DESIGN OF NAVIGATION ALGORITHM FOR F_BOT FOR PATROLLING USING FUZZY LOGIC

Oleh

Ahmad Ashaari Bin Yusof

Disertasi ini dikemukakan kepada

UNIVERSITI SAINS MALAYSIA

Sebagai memenuhi sebahagian daripada syarat keperluan
untuk ijazah dengan kepujian

SARJANA MUDA KEJURUTERAAN (KEJURUTERAAN MEKATRONIK)

Pusat Pengajian kejuruteraan
Elektrik dan Elektronik
Universiti Sains Malaysia

March 2006

Abstract

The main objective of this project is to build a mobile robot that would be able to navigate from one position to the other assigned position. It is also able to determine the shortest path towards the goal position without hitting any obstacle along the path. This mobile robot uses the Fuzzy Logic controller to navigate the robot in varying environments to a specified position. The implementation has been carried out using inexpensive components and tools. As the mobile robot is using the fuzzy logic controller to control its movement, it is named as F_Bot The F_Bot has two front wheels navigated by two servo motors 5V and a free castor wheel is placed at front. The robot will be moving towards the specified goal position determined by the user by using the dead reckoning method. The data acquisition is done by the PIC microcontroller from various sensors including ultrasonic range detector sensor, infrared sensor and encoder. The data will be sent to the computer using serial data transmission method. The data will be processed by the computer using Matlab and Fuzzy Logic to get the correct angle and the acceleration for the robot. This data will be sent back to the PIC microcontroller to control the motors for navigation purposes. The servo motors are easily controlled by feed-in pulses directly from the PIC microcontroller. The PIC microcontroller 16F877 can perform a large number of applications especially for control and sensing applications. In addition, MATLAB is a standard and cost- effective tool within the engineering community for scientific applications. The Fuzzy Logic is used here to predict the direction of movement and the speed for the robot. Since the input data for predicting the direction and speed is too large, the fuzzy logic is used for fast prediction.

Abstrak

Tujuan utama projek ini adalah untuk menghasilkan rekabentuk robot yang mampu menjelajah atau bergerak dari satu lokasi ke lokasi yang ditetapkan oleh pengguna. Ia juga mampu bergerak menuju ke kedudukan tersebut mengikut laluan yang paling dekat tanpa melanggar objek objek yang berada disepanjang laluan tersebut. Robot ini menggunakan aplikasi logik fuzzi sebagai pengawal utama bagi membolehkan ia mencapai matlamat utama rekabentuk ini. Robot ini dibina menggunakan perkakasan yang berkos rendah. Oleh kerana robot ini menggunakan aplikasi logik fuzzi. Maka ia dinamakan F_Bot. F_Bot menggunakan 2 buah roda yang digerakkan menggunakan 2 buah motor servo dan sebuah roda kastor yang dipasang dibahagian hadapan robot. F_Bot bergerak menuju ke kedudukan yang ditetapkan oleh pengguna dengan menggunakan konsep “dead reckoning”. Pengawal mikro PIC mempunyai dua fungsi di dalam projek ini. Ia berfungsi bagi mengumpul data-data dari semua sensor dan menghantar data-data tersebut secara serial ke Matlab (logik fuzzi) selaku pengawal utama. Antara penderia yang digunakan adalah penderia infra merah dan penderia ultrasonic. Pengawal mikro PIC juga berfungsi untuk menerima data kawalan dari Matlab dan mengawal operasi servo motor berdasarkan data kawalan yang diterima. Motor servo boleh dikawal dengan mudah dengan menghantar isyarat denyut yang dihasilkan oleh pengawal mikro PIC secara terus. Disepanjang projek ini berjalan, wujud pelbagai masalah teknikal dan pembangunan perisian dalam usaha menyiapkan projek ini. Masalah utama yang dihadapi adalah mencari cara yang terbaik untuk mengintegrasikan pelbagai perkakasan supaya F_Bot dapat beroperasi dengan baik. Pengawal mikro PIC selalu digunakan dalam pelbagai aplikasi. Antaranya adalah aplikasi penderiaan dan pengawalan. Sebagai tambahan, Matlab merupakan perkakasan yang piawai dan berkos berkesan dikalangan komuniti kejuruteraan dalam aplikasi saintifik. Logic fuzzi digunakan bagi menjangkakan arah pergerakan dan kelajuan robot. Memandangkan data masukan dari penderia-penderia bagi menjangkakan arah pergerakan dan kelajuan robot adalah terlalu besar, maka logik fuzzi digunakan bagi melakukan jangkaan secara lebih pantas.

