 7.5°, 15°, 18°.

Step per revolution :The term describes the total number of steps required for the output shaft to rotate by 360°.

$$\text{Steps per revolution, } S = 360° / \theta g.$$

Stepping rate : This parameter (number of angular movements/sec) gives a measure of the speed of operation of a stepping motor. The maximum stepping rate is specified at a given torque produced at the output shaft. If the motor is subjected to run a stepping rate higher than the specified one, then the motor may lose steps and it may not be able to provide the rated torque at that speed. Motors having stepping rate as high as 5000 steps/sec are available.

It may be noted that stepping rate does not specify the rotational speed (rpm). The relation between the rotational speed and the stepping rate is given by

$$n = 60\, f / S$$

where

n = rotational speed (rpm)

f = stepping rate (steps / sec)

S = steps per revolution

Step accuracy : This is a percentage error between the angle of the motor shaft accomplished after one sequence of excitation and the motor's rated step angle. Under normal running conditions, accuracy as high as 96 % can be achieved. One most important attribute of inaccuracy, as related to stepping motor, is that it is noncumulative. Thus, overall accuracy in any operation implemented by large number of sequences does not get impaired.

Holding torque : With the motor energized and in standstill (equilibrium) position, the torque required from an external source to break away the motor from this position is called the holding torque. Holding torque is a basic characteristic of stepping motors and provides positional integrity under standstill conditions.

The relation between the external torque and displacement (T / θ) is typically of the form shown in Fig. 3.31. The holding torque, which is the maximum static torque, occurs at θ = θM . At displacements larger than θM , the static torque does not act in a direction towards the original equilibrium position, but acts in the opposite direction towards the next equilibrium position. The angle at which the holding torque is produced is not always separated from the equilibrium point by one step angle.

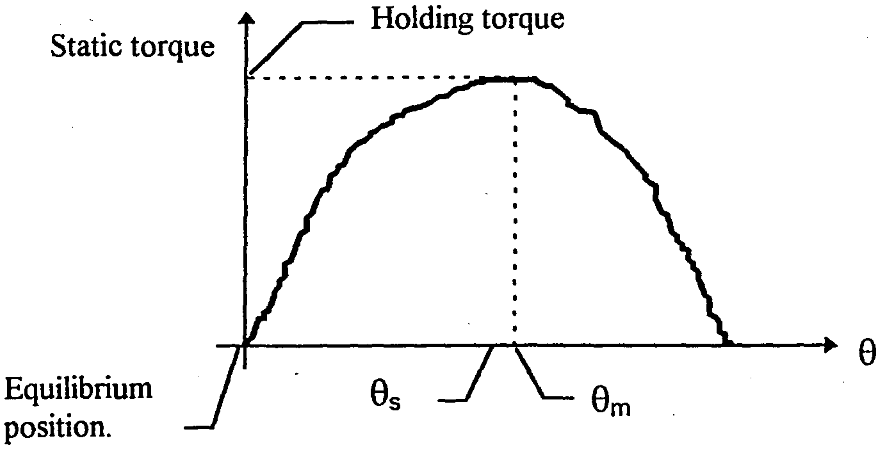


FIGURE 3.31. T/θ Characteristic

Detent torque : This torque is also referred to as the residual torque. It is specified at standstill condition under no excitation. It is the result of permanent magnet flux of PM rotor acting on the residual flux on stator poles. Bearing friction also adds to this torque. This torque of PM stepping motors (about 1/10th the holding torque) is useful in holding a load in the proper position even when the motor is de-energized. The position, however, will not be held as accurately as when the motor is energized.

Figure 3.32 compares the T/I characteristics of a typical hybrid motor with those of a VR motor, the step angle of both being 1.8°.

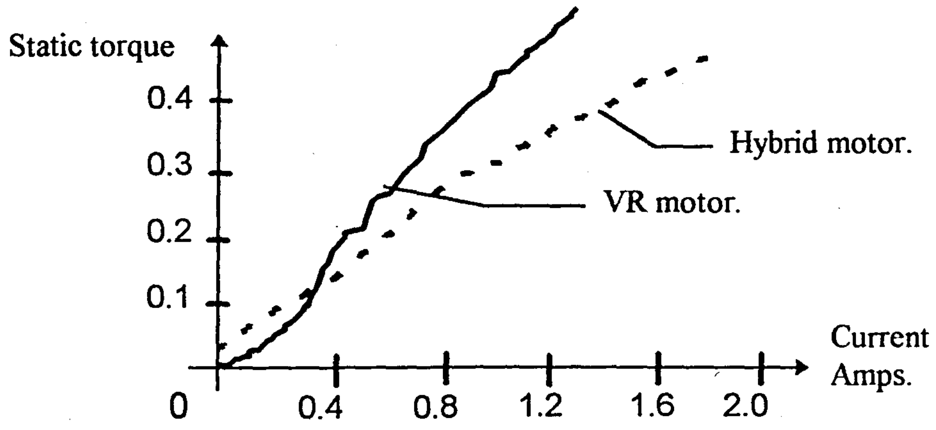


FIGURE 3.32. T/I characteristics.

Dynamic torque : Under running condition at a given stepping rate, this torque can be determined from the torque vs. stepping rate curves. This is, in fact, the torque produced by the motor on excitation under rated conditions. The dynamic torque of a motor is the most important data and it plays a major role in selection of a motor for a specified application. The data is affected mainly by the following three factors.

Stepping rate

The dynamic torque of the motor gets reduced at higher stepping rates.

Excitation current in winding of the motor

The magnetic pull between stator teeth and rotor teeth is dependent on magnetization of both the sides. In permanent magnet and hybrid motors, rotor magnetization is fixed at the time of manufacturing, thus only stator magnetization plays a role in determining the torque. The magnetization of stator is obviously dependent on excitation current through its windings. the excitation current and produced torque are almost linearly related until the magnetic flux paths within the motor saturate. As the motor nears saturation, it becomes less efficient and thus does not justify the additional power input.

Kind of excitation

Stepping motor can be excited in various modes. Bipolar e-citation gives 30 to 40 % improvement in torque-speed characteristics over unipolar scheme in the low speed range. these issues will be taken up in the next section.

A typical torque vs. stepping rate characteristic graph is shown in Fig. 3.33 in which curve 'a' gives pull-in torque vs. rotor steps/sec and curve 'b' gives pull-out torque vs. rotor steps / sec.

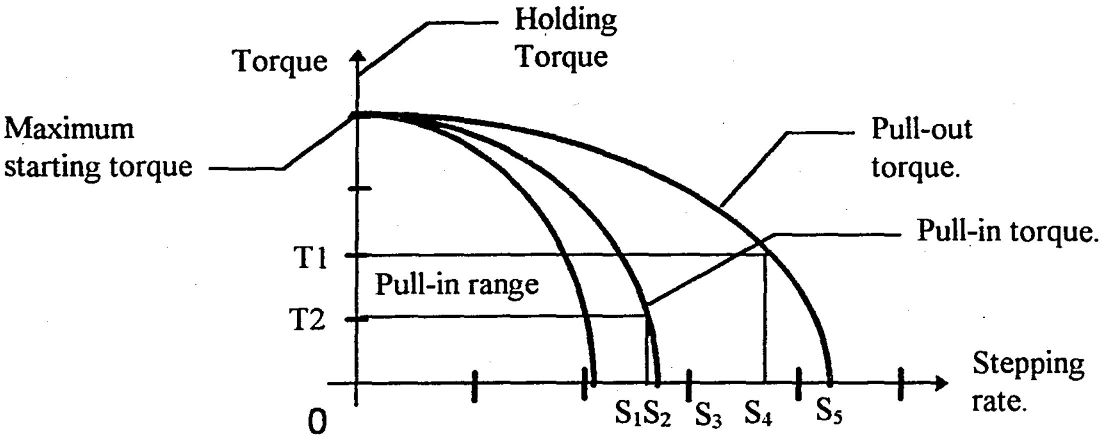


FIGURE 3.33. Dynamic characteristics

The pull-in range of the motor is the range of frictional load torque at which the motor can start and stop without losing steps. For a frictional load requiring torque T1 to overcome friction, the maximum pull-in-rate is S1 steps per sec. S2 is the maximum pull-in rate at which the unloaded motor can start and stop without losing steps.

When the motor is running, the stepping rate can be increased above the maximum pull-in-rate and when this occurs the motor is operating in the slew-range region. The slew range is the range of switching speeds within which the motor can run unidirectionally, but cannot be started or reversed.

(at shaft torque T1 ,the motor cannot be started or reversed at step rate S3).

When the motor is running in the slew range, it can follow changes in the stepping rate without losing step but only with a certain acceleration limit.

For a frictional load requiring torque T1 to overcome friction, the maximum switching rate at which the motor can run is S4 . S5 is the maximum slewing rate at which the unloaded motor can run without losing steps.

Curve `c' in Fig. 3.33 gives the pull-in torque with external inertia. It is obvious that if the external load results in a pull-in torque curve `c`, the torque developed by the motor at step rate S1 is T2 > T1. The stepping motors are more sensitive to the inertia of the load than they are to its friction.

Torque-to-inertia ratio : It is the holding torque of a motor divided by the rotor inertia. The better the torque-to-inertia ratio, the better the response.

It is important to note that both the torque and the inertia of a motor increase directly as the stack length of the motor, while the torque increases with the diameter, the inertia increases with the square of the diameter. To obtain a high performance motor, it is preferable to design a long motor (related to the diameter) rather than a short motor of large diameter.

Step response : When a single step of a motor is made, a typical response is as shown in Fig.3.34. The rotor should be allowed to settle down before the next stepping pulse is applied. However, at very high stepping rates, the next pulse could be applied at one of the instants a, b, c, or d in Fig. 3.34. In such a case, it is quite possible that the motor may either understep or overstep or not step at all, i.e., there may be ambiguity in the step position.

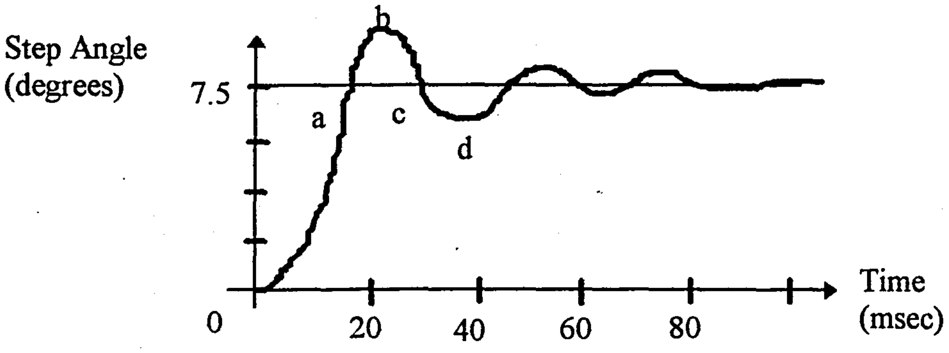


FIGURE 3.34. Single step response

The actual response for a given motor is a function of the power input provided by the drive and the load. Increasing the frictional load or adding external damping can thus modify this response. Mechanical dampers such as slip pads or plates, or devices such as a fluid-coupled flywheel can be used but add to system cost and complexity. Figure 3.35 shows typical step response curves for an unloaded motor and the motor with mechanical damping. As is seen, the settling time gets reduced from 120msec to 20 msec.