Acknowledgement

At the outset, I am thankful to God Almighty for his continuous grace and sustenance which carries me through the endeavour.

I heartily acknowledge Dr. Andal Jaya Laksmi from the Department of Electric & Electronic Engineering, for expressing her confidence in me by giving such a challenging project.

Words would be insufficient to express my heart-felt gratitude to the department faculties for rendering their unflinching support and guidance throughout the duration of this project work. I also would like to thank to those in the faculty of the E&E who have directly or indirectly involved with this project.

Last but not least; I would like to thank my fellow course-mates and some junior and senior students, especially Miss Siti Hajar bte Mohd and Mr. Najib, for their kind and highly valuable advices.

TABLE OF CONTENTS

Contents Page No.

LIST OF FIGURES

LIST OF TABLES

APPENDIX

Chapter 1: Introduction

1.1 Background

This report presents the progress made in the development of the navigation algorithm using fuzzy logic for the semi-autonomous mobile robot.

1.2 Problem statement

Mobile robot are increasingly being employed in various applications such as forestry, mining, search and rescue, hazardous site inspection, and patrolling. These applications often require robots to travel across unprepared terrain to inspect a location. For example, a robot has to perform a patrolling job in the terrain that has many obstacles. To perform this task, the robot has to rely on the on-board sensors for navigation and control. These sensors generally contain significant uncertainty and error in their measurements. The outdoors operation often require the robot to operate autonomously, which requires real-time decision making that is constrained by limited on-board computational sources. The combined effects of sensor measurement uncertainty and error, and limited computation make the navigation planning and control of mobile robot a problem.

1.3 Project and its aims.

This project aims at building a semi-autonomous mobile robot. The term “semi-autonomous” here means the mobile robot performs a given task from the user based on its behaviour or intelligence.

The project contains challenges from a mechanical, electronic and software part. The mechanical challenge is to design and construct a suitable shape mobile platform with a specified weight and payload it could carry, while the electronic challenge is to design a control system for the robot between the mechanical drive and the signal from the computer and sensor, and proper feedback system for a good navigation purpose.
As a robot, all the mechanical and electronic parts here must be controlled and manipulated by a microcontroller. This is where the software challenge applied. In this project, the

application is to let the robot navigates from its current position to a given position by itself without colliding any obstacle along the path.

1.4 Overall system architecture

1.4.1 Implementation language

The plan was to develop part of the system in Picbasic Pro and the remainder in Matlab. Ultimately this will make developing the system much simpler. Matlab not only provides fuzzy logic toolbox facilities, but easily display the results at each step in the path planning process. This ability to view the results easily makes development much simpler.

1.4.2 System structure

Figures 1.1 and 1.2 show the overall system architecture being employed in this project.

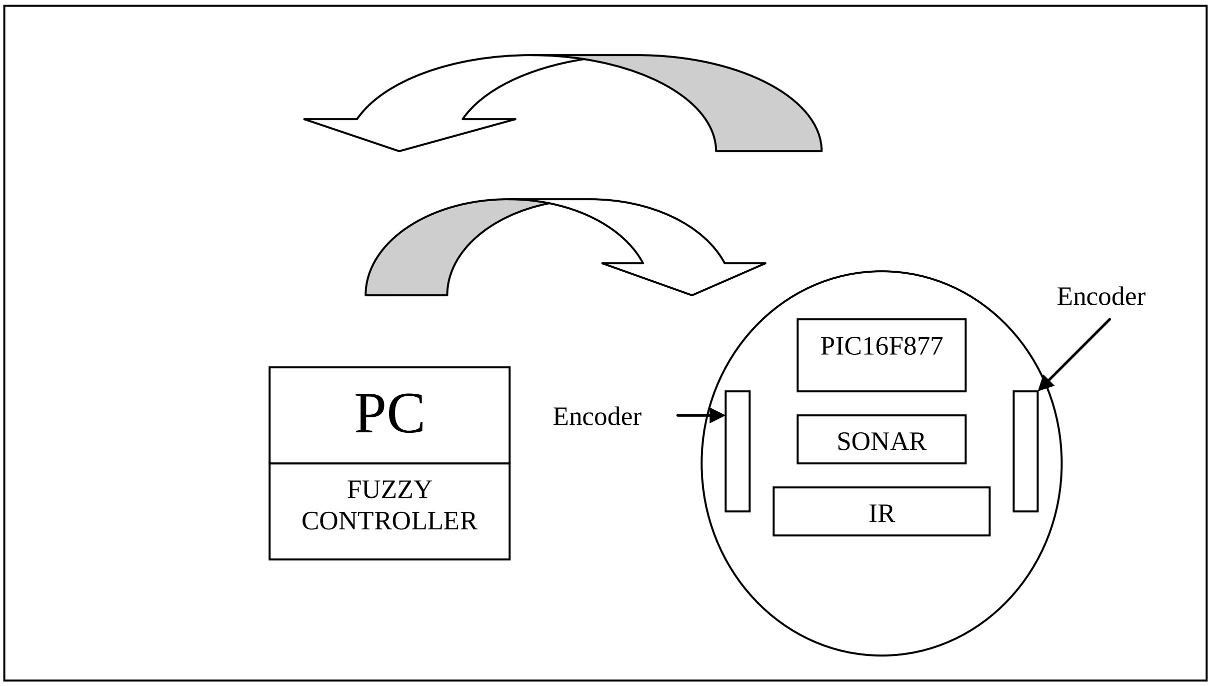


Figure 1.1: F_Bot System Architecture

Figure 1.2: The F_Bot Block Diagram

1.5 Report Overview

This report is divided into five main parts - Introduction, Literature Review, Design and Construction, Product and Conclusion. The first being this introductory part. The second chapter will briefly deal about robotics and fuzzy logic application, odometry theory and serial communication. The third chapter will include the three aspects of engineering involved in the design and construction of the F_Bot - Mechanical, Electrical, and Software respectively. The fourth chapter is about the final product and the progress. Finally, the fifth chapter in conclusion. References and appendices are attached after chapter fifth.

Chapter 2: Literature Review

2.1 Robotics

At the down of the new millennium, robotics is undergoing a major transformation in scope and dimension. From a largely dominant industrial focus, robotics is rapidly expanding into the challenges of unstructured environments. Interacting with, assisting, serving, and exploring with humans, the emerging robots will increasingly touch people and their lives.

Since robots are used mainly in manufacturing, we see their impact is seen in the products we use every day. Usually, this results in a cheaper product. Robots are also used in cases where it can perform a better job than a human does such as surgery where high precision is a benefit. And, robots are used in exploration in dangerous places such as in volcanoes which allows us to learn remotely without endangering ourselves.

The word "robot” was coined by Karel Capek who wrote a play entitled "R.U.R." or "Rossum's Universal Robots" back in 1921. The base for this word comes from the Czech word 'robotnik' which means 'worker'. In his play, the machines were modelled after humans had great power but without common human failings. In the end, these machines were used for war and eventually turned against their human creators.