Electronic damping can also improve the step response. This method requires a degree of accurate 'tuning'. The motor is commanded to take a

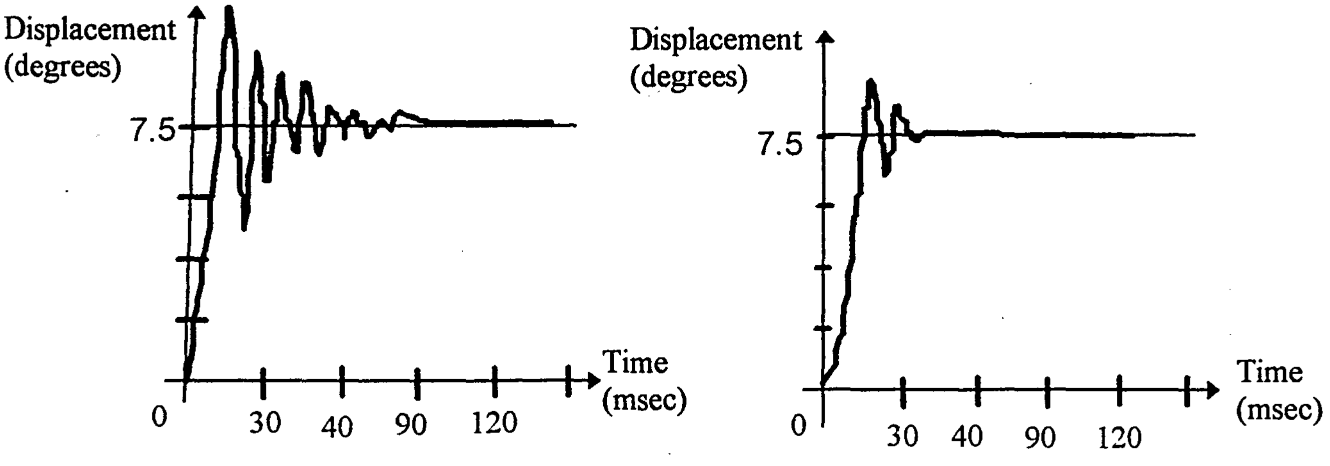


FIGURE 3.35. Single step response of an unloaded motor and the motor with mechanical damping

step as shown in Fig. 3.36 (point a). At some time interval, (point b), the electronic circuit commands the motor to 'reverse' or a winding is 'shut off', thereby reducing the forward velocity of the step. The motor is then commanded to complete the step (point c) and it does so with a minimum settling time. It is important to realize that the timing of the reverse command (point b) is critical and determines the damping characteristics of the specific motor-load combination.

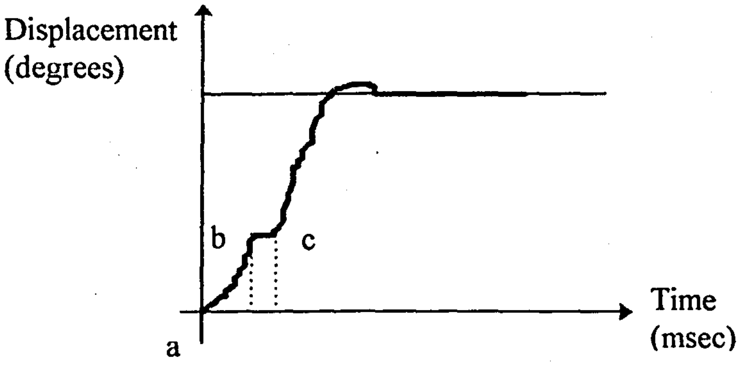


FIGURE 3.36. Single step response with electronic damping

Multiple stepping can also be used to advantage to improve the step response. A 7.5° motor moving 12 steps to give a 90 ° output would have less overshoot, be stiffer and relatively more accurate than a motor with a 90° step angle. Also the pulses can be timed to shape the velocity of the motion, slow during start, accelerate to maximum velocity, then decelerate to stop with minimum ringing.

Resonance : Some stepping motors exhibit a phenomenon known as resonance, which occurs under certain conditions as follows. the mass of the motor and load combine with magnetic stiffness of the motor to form a spring-like system. At certain stepping rates, a resonant effect causes the rotor to oscillate at random and lose step rate curve, which is shown by a broken line on manufacturer's performance curves.

In applications where the motor must be operated at the 'natural' frequency resulting in resonant behavior, adding inertia loading (flywheel) will usually allow satisfactory performance. The natural frequency is lowered as inertia is increased. However, adding inertia can cause a reduction in overall system performance. Stepping motors are available with viscous coupled inertia damper, called *Lanchester Damper.*

Alternative method is to operate at a higher stepping rate wherever possible. Also the characteristics of the electronic drive can be changed to permit a 'softer' step; of course at the cost of speed-torque performance.

Ramping : The process of controlling the switching frequency to accelerate a motor from zero speed to maximum speed (without losing steps) , as well as to decelerate the motor from maximum speed to zero speed is called ramping. Ramping increases the capability of driving the motor and load to higher speed levels, particularly with large inertial loads. A typical ramping control plot for an incremental movement with equal acceleration and deceleration times would be as shown in Fig. 9.15.

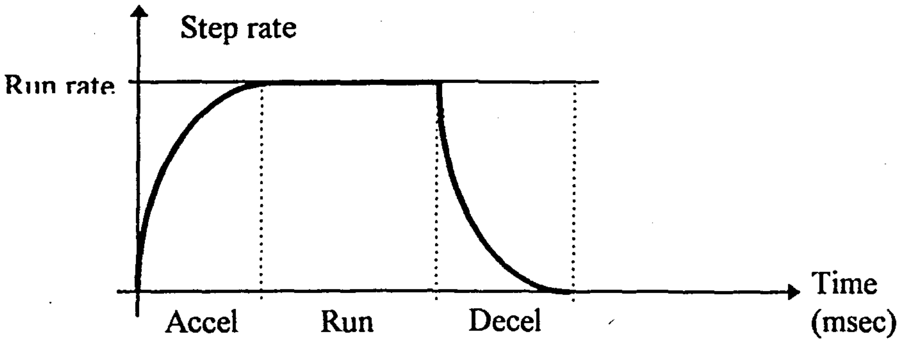


FIGURE 3.37. A ramping control plot

3.2.1.4 Operational Features Of Stepping Motors

Having read through last section on constructional features of stepping motors, we know that various phases of a stepping motor have to be energized in a particular sequence to rotate the motor in clockwise or counter-clockwise direction. the switching sequence for a particular motor is supplied by the manufacturer.

To highlight the principles involved, in the following we consider, specifically the 2-phase permanent magnet motor whose constructional details were given in the last section .

For the special properties of the stepper motor used in my robot, I generally ignored them assuming that the motor is able to rotate two very light sensors tied directly on its shaft. Starting with this assumption, no changes needed to be done. And all the system worked quite well. But apart from the bipolar driving scheme in figure 3.42, I had several problems in maintaining good motion control on stepper motors. These problems are generally encountered because of the difficulty of designing a good driver circuitry for the windings of the motors which will not effect the working of the driver microcontrollers. By using a bipolar stepper motor and a suitable driver IC (that was the L298 from SGS Thompson) the system was easier to construct and worked with no difficulty. The multiple source capability of the driver IC was also very useful for reliability.

3.2.1.5 The Four-Step Switching Sequence

The four-step switching sequence for the 2 phase PM motor is given in Fig. 3.38. Each time the coil excitations are switched as per the sequence given in the chart of Fig. 3.38, the motor takes a `step`. The chart is circular in the sense that the next entry after step 4 is step 1. To rotate the motor in a clockwise direction, the chart is traversed from top to bottom and to rotate the motor in counter-clockwise direction, the chart is traversed from bottom to top.

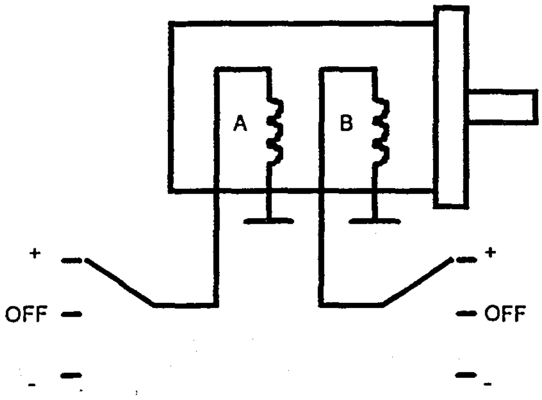


FIGURE: 3.38. Four-step switching sequence (continued)

Step	Coil A	Coil B
1	+	Off
2	Off	+
3	-	Off
4	Off	-

FIGURE: 3.38. Four-step switching sequence (continued)

For every `step` taken, the rotor moves 1/4 th of a pole pitch; so for every four steps taken the rotor moves one full pole pitch.

It may be noted that in the four-step sequence of Fig. 3.38, only one winding is energized at a time; the holding and dynamic torque with rated voltage applied will be low in this case. This method of excitation has the advantage of increased efficiency; however the step accuracy is reduced.

3.2.1.6 The Eight-Step Switching Sequence

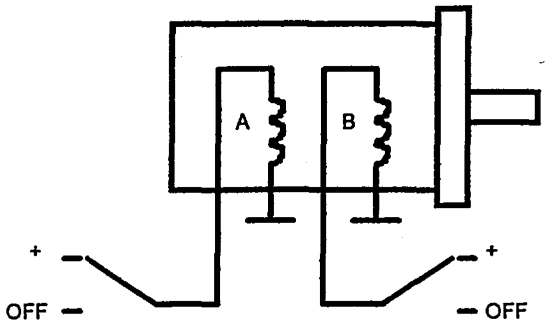
It is also possible to rotate the motor according to an eight-step sequence to obtain a half step-such as 3.75° step from a 7.5 ° motor. Shown in Fig.3.39 is a chart giving the winding switching sequence of operating in this mode include finer resolution, the reduction of resonant amplitudes and greater capability. However, the holding torque will vary for every other step since only one winding will be energized for a step position, but on the next step two windings are energized. This gives the effect of a strong step and a weak step.

3.2.1.7 Bipolar Operation

In a bipolar operation, the stator flux is reversed by reversing the current in the winding. Figure 3.40 suggests a bipolar four-step switching sequence wherein both the coils are always powered during the rotor operation and the magnetic polarity of the stator poles is switched by

Step	Coil A	Coil B
1	+	+
2	+	Off
3	+	-
4	Off	-
5	+	-
6	+	Off
7	+	+
8	Off	+

FIGURE 3.39. Eight-Step switching sequence



Step	Coil A	Coil B
1	+	+
2	+	-
3	-	-
4	-	+

FIGURE 3.40. Bipolar four-step switching sequence

switching the direction of current in the coil. The bipolar winding gives the optimum motor performance at low to medium step rates (Fig. 9.19).

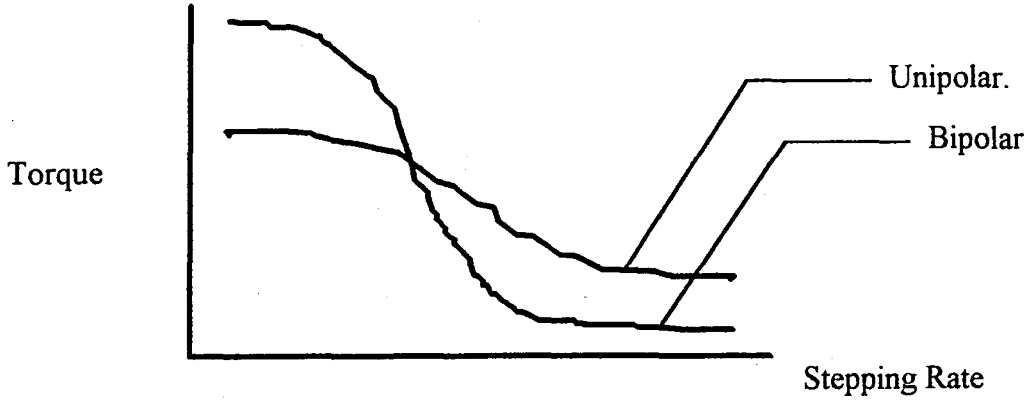
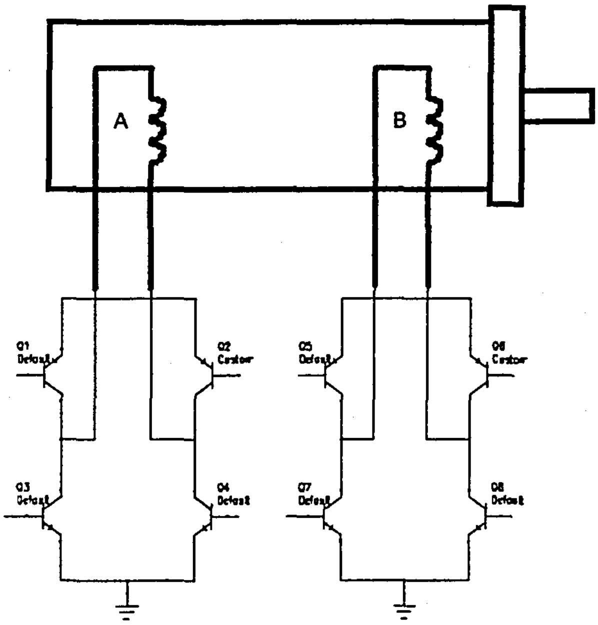


FIGURE 3.41. Torque/Stepping rate curves

Unfortunately, the drive circuits for this method of control are usually very complex and costly. Figure 3.42 shows a simple scheme of bipolar drive. This scheme requires eight power transistors, while as we shall see shortly, the unipolar scheme requires only four power transistors. In addition, in the bipolar drive, care must to taken to design the circuit so that the transistors in series do not short the power supply by becoming `on` at the same time.



Step	q1-q4	q2-q3	q5-q8	q6-q7
1	ON	OFF	ON	OFF
2	ON	OFF	OFF	ON
3	OFF	ON	OFF	ON
4	OFF	ON	ON	OFF

FIGURE 3.42. A simple bipolar drive scheme

3.2.1.8 Bifialer Winding

For a step angle of 7.5°, an alternative to 2-phase stepping motor illustrated in Fig. 3.29 is a four-phase stepping motor in which the coils of four phases A1, B1, A2 and B2 are would in a particular fashion. The stator cup A has now two coils A1 and B1 on the same bobbin. Flux is reversed by energizing one coil or the other from a single power supply (Fig. 3.43).

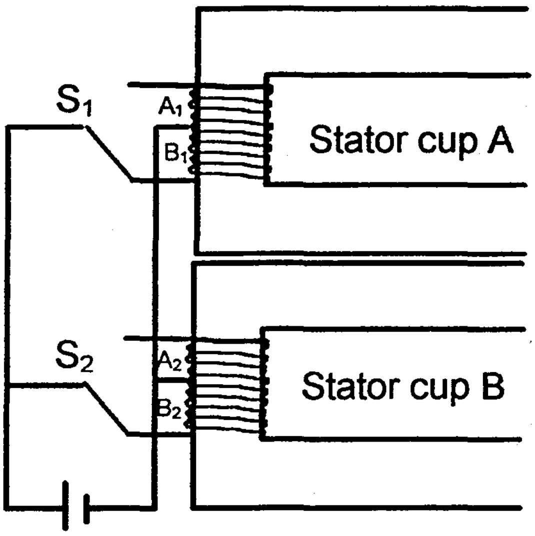
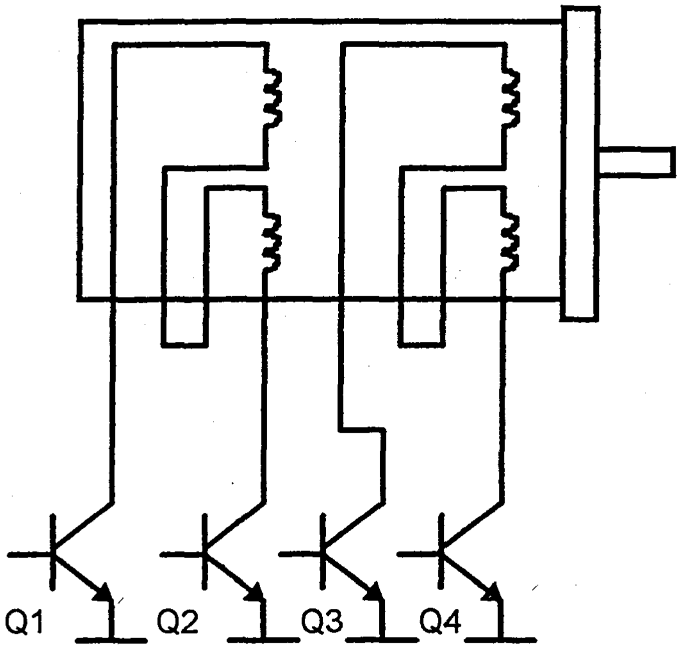


FIGURE 3.43. Bifialer winding scheme

In bifialer scheme, the number of winding is twice the corresponding value in a monofialer scheme. In order to get twice as much wire in the same relative frame size, it is necessary to reduce the diameter of the wire thereby increasing the resistance of the winding and resulting in a better L/R time constant. The better the L/R constant, the better the performance as we shall see in a later section.

3.2.1.9 Unipolar Operation

The bifialer scheme leads to unipolar drive of the stepping motor; current in one phase gives rise to only one type of magnetic polarity. The 3.44 illustrates a simple unipolar drive scheme. It is observed that compared to bipolar scheme of Fig. 3.42, the number of power transistors is reduced to half. Also pulse timing is not as critical to prevent a current short through two transistors as is possible with a bipolar drive.



Step	Q1	Q2	Q3	Q4
1	ON	OFF	ON	OFF
2	ON	OFF	OFF	ON
3	OFF	ON	OFF	ON
4	OFF	ON	ON	OFF

FIGURE 3.44. A simple unipolar driving scheme.

3.3 Stepping Motor Drive Circuits

we know that a stepping motor is driven from one step position to the next by switching a dc supply from one set of stator windings to another. For a standard four-phase motor, the stator excitation sequences for half-and full-step operation were listed in Figures 3.38-3.44. Since the generation of these sequences is a straight forward logic design problem, they can be generated either in control program if the motor is to be directly controlled from a microprocessor, or dedicated logic can be designed to generate the sequence. In both the cases, level conversion circuits are needed to amplify the low-power logic levels so that rated winding currents can be achieved. We will refer to these circuits as drive circuits. The software solution to logic sequencing is assumed, i.e., microprocessor performs the function of logic sequencer.

Drive circuits can be simple or complex, depending upon performance requirements. In the following, brief description of various drive types is given. To limit the discussion to mainly the principles involved, four-phase unipolar drives are considered.

L-R Drive

The simplest method of stepping motor control is as shown in Fig. 9.23. The output signals from a microcomputer port are transmitted to the input terminals of power transistors which control the turning on/off of the motor windings. As we shall see shortly, the performance of this system is largely governed by the inductance-to-resistance ratio (L/R) of the motor windings.

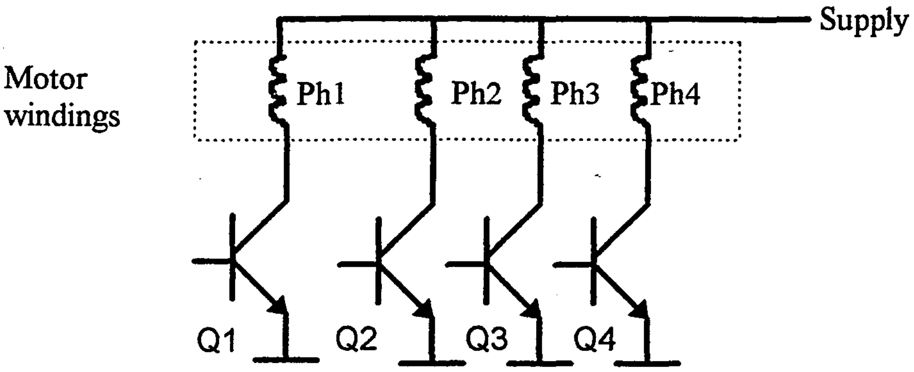


FIGURE 3.45. L/R Unipolar Drive

A problem with the type of switching described above is that when a power transistor in Fig. 3.45 is turned off, a high voltage builds up due to L di / dt , which may damage the transistor. This surge in voltage can be suppressed by connecting a diode in parallel with the winding in the polarity shown in Fig. 3.46a. Now there will be a flow of circulating current after the transistor is turned off and the current will decay with time. This method is very simple, but a drawback is that the circulating current lasts for a considerable length of time and it produces a braking torque.

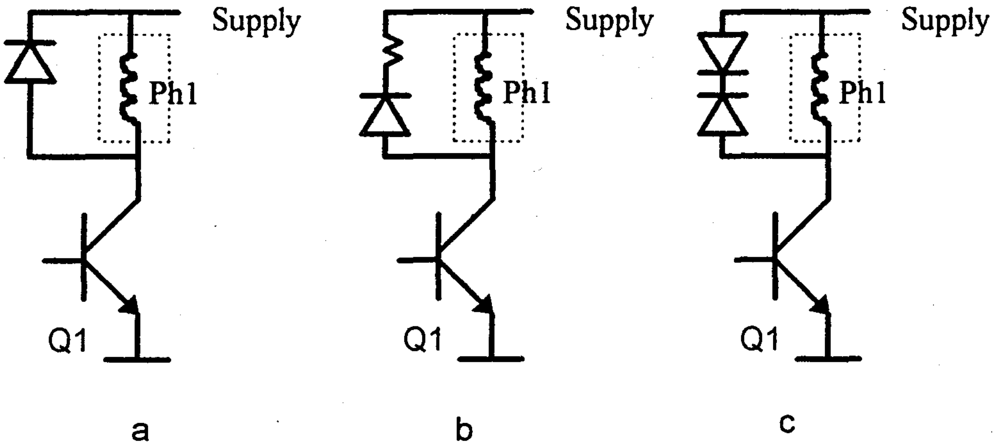


FIGURE 3.46. Diode surge-suppressor in drive circuits

A resistor connected in series with the diode (Fig. 3.46b) will damp the circulating current quickly. In this case the voltage applied to the collector at turn-off is

VCE = Supply potential + forward potential of diode

+ voltage drop across R

The higher the resistance R, the quicker the current dacays but the higher the collector potential. Zener diodes are often used in series with ordinary diodes as shown in Fig. 3.46c. In this scheme, the current decays quickly after turn-off. VCE in this case is independent of current since Zener potential is fixed. This makes the determination of the rating of the maximum collector potential easy.

Figure 3.47 shows the schematic of a four-phase driver employing simple transistor switching with surge suppressing diodes. Since the output currents from a microcomputer are not enough to drive the power transistors, buffers have been used for current amplification.

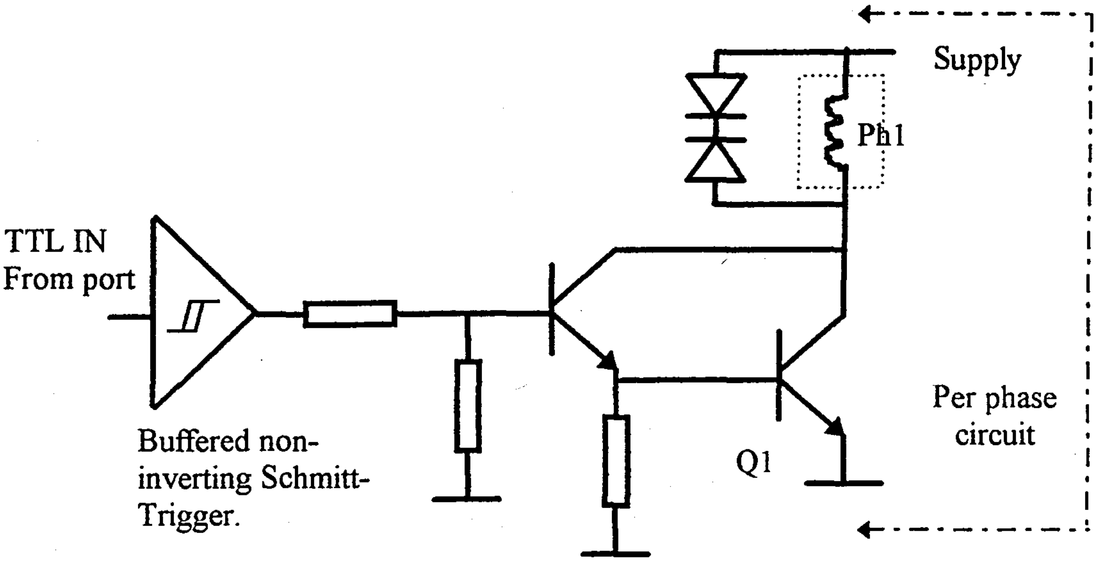


FIGURE 3.47. L/R unipolar drive with surge-suppressing diodes

Another problem with the simple transistor switching drive is that when a transistor is turned on to excite a phase, the winding inductance has a tendency to oppose the current build-up. High build-up time results in decreased torque and slow response. This parameter which is largely governed by the inductance-to-resistance ratio (L / R) of the drive circuit, becomes more critical at high switching rates.

The effect can be compensated by raising the power supply voltage and adding a resistor in series with the winding, i.e., by changing L / R to L / 4R, L / 5R, ... This way the current build-up time will be shortened.

Note that as the L / R is changed, more total power is used by the system. For L / 4R, the series resistor is selected to be 3 times the motor winding resistance with a watts rating = (current per winding)2 x R. The power supply voltage is increased to 4 times motor rated voltage so as to maintain rated current to the motor. The power supplied will thus be 4 times that of an L / R drive.

Greater increase in performance can be achieved by further increase in the power supply voltage and the value of the series resistor. However, practical limitations of power supply size and power dissipation of series resistors must be considered.

To reduce power consumption, various devices such as a bi-level power supply or a chopper may be used.

Bi-level Drive

Figure 3.48 shows two schemes of bilevel drive for one phase of stepping motor. The input pulse turns both the transistors Q1 and Q2 on and the higher voltage is applied to the winding. The diode D1 is now reverse biased to isolate the lower supply voltage from the higher voltage supply (D2 are transistor protection diodes). The current builds up quickly due to higher voltage. The time-constant of the monostable multivibrator is selected so that transistor Q1 is turned off when the winding current tends to exceed the rated current. After the high voltage source is cut-off, the diode is forward biased and the winding current is supplied from the low voltage supply. In this scheme, as the stepping rate is increased, the higher voltage is turned on for a greater percentage of time.

Chopper Drive

In this drive, the voltage applied to the motor is sliced or chopped, that gives it the name `Chopper Drive`. The basic function of a chopper (pulse-width modulation, PWM) drive is illustrated in Fig 3.49. The voltage at the winding current pick-up is compared with the reference voltage by means of an operational amplifier with a high gain. The reference voltage is the superposition of high-frequency triangular wave on a dc component.

If the dc component of the reference signal and the pick-up voltage are the same, the waveform at the output terminal of the amplifier will be a square wave as shown in Fig. 3.50a which drives the transistor Q1 in the ON / OFF mode (The diodes in the circuit are surge-suppression diodes). When the current detected is smaller than the demanded value, the ON interval of Q1 becomes larger than the OFF interval, to draw more current from the supply (Figure 3.50b).

Let us now consider the current waveform. Just after Q2 is turned on,

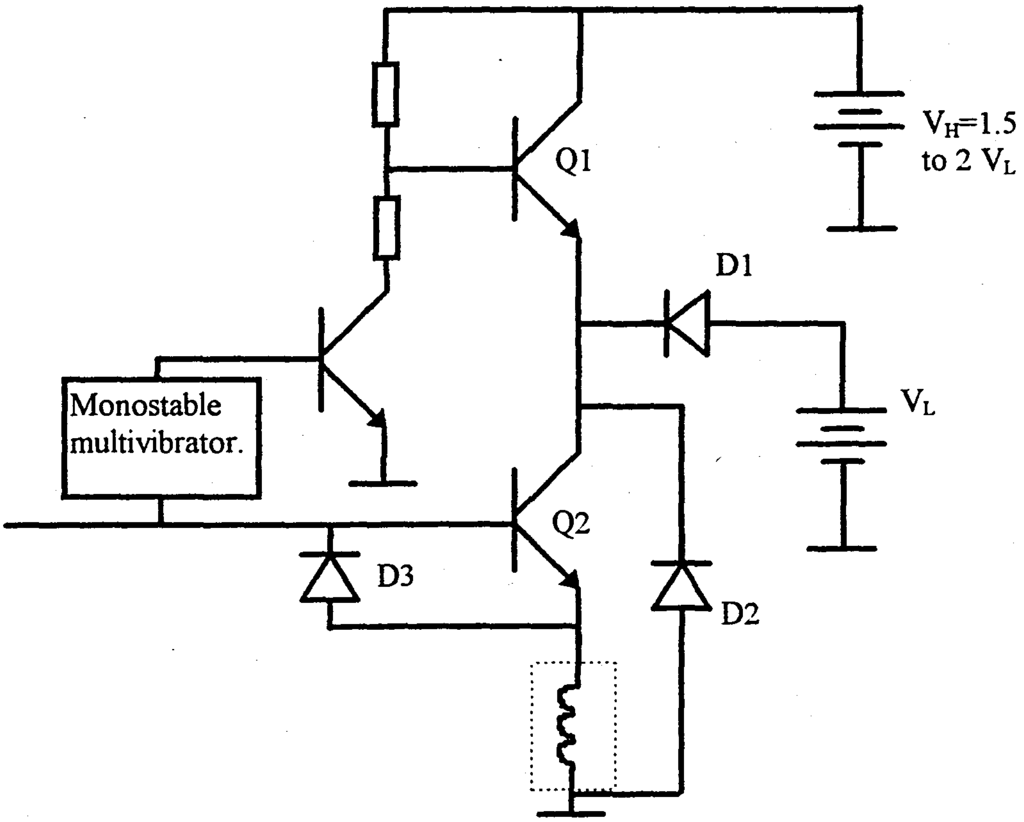


FIGURE 3.48. (a) Bi-level Unipolar drive.

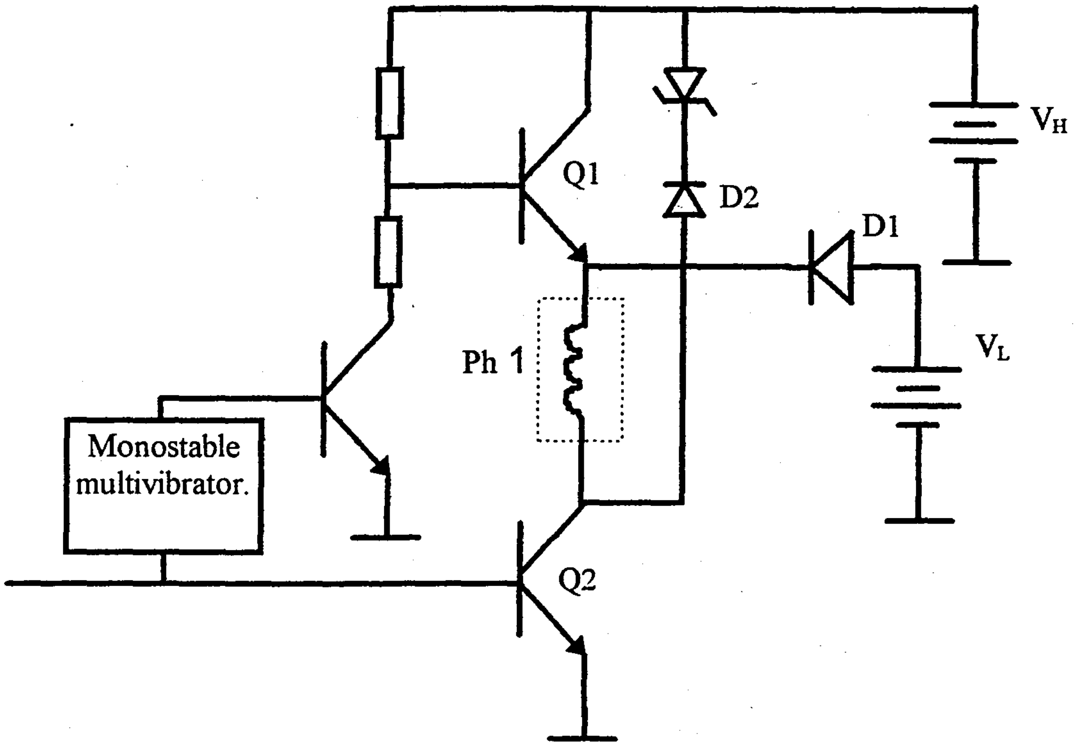


FIGURE 3.48. (b) Bi-level Unipolar drive

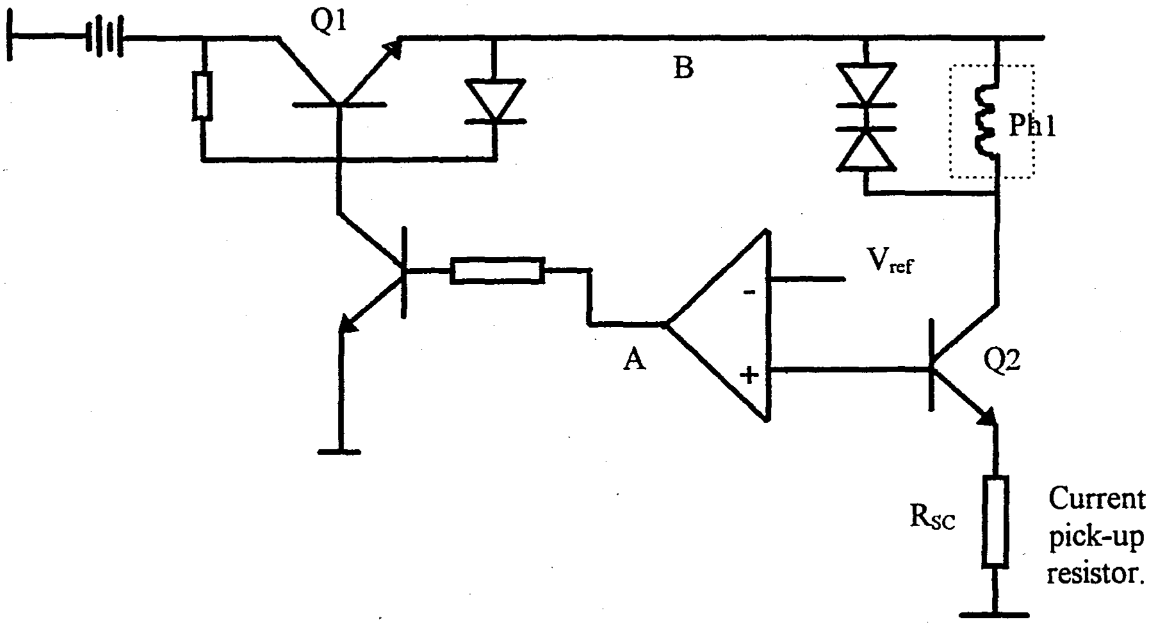


FIGURE 3.49. Chopper Unipolar drive

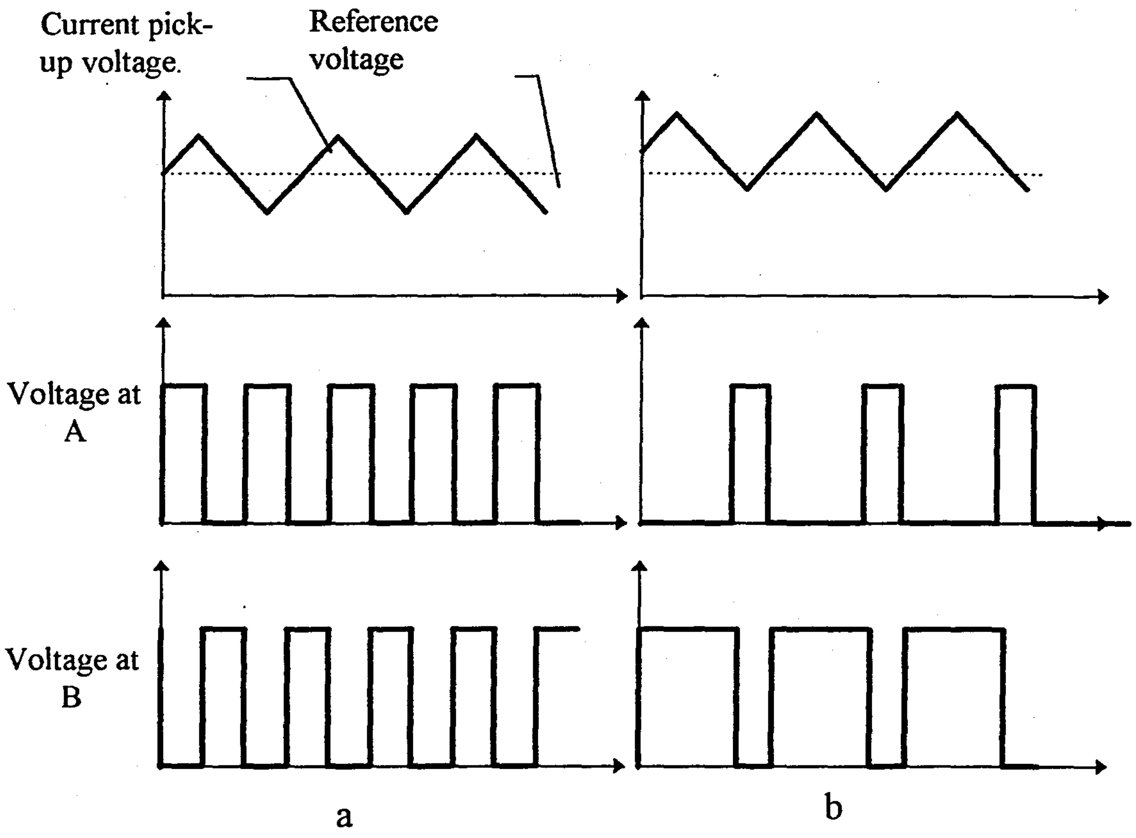


FIGURE 3.50. Pulse Width modulated waveform

the current is building up but is lower than the corresponding reference value. In this mode, Q1 is in the ON state; the high supply voltage is applied to Phase 1 winding and the current builds up quickly. When the pick-up voltage exceeds the minimum of reference voltage, the transistor Q1 is driven in the PWM mode and the winding current will become as shown in Fig. 3.51. When the transistor Q2 is turned off, the current decays quickly due to diode/zener suppresser.

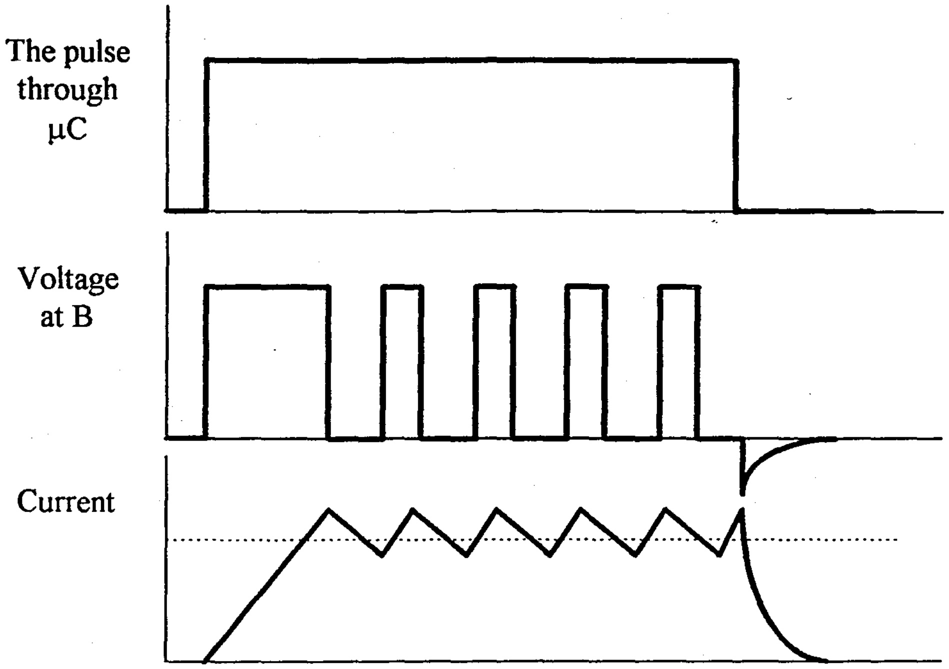


FIGURE 3.51. PWM current wave

The chopper drive offers good current build up with low loss. The disadvantages of this drive are its complexity and generation of electrical noise. Suitable rf shields to the housing of motor, drive and allied circuitry can take care of the noise problem to a certain extent.

3.3.1 Interfacing Of Stepping Motors To Microprocessors

The most profound impact of the advent of microprocessors has been found in automation. In motion control technology, the rise of stepping motor has in fact begun

with the availability of easy-to-use digital building blocks. The application of stepping motor has, however, shot up with the availability of low cost microprocessors.

In this section we discuss interfacing of stepping motors to microprocessors. For an interfacing example, I have used SRISYN stepping motor from Srijan Control Drives, Poona (India). The specifications of the stepping motor are :

Input	:	12	V DC
Current/phase	:	0.6	Amp.
Torque	:	7	Kg-cm

Figure 3.52. shows the connection for operation of the motor.

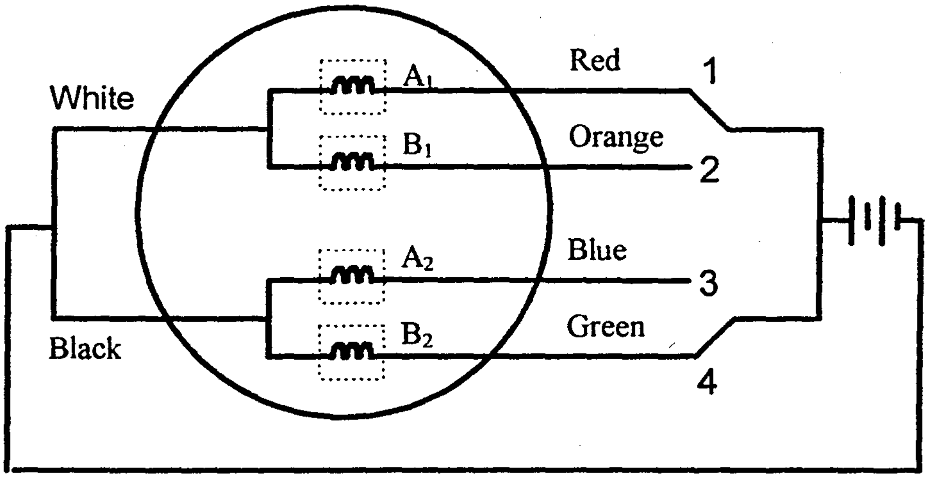


FIGURE 3.52. Schematic of a four phase hybrid motor.

Figure 3.53 shows the schematic of the four-phase hybrid motor driven by an Output Port using its four least significant bits. The excitation sequence is usually stored

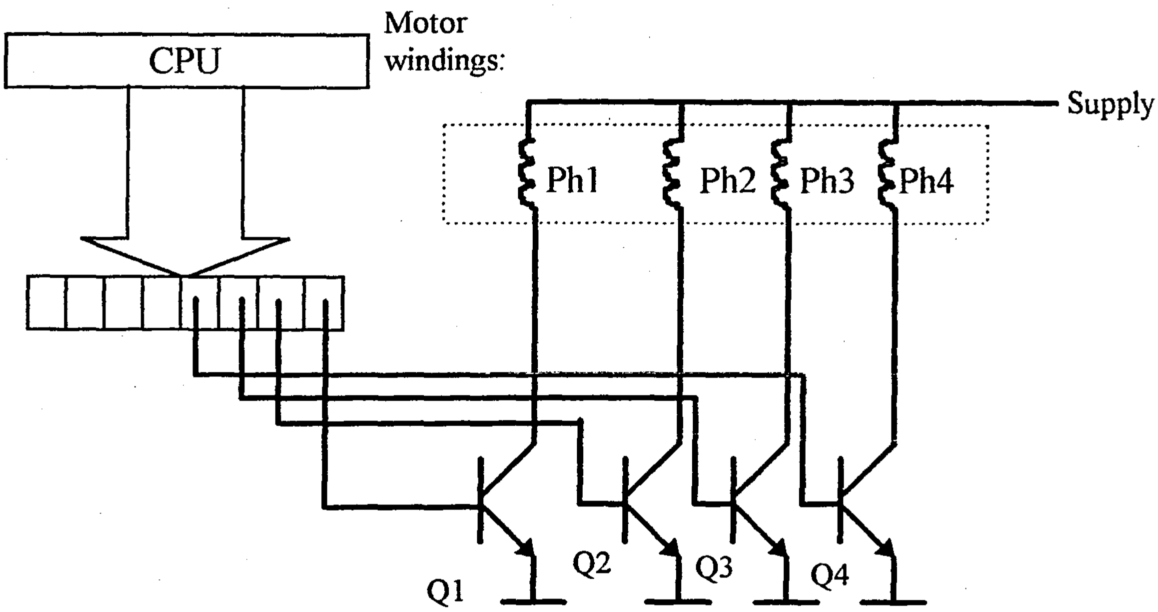


FIGURE 3.53. Drive system schematic of a four phase hybrid motor

in a table of numbers. For the system of Fig. 3.53 the excitation sequence may be of the form given in Fig. 3.54. Note that the data lines D7 to D4 are not used in sequence generation program; the most significant nibble of each entry in the chart given in Fig. 3.54 could be anything- we have assumed it to be F.

The table is circular, in the sense that the next table entry after F5 is F9. To step the motor in a clockwise direction the table is traversed from top to bottom and to step the motor in counter-clockwise direction, the table is traversed from bottom to top.

A2	B2	A1	B1	
D3	D2	D1	D0	HEX TABLE
1	0	0	1	F9
1	0	1	0	FA
0	1	1	0	F6
0	1	0	1	F5

FIGURE 3.54. Excitation sequences for a four-phase hybrid motor.

3.3.2 Simple Principles Of Step Motor Operation

The operation of step motors is best explained by means of figure 3.55 which shows a figure of an elementary motor with a single permanent magnet for a rotor and two pairs of electromagnetic poles for stator. This motor will only have four steps per revolution, but operates on exactly the same principle as one with 48 or 200 steps. The main difference is that both the rotor and stator have several pairs of magnetic poles instead of the few shown so that they can align 48 or 200 different positions.

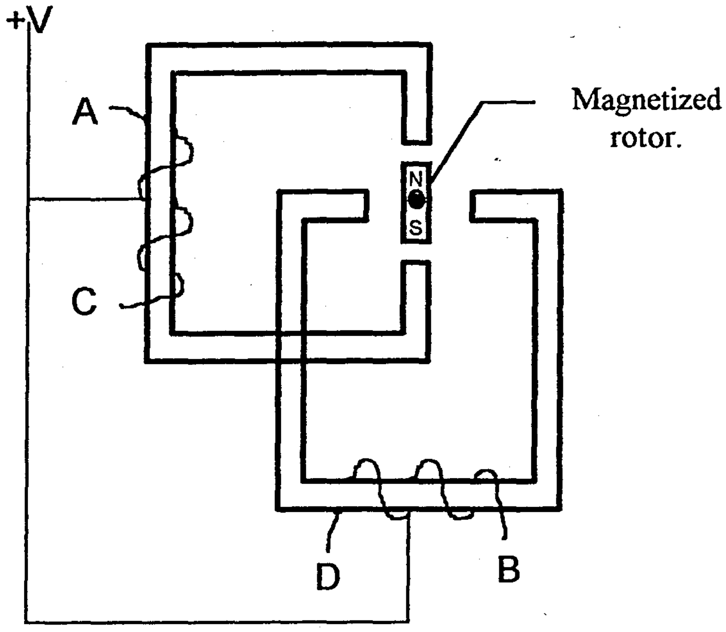


FIGURE 3.55. An elementary stepper motor.

If the rotor of the simple motor were rotated by hand, it would tend to "notch" into one of four preferred positions with the magnet aligned either way round with each pair of poles. This effect is shown by ordinary stepping motors which have a very "notchy" feel when rotated. Counting the notches gives the number of steps per revolution in Full Step mode.

To energize the simple motor, terminals A, C, D, and B are connected individually or in combinations to the negative terminal of the motor power supply. If terminal A is connected, then current flowing from the motor supply through the winding magnetizes the associated iron core in one direction. Connecting terminal C instead of A magnetizes the core in the opposite direction.

If terminals A and C are connected to negative together, then the two currents magnetizing effects oppose each other and the core is not magnetized at all. The same effects apply when connecting points B and/or D to negative, the magnetization of the associated core follows a similar pattern. The two cores with their windings operate entirely separately from each other.

3.3.3 Wave Drive

The simplest way to drive the motor is called wave drive. Figure 3.56 shows the stator magnetizing sequence and corresponding rotor positions, and table 3.1 shows which terminals are connected to negative for each step. Ignore for now the other information in the tables which refer to other connections of the driver i.e. The relevant information is in the columns marked A to D and rows 1 to 4.

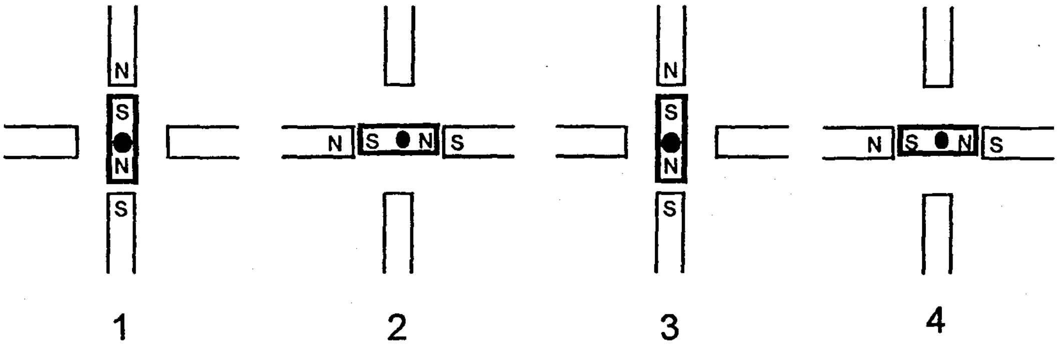


FIGURE 3.56. Wave drive magnetizing sequence

TABLE 3.1. Wave drive sequence.
Step 1234 for one direction, 4321 for the other.

Step	A	B	C	D
PRO	ON	OFF	OFF	OFF
1	ON	OFF	OFF	OFF
2	OFF	ON	OFF	OFF
3	OFF	OFF	ON	OFF
4	OFF	OFF	OFF	ON

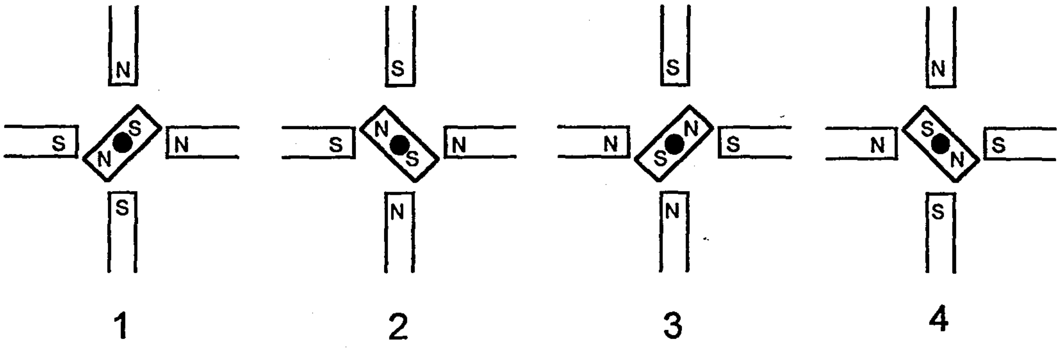


FIGURE 3.57. Two phase or Full Step drive magnetizing sequence

TABLE 3.2. Two phase drive sequence. Step 1234 for one direction, 4321 for the other.

Step	A	B	C	D
PRO	ON	OFF	OFF	ON
1	ON	OFF	OFF	ON
2	ON	ON	OFF	OFF
3	OFF	ON	ON	OFF
4	OFF	OFF	ON	ON

In Wave Drive, as each winding is energized, the magnetic rotor moves to align with the electromagnetized poles. By switching in the correct sequence, the magnetic rotor moves to each position in turn, rotating fully after four steps. By energizing the windings in the reverse sequence, the rotor can be made to revolve in the opposite direction.

Wave drive is the simplest method to describe but not a very efficient way to run a stepper motor. This is because only one winding is used at a time and so only half of the winding wire and space, and the stator core material is utilized. To improve upon this , Two Phase or Full step drive is used.

3.3.4 Full Step

Full step (or two phase) drive involves a similar four step sequence to Wave Drive but two windings are energized at each step. Fig 3.57 and Table 3.2 show the rotor positions and the winding energization patterns.

Note that the rotor aligns with the stronger magnetic field between the two sets of poles. The torque is increased substantially over Wave Drive as two windings now provide the magnetic field instead of one.

3.3.5 Half Step

A third method of operation is Half Step mode. This is a combination of the two previous ones and takes advantage of the rotor's ability to align alternately with the stator poles and between them, to double the number of steps available from the motor. Fig 3.58 and Table 3.3 show the rotor positions and winding energization patterns.

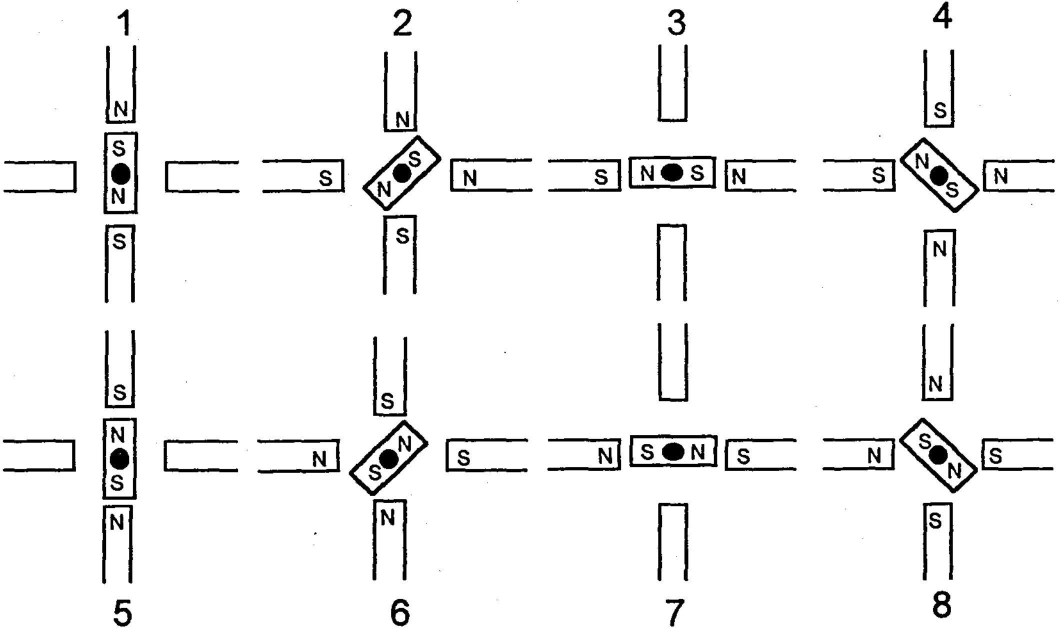


FIGURE 3.58. Half step mode magnetizing sequence.

TABLE 3.3. Half step drive sequence. Step 123..8 for one direction, 876..1 for the other.

Step	A	B	C	D
PRO	ON	OFF	OFF	ON
1	ON	OFF	OFF	ON
2	ON	ON	OFF	OFF
3	OFF	ON	ON	OFF
4	OFF	OFF	ON	ON
5	OFF	OFF	ON	OFF
6	OFF	OFF	ON	ON
7	OFF	OFF	OFF	ON
8	ON	OFF	OFF	ON

In this mode, the torque varies up and down with each half step as the motor moves alternatively between Wave Drive and Full Step modes. This would seem to be a disadvantage but it is not a serious one. As the motor does not have to move so far with each step the varying torque does not reduce the performance significantly, and the increased smoothness of running due to twice the number of steps being used gives big reductions in noise and vibration levels.

3.3.6 Speed Considerations

There is no lower speed limit to step motor operation. One step every week is quite acceptable. At the opposite end of the spectrum however, the maximum speed is limited by many things.

The main limit is determined by the inductance of the windings which reduces the rate at which the winding current can rise. Above a certain speed the winding current reduces until the torque becomes too small to be any use. This limit can be pushed up effectively by increasing the motor drive voltage at high speeds, but there is still a limit , and stepping motors can only be considered as low speed drives.

Switching of the windings is usually carried out electronically by power transistors. To use mechanical switches such as relays is impractical because of the operating speed required. In my project robot , a special i.c. is used which contain 8 power transistors and all the logic circuitry to switch them on and off according to the given signals by the processor.

The i.c. has several logic (TTL compatible) inputs and two enable inputs for the two windings on our stepper motor. One of them is the SGS Thompson's L298 power motor driver chip. The chip is capable of driving two DC motors or a single stepper motor, with currents up to four amperes. The technical specifications and datasheets of this chip can be found in Appendix A.

3.4 Motor Speed Control For Robots

In most DC motor speed control circuits, a voltage proportional to the desired speed and an error, or correction signal is obtained. The error signal is amplified and used to adjust the speed of the motor in such a way as to reduce the error signal to a near-zero value. Provided that the circuit has been properly designed, the motor will run at a speed close that desired.

The voltage supplied to the motor will determine the output power or torque, as well as the speed, so that at low speeds, very little torque is produced. The normal method of overcoming this problem is to supply the motor with a duty cycle proportional to the error voltage, so that the full torque is produced at low speeds.

There are a number of ways that can be used to derive a voltage proportional to motor speed. One method which has been favored in the past, involves measuring the back-emf of the motor. This can lead to problems, as both the input and the output are obtained from the same point, namely the motor.

It is possible to obtain a voltage proportional to motor speed by using an opto-electronic tachometer system. In my system, the direction of the robot is checked and corrected in every scan of the sensory head so small changes in the speed of the two DC motors on the sides of the robot did not effect the working of the robot. in my robot, DC motor speed control is only used on the remote control mode of the robot when the control is completely on a remote transmitter which transmits IR signals carrying the commands for going, turning etc. The commander selects the speed as slow or fast by pressing one of the two speed buttons on its IR transmitter (in fact an ordinary TV remotecontroller). Then the main processor applies full driving or a %50 duty cycle pulse width modulation to the DC motors to obtain a speed control.

4. CONCLUSION

The robot was capable of moving without any collision until a goal is inserted to the system. Since the ultrasonic receiver must both listen the goal and measure the distance of the obstacles around by time multiplexing, occurred new problems. The increase in ultrasonic sources means increase in ultrasonic echoes.

To provide absolute obstacle detection, a laser scanning system must be used. So that the goal signal can not make any distortions on this system. Another shortcoming observed was that sometimes goal echoes from an obstacle were stronger than the goal itself. If so, the robot moves towards that obstacle and collides. It is important that to achieve the required goal, the assembly must satisfy certain constraints such as geometry and stability. If any of the components of the system tend to move away from its intended location, the system is considered unstable. As friction plays an important role in maintaining stable equilibrium of the components. The stability of the system was also dependent to the mass and inertia.

During the assembly of the robot, we tried to make it as light as possible. And so the power sources were chosen to be both light and powerful enough. After that the stability problem has been ignored. To develop a model of the robot was extremely difficult, since it consists of several components and interconnections. There are computational difficulties in order to make a realistic stability analysis.

5. A SHORT OVERVIEW OF PATH FINDING, COLLISION AVOIDANCE ALGORITHMS

Obstacle avoidance is one of the main problems concerning mobile robot applications. Given the goal position the robot should be able to find a path to reach it without colliding with unexpected obstacles. Mobile robots must obey some sort of collision avoidance algorithms. Old primitive algorithms can only detect an obstacle and stop the robot without making some quantitative measurements concerning the type and location of the obstacle.

Although usually partial information is available, the most general form of the problem is in complete ignorance about the environment. Most of the time the problems of obstacle detection and obstacle avoidance are treated separately. However controlling a robotic system requires both obstacle detection and avoidance.

It has been shown that a more advanced approach to robot programming is task-level programming, where the operator does not specify a sequence of manipulator motions and functions. thus the problem of automatic robot programming can be formulated as follows: Given a description of the initial and the goal situations along with a complete model of the robot world, find a robot motion allowing to reach the goal situation without generating any collision between the robot and the objects belonging to the robot workspace.

The find path problem was first addressed by Shannon in the late 1940s when he built an electronic mouse moving in a checkboard-like maze. Later the same problem was studied when researchers tried to move a mobile robot with a nearly circular base (to avoid the rotation problem) in a plane. To simplify the problem, often all relevant obstacles are projected onto a plane and a 2D object (the robot itself) must find a path between two positions, even though this method might not find all possible paths. Thus for a realistic robot model, rotation and height information must be included. Another example for the findpath problem is the routing of VLSI circuits, which can be seen as finding a path in a plane.

For several connected bodies, the motion is more complicated than that is of independent bodies, as the motion of one joint usually results in the displacement of another joint. For articulated robots this problem is hard to solve in Euclidian space, as the displacement is highly non-linear. In the motion coordinating problem, not only the path, but also the trajectory (time history along path) must be planned.

In all preceding cases the obstacles were assumed to be fixed and not allowed to move while the robot is moving. This assumption is valid only for areas like work cells, but for mobile robots this is no longer the case. To complicate the problem the acceleration and velocity bounds of the robot have to be considered. This problem is named after the problem of steering a spaceship through an asteroid field.

Appendix A:

```
;       *******************************************************
;       The assembly listing  of  the program  for  Step motor
;       control.  Designed  for  controlling  the  rotation of
;       ultrasonic ranging sensor pairs in robotic applications.
;       Written for SGS Thompson ST 62xxx series microcontrollers.
;       *******************************************************
        .romsize      4
        .vers   "st6225"
         .org   0080h
reset:
        ldi     wd,0ffh
        ret
        ret
        ret
        ret
        ret
        ret
        reti
        ldi     ddra,0ffh
        ldi     ora,0ffh
        ldi     dra,01010101b
        ldi     ddrb,0ffh
        ldi     orb,0ffh
        ldi     drb,0ffh
        ldi     ddrc,0ffh
        ldi     orc,0ffh
        ldi     drc,0ffh
        ldi     wd,0ffh
main    ldi     sayac,120
```

```
main1  ldi    wd,0ffh
       ldi    ddra,0ffh
       ldi    ora,0ffh
       ld     a,sayac
       ld     drb,a
       call   yon1
       call   bekler
       dec    sayac
       jrz    gerber
       jp     main1
gerber ldi    sayac,120
main2  ldi    wd,0ffh
       ldi    ddra,0ffh
       ldi    ora,0ffh
       ld     a,sayac
       ld     drb,a
       call   yon2
       call   bekler
       dec    sayac
       jrz    gerber2
       jp     main2
gerber2 jp    main
bekler
       ldi    gecik,0ffh
       call   dly
       ldi    gecik,0ffh
       call   dly
       ldi    gecik,0ffh
       call   dly
       ldi    gecik,0ffh
       call   dly
       ldi    gecik,0ffh
```

```
        call   dly
        ldi    gecik,0ffh
        call   dly
        ret
; Delay subroutine.
dly     ldi    wd,0ffh
        dec    gecik
        jrnz   dly
        ret
yon1
        ld     a,pamask
        cpi    a,01011100b
        jrnz   nm1
        ldi    pamask,11000101b
        ld     a,pamask
        ld     dra,a
        ret
nm1     cpi    a,11000101b
        jrnz   nm2
        ldi    pamask,01010011b
          ld     a,pamask
          ld     dra,a
          ret
nm2     cpi    a,01010011b
          jrnz   nm3
          ldi    pamask,00110101b
          ld     a,pamask
          ld     dra,a
          ret
nm3     cpi    a,00110101b
          jrnz   nm4
          ldi    pamask,01011100b
```

```
        ld     a,pamask
        ld     dra,a
        ret
nm4     ldi    pamask,01011100b
        ld     a,pamask
        ld     dra,a
        ret
yon2
        ld     a,pamask
        cpi    a,00110101b
        jrnz   nm12
        ldi    pamask,01010011b
        ld     a,pamask
        ld     dra,a
        ret
nm12  cpi    a,01010011b
      jrnz   nm22
      ldi    pamask,11000101b
      ld     a,pamask
      ld     dra,a
      ret
nm22  cpi    a,11000101b
      jrnz   nm32
      ldi    pamask,01011100b
      ld     a,pamask
      ld     dra,a
      ret
nm32  cpi    a,01011100b
      jrnz   nm42
      ldi    pamask,00110101b
      ld     a,pamask
      ld     dra,a
```

```
        ret
nm42    ldi     pamask,00110101b
        ld      a,pamask
        ld      dra,a
        ret
nmi     ldi     sayac,120
        reti
portbc  reti
        .org    0ff0h       ; exception vectors
        jp      portbc      ; adc vector
        jp      portbc      ; timer vector
        jp      portbc      ; port b & c vector
        jp      portbc      ; port a vectoru
        .org    0ffch
        jp      nmi         ; NMI vector
        jp      reset       ; RESET vector
        .end
```

Appendix B:

```
;       **************************************************
;       The assembly listing  of  the program  for  ultrasonic
;       ranging and goal detecting algorithm. For SGS
;       Thompson ST 62xxx series microcontrollers.
;       **************************************************
        .romsize        4
        .vers    "st6225"
.org   0080h
reset:
        ldi     sayac2,10     ; For the goal detecting algorithm.
cirk   ret
        ret
        ret
        ret
        ret
        ret
        ld      a,conti
        cpi      a,38
        jrz     ikinci
        ldi     conti,38
        ldi     key,10          ; How many times to look for goal after
                              ; the initialization.
ikinci
        ldi     ddra,0ffh       ; Port A is Push-Pull output.
        ldi     ora,0ffh
        ldi     ddrc,11110000b   ; Port c is Push-Pull Output
                                 ; (PC4 is the output pin of
        ldi     orc,11110000b    ; 40 khz ultrasonic signals)
        clr     drc
        ldi     tmr,128
```

```
volt  ldi   ddrb,11001011b
      ldi   orb,11101011b
      ld    a,pbmask
      andi   a,01100011b    ; pb0`is preserved.
      ld    pbmask,a
      ldi   tscr,00111110b
      reti
birdaha ldi    wd,0ffh
      call   ultra1          ; The sub routine which looks for goal.
      call   ultra           ;The subroutine for measuring distance.
      set    0,pbmask
      ld     a,pbmask
      ld     drb,a           ; Value is on the port signal
      call   bekler          ; Delay time for the data to be read by
                             ; the main data processor.
      res    0,pbmask
      ld     a,pbmask
      ld     drb,a           ; "Value is on the port" signal is terminated.
      jp     birdaha
ultra1  ldi    sayac2,20
      dec    key
      jrnz   ultra15
      ldi    key,10
      ret
ultra15 ld     a,sayac2
      jrnz   ultra15
echo   ldi    sayac2,5
      call   cirla
      set    1,pbmask
      ld     a,pbmask
      ld     drb,a          ; "GOAL is present" signal.
      call   bekler
```

```
        res     1,pbmask
        ld      a,pbmask
        ld      drb,a
        call    bekler
        call    bekler
        ret
ultra   ldi     sayac2,5
        set     4,drc
        ldi     wd,0ffh
        res     4,drc
        nop
        set     4,drc
        ldi     wd,0ffh
        res     4,drc
        nop
        set     4,drc
        ldi     wd,0ffh
        res     4,drc
        nop
        set     4,drc
        ldi     wd,0ffh
        res     4,drc
        nop
        set     4,drc
        ldi     wd,0ffh
        res     4,drc
        ldi     tcr,0ffh
        ldi     sayar,255
        ldi     dist,255        ; Distance data
basari
basari2 ld      a,sayac2
        jrnz    basari
```

```
echovar ldi    sayac2,5
       ldi     a,255  ; The transmitted signal has arrived to the goal and
                      ; the echo came back. The data on the TCR register
                      ; is the distance data
                      ; But the data is in the reverse order.
       sub     a,tcr  ; After complemented, the distance data is sent back
                      ; on accumalator a.
       cpi     a,25
       jrc     r1kuc
       jp      r1buy
r1kuc  ldi     dra,00000010b
       ret
r1buy  cpi     a,32
       jrc     r2kuc
       jp      r2buy
r2kuc  ldi     dra,00000001b
       ret
r2buy  ldi     dra,0
       ret
; Delay subroutine
bekler ldi     gecik,0ffh
       call    dly
       ldi     gecik,0ffh
       call    dly
       ldi     gecik,0ffh
       call    dly
       ldi     gecik,0ffh
       call    dly
       ldi     gecik,0ffh
       call    dly
       ldi     gecik,0ffh
       call    dly
```

```
        ldi    gecik,0ffh
        call   dly
        ldi    gecik,0ffh
        call   dly
        ldi    gecik,0ffh
        call   dly
        ldi    gecik,0ffh
        call   dly
        ldi    gecik,0ffh
        call   dly
        ldi    gecik,0ffh
        call   dly
        ldi    gecik,0ffh
        call   dly
        ldi    gecik,0ffh
        call   dly
        ldi    gecik,0ffh
        call   dly
        ldi    gecik,0ffh
        call   dly
        ldi    gecik,0ffh
        call   dly
        ldi    gecik,0ffh
        call   dly
        ret
;       Sound generation subroutine is run when a goal signal is detected.
cirla
        ldi    v,0ffh
hihi    ldi    wd,0ffh
        ld     a,pbmask
        set    3,a
        ld     drb,a
```

```
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
ld    a,pbmask
res    3,a
ld    drb,a
nop
nop
nop
nop
nop
nop
nop
```

```
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        dec    v
        jrz    done
        jp     hihi
done    ret
dly     ldi    wd,0ffh
        dec    gecik
        jrnz   dly
        ret
nmi     dec    sayac2
        reti
timer   ld     intdep1,a
        ld     a,tcr
        jrz    baslat
        res    7,tscr
baslat  ld     a,intdep1
        reti
portbc  dec    sayac2
```

```
reti
.org   0ff0h                          ; exception vectors
jp     nmi                           ; adc vector
jp     timer                         ; timer vector
jp     portbc                         ; port b & c vector
jp     nmi                           ; port a vector
.org   0ffch
jp     nmi                           ; NMI vector
jp     reset                         ; RESET vector
.end
```

Appendix C:

```
;       ******************************************************
;       The assembly listing  of  the program  for  DC motor servo
;       position control in robotic applications.
;       Written for SGS Thompson ST 62xxx series microcontrollers.
;       ******************************************************
        .romsize     4
        .vers   "st6225"


        .org    0080h
reset:
        ldi     sayac2,5
cirk    ret
        ret
        ret
        ret
        ret
        ret
        ldi     ddra,0ffh       ; PORT A PUSH-PULL output
        ldi     ora,0ffh
        clr     dra
        ldi     ddrb,00001001b
        ldi     orb,00001001b
        clr     drb
        ldi     ddrc,11110000b  ; PORT C PUSH-PULL output
        ldi     orc,11110000b
        clr     drc
        clr     pbmask
        ldi     tmr,128
volt
        ldi     ddrb,11001001b
```

```
     ldi    orb,11101001b
     ldi    drb,01100000b
     ldi  pbmask,01100000b
     ldi    tscr,00111110b
     reti
     ldi    altsin,2
     ldi    ustsin,6
birdaha ldi     wd,0ffh
     call   ultra
     set    0,pbmask
     ld     a,pbmask
     ld     drb,a
     call   bekler
     res    0,pbmask
     ld     a,pbmask
     ld     drb,a
     jp     birdaha
     res    4,ddrb
     set    4,orb
     set    4,pbmask
     res    0,pbmask
     ld     a,pbmask
     ld     drb,a
     ldi    adcr,00110000b      ; start conversion .
bikl   jrr    6,adcr,bikl
     ld     a,addr
     ld     tmr,a
     cp     a,altsin
     jrc    zip1
     cp     a,ustsin
     jrnc   zip2
     jp     durdur
```

```
zip2  set   7,pbmask
      set   6,pbmask
      ld    a,pbmask
      ld    drb,a
      jp    zip3
zip1  res   7,pbmask
      res   6,pbmask
      ld    a,pbmask
      ld    drb,a
zip3  ldi   wd,0ffh
      jp    birdaha
durdur set  6,pbmask
      res   7,pbmask
      ld    a,pbmask
      ld    drb,a
      ld    a,altsin        ; Put the angle value to the databus
      ld    dra,a
      set   0,pbmask
      ld    a,pbmask
      ld    drb,a           ; 'The data is on the bus' signal
      call  ultra
      inc   altsin
      inc   ustsin
      ld    a,ustsin
      cpi   a,250
      jrc   giiit
      ldi   altsin,2
      ldi   ustsin,6
      clr   dra
giiit jp    birdaha
      call  bekler
      res   4,ddrb
```

```
        set    4,orb
        set    4,pbmask
        ld     a,pbmask
        ld     drb,a
        ldi    adcr,00110000b      ; start conversion .
bekl    jrr    6,adcr,bekl
        ld     a,addr
        ld     tmr,a
dirkpi  inc    altsin
        inc    altsin
        inc    altsin
        inc    altsin
        inc    ustsin
        inc    ustsin
        inc    ustsin
        inc    ustsin
        ld     a,ustsin
        cpi    a,250
        jrc    devam
        ldi    wd,1
devam
        ldi    ddrb,11000001b
        ldi    orb,11100001b
        ldi    drb,01100000b
        ldi pbmask,01100000b

        jp     birdaha
ultra
        set    4,drc
        ldi    wd,0ffh
        res    4,drc
        nop
```

```
        set     4,drc
        ldi     wd,0ffh
        res     4,drc
        nop
        set     4,drc
        ldi     wd,0ffh
        res     4,drc
        nop
        set     4,drc
        ldi     wd,0ffh
        res     4,drc
        nop
        set     4,drc
        ldi     wd,0ffh
        res     4,drc
        ldi     tcr,0ffh
        ldi     sayar,255
        ldi     dist,255
basari
basari2 ld      a,sayac2
        jrnz    basari
echovar ldi     sayac2,5
        ldi     a,255
        sub     a,tcr
        ld      dra,a
        ld      dist,a
        ret
;               BEKLER
bekler  ldi     gecik,0ffh
        call    dly
        ldi     gecik,0ffh
        call    dly
```

```
ldi    gecik,0ffh
call   dly
ldi    gecik,0ffh
call   dly
ldi    gecik,0ffh
call   dly
ldi    gecik,0ffh
call   dly
ldi    gecik,0ffh
call   dly
ldi    gecik,0ffh
call   dly
ldi    gecik,0ffh
call   dly
ldi    gecik,0ffh
call   dly
ldi    gecik,0ffh
call   dly
ldi    gecik,0ffh
call   dly
ldi    gecik,0ffh
call   dly
ldi    gecik,0ffh
call   dly
ldi    gecik,0ffh
call   dly
ldi    gecik,0ffh
call   dly
ldi    gecik,0ffh
call   dly
ldi    gecik,0ffh
call   dly
ldi    gecik,0ffh
call   dly
```

```
        ret
; sound generation program
cirla
        ldi     v,0ffh
hihi    ldi     wd,0ffh
        ld      a,pbmask
        set     3,a
        ld      drb,a
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        ld      a,pbmask
        res     3,a
```

```
      ld     drb,a
      nop
      nop
      nop
      nop
      nop
      nop
      nop
      nop
      nop
      nop
      nop
      nop
      nop
      nop
      nop
      nop
      nop
      nop
      nop
      nop
      nop
      nop
      dec    v
      jrz    done
      jp     hihi
done  ret
dly   ldi    wd,0ffh
      dec    gecik
      jrnz   dly
      ret
nmi   dec    sayac2
```

```
        reti
timer  ld     intdep1,a
        ld     a,tcr
        jrz    baslat
        res    7,tscr
baslat ld     a,intdep1
        reti
portbc dec    sayac2
        reti
        .org   0ff0h                   ; exception vectors
        jp     nmi                    ; adc vector
        jp     timer                  ; timer vector
        jp     portbc                  ; port b & c vector
        jp     nmi                    ; port a vector
        .org   0ffch
        jp     nmi          ; NMI vector
        jp     reset        ; RESET vector
        .end
```

Appendix D:

```
;          Goal signal generating algorithm
;          written for my mobile robots
                    .romsize        4
                    .vers    "st6225"
           .org     0080h
reset:
cirk       ret
           ret
           ret
           ret
           ret
           ret
           ldi      ddra,0ffh
           ldi      ora,0ffh
           ldi      ddrc,11110000b
           ldi      orc,11110000b
           clr      drc
           reti
birdaha ldi      wd,0ffh
           call     ultra
           call     bekler
           jp       birdaha
ultra      ldi      dra,0ffh
           ldi      wd,0ffh
           ldi      dra,0
           nop
           ldi      dra,0ffh
           ldi      wd,0ffh
           ldi      dra,0
           nop
```

```
ldi     dra,0ffh
ldi     wd,0ffh
ldi     dra,0
nop
ldi     dra,0ffh
ldi     wd,0ffh
ldi     dra,0
nop
ldi     dra,0ffh
ldi     wd,0ffh
ldi     dra,0
nop
ldi     dra,0ffh
ldi     wd,0ffh
ldi     dra,0
nop
ldi     dra,0ffh
ldi     wd,0ffh
ldi     dra,0
nop
ldi     dra,0ffh
ldi     wd,0ffh
ldi     dra,0
nop
ldi     dra,0ffh
ldi     wd,0ffh
ldi     dra,0
nop
ldi     dra,0ffh
ldi     wd,0ffh
ldi     dra,0
nop
```

```
            ldi      dra,0ffh
      ldi   wd,0ffh
     ldi    dra,0
nop
ldi         dra,0ffh
ldi         wd,0ffh
ldi         dra,0
nop
ldi         dra,0ffh
ldi         wd,0ffh
ldi         dra,0
nop
ldi         dra,0ffh
ldi         wd,0ffh
ldi         dra,0
nop
ldi         dra,0ffh
ldi         wd,0ffh
ldi         dra,0
nop
ldi         dra,0ffh
ldi         wd,0ffh
ldi         dra,0
nop
ldi         dra,0ffh
ldi         wd,0ffh
ldi         dra,0
nop
ldi         dra,0ffh□
ldi         wd,0ffh
ldi         dra,0
nop
```

```
ldi     dra,0ffh
ldi     wd,0ffh
ldi     dra,0
nop
ldi     dra,0ffh
ldi     wd,0ffh
ldi     dra,0
nop
ldi     dra,0ffh
ldi     wd,0ffh
ldi     dra,0
nop
ldi     dra,0ffh
ldi     wd,0ffh
ldi     dra,0
nop
ldi     dra,0ffh□
ldi     wd,0ffh
ldi     dra,0
nop
ldi     dra,0ffh
ldi     wd,0ffh
        ldi     dra,0
        nop
        ldi     dra,0ffh
        ldi     wd,0ffh
        ldi     dra,0
        nop
        ret
;
;¦               BEKLER               ¦
;
```

```
bekler ldi      gecik,0ffh
       call     dly
       ldi      gecik,0ffh
       call     dly
       ldi      gecik,0ffh
       call     dly
       ldi      gecik,0ffh
       call     dly
       ldi      gecik,0ffh
       call     dly
       ldi      gecik,0ffh
       call     dly
       ldi      gecik,0ffh
       call     dly
       ldi      gecik,0ffh
       call     dly
       ldi      gecik,0ffh
       call     dly
       ldi      gecik,0ffh
       call     dly
       ldi      gecik,0ffh
       call     dly
       ldi      gecik,0ffh
       call     dly
       ldi      gecik,0ffh
       call     dly
       ldi      gecik,0ffh
       call     dly
       ldi      gecik,0ffh
       call     dly
       ldi      gecik,0ffh
       call     dly
```

```
        ldi     gecik,0ffh
        call    dly
        ldi     gecik,0ffh
        call    dly
        ret
;           sound generation program
cirla
        ldi     v,0ffh
 hihi   ldi     wd,0ffh
        ld      a,pbmask
        set     3,a
        ld      drb,a
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
```

```
        nop
        nop
        ld      a,pbmask
        res     3,a
        ld      drb,a
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        dec     v
        jrz     done
        jp      hihi
done    ret
dly     ldi     wd,0ffh
```

```
        ret
; sound generation program
cirla
        ldi     v,0ffh
hihi    ldi     wd,0ffh
        ld      a,pbmask
        set     3,a
        ld      drb,a
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        ld      a,pbmask
        res     3,a
```

REFERENCES NOT CITED

Gopal, M. *Digital Control Engineering* ISBN 0-470-20868-6, 1988.

Intel, *Embedded Control Applications Data book,* 1988.

Seymour, C., "Build your own robot," *Electronics Today International,* p.61 March 1982.

Seymour, C., "Robot chassis types," *Electronics Today International,* p.82 August 1982.

Seymour, C., "An infra red eye for mobile robots," *Electronics Today International,* p.84 January 1982.

Cullinan, D., "Brush type DC motors and their diving techniques," *Electronics Today International,* p.34 May 1982.

Cullinan, D., "Step motors and their interfacing to microprocessors," *Electronics Today International,* p.66 June 1982.

Cullinan, D., "Special driver integrated circuits suitable for robotics," *Electronics Today International,* p.69 June 1982.

Juarez, T., "Robot vision," *Electronics Today International,* p.59 July 1982.

Donovan, L., "Motor power requirements of a mobile robot," *Electronics Today International,* p.26 September 1982.

Tomac, J., "Robotic wheels," *Elektor Electronics,* p.64 February 1996.

Elfes, A., "A Sonar -Based Mapping and Navigation system," *IEEE Int. Conf. Robotics and Automation*, pp.1151-1157, 1986

Elfes, A., "Sonar -Based Mapping and Navigation," *IEEE J. Robotics Automat, RA-3*, Vol. 3, 249, 1987.

Cox, J. and Blanche Ingemar, "An Autonomous Robot Vehicle For Structured Environments," *IEEE J. Robotics and Automation*. p. 978, 1988.

Borenstein and Koren, "Obstacle Avoidance With Ultrasonic Sensors," *IEEE J. Robotics Automat*, RA-4, (2), 213 1988.

Krogh, B.H. and C. E. Thorpe, "Integrated path planning and dynamic steering control for autonomous vehicles," *IEEE Int. Conf. Robotics Automat*, San Francisco, CA, 1664, Apr. 7, 1986.

Walnut, C., *Robotics And Industrial Electronics*, Book 2 of Heathkit Zenith Educational systems 1983.

Robillard, M.J., *Hero 1: Advanced programming and interfacing*, Howard W. Sams & Co. Inc. 4300 West 62nd St. Indianapolis, Indiana 46268 USA.

Haralick, R. M. and L. G. Saphiro, *Computer and Robot Vision*, Addison Wesley Pub. Co., v.1 ISBN 0-201-10877-1, 1992.

Klafter ,R. D. and T. A. Chmielewski, *Robotic Engineering*, Perentice Hall Internal editions , ISBN0-13-782053-4 1989.

Snyder, W. E., *Industrial Robots and Control*, Perentice Hall, Inc., Anglewood Cliffs, New Jersey 07632., ISBN 0-13-463159-5, 1985.