A popular science fiction writer Isaac Asimov created the Three Laws of Robotics:

#1 A robot must not injure a human being or, through inaction, allow a human
being to come to harm.
#2 A robot must always obey orders given to it by a human being, except where
it would conflict with the first law.
#3 A robot must protect it's own existence, except where it would conflict with
the first or second law.

The population of robots is growing rapidly. This growth is lead by Japan that has almost twice as many robots as the USA. All estimates suggest that robots will play an everincreasing role in modern society. They will continue to be used in tasks where danger, repetition, cost, and precision prevents humans from performing.

2.2 Fuzzy logic

The concept of Fuzzy Logic was conceived by Lofti Zadeh [5], a professor at the university of California at Berkley, and presented not as a control methodology, but as a way of processing data by allowing partial set membership rather than crisp set membership of non-membership. Professor Zadeh has reasoned that people do not require precise, numerical information input, and yet they are capable of highly adaptive control.

2.2.1 What is Fuzzy Logic?

Fuzzy logic is not logic that is fuzzy, but logic that is used to describe fuzziness. Fuzzy logic is the theory of fuzzy sets, sets that calibrate vagueness. Fuzzy logic is based on the idea that all things admit of degrees. Temperature, height, speed, distance, beauty – all comes on a sliding scale.

- The motor is running really hot;
- Tom is a very tall man; and
- Electric cars are not very fast.

FL is a problem solving control system methodology that lends itself to implementation in system ranging from simple, small, embedded microcontrollers to large, networked, multi-channel PC or workstation-based data acquisition and control systems. It can be implemented in hardware, software or a combination of both. FL provides a simple way to arrive at a definite conclusion based upon vague, ambiguous, imprecise, noisy, or missing input information. FL is an approach to control problems' mimics how a person would make decisions, only much faster with a quicker technique.

2.2.2 Why use Fuzzy Logic?

FL offers several unique features that make it a particularly good choice for many control problems. These unique features are:

1) It is inherently robust since:

- It does not require precise, noise-free input;
- It can be programmed to fail safely if a feedback sensor quits or is destroyed;
- It has smooth control function despite a wide range of input variations;

2) Since the FL controller processes user-defined rules governing the target control system, it can be modified and tweaked easily to improve or drastically alter system performance. New sensors can easily be incorporated into the system simply by generating appropriate governing rules;

3) FL is not limited to a few feedback inputs and one or two control outputs, nor is it necessary to measure or compute rate-of-change parameters in order for it to be implemented. Any sensor data that provides some indication of a system's actions and reactions is sufficient. This allows the sensors to be inexpensive and imprecise thus keeping the overall system cost and complexity low;

4) FL can control non-linear systems that would be difficult or impossible to model mathematically. This opens doors for control systems that would normally be deemed unfeasible for automation.

2.2.3 How to use Fuzzy Logic?

1) Define the control objectives and criteria: What am I trying to control? What do I have to do to control the system? What kind of response do I need? What are the possible system failure modes?;

2) Determine the input and output relationships and choose a minimum number of variables for input to the FL engine;

3) By using the rule-based structure of FL, break the control problem down into a series of IF X AND Y THEN Z rules that define the desired system output response for the given system input conditions. The number and complexity of rules depend on the number of input parameters to be processed and the number fuzzy variables are associated with each parameter. If possible, use at least one variable and its time derivative. Although it is possible

to use a single, instantaneous error parameter without knowing its rate of change, this cripples the system's ability to minimize overshoot for a step inputs;

4) Create FL membership functions that define the meaning (values) of Input/Output terms used in the rules;

5) Create the necessary pre- and post-processing FL routines if implemented in S/W, otherwise program the rules into the FL H/W engine; and

6) Test the system, evaluate the results, tune the rules and membership functions, and retest until satisfactory results are obtained.

2.2.4 Fuzzy sets

Fuzzy sets are an extension of classical sets. In a classical set, an element may be a member of the set or not a member of the set. In Fuzzy set theory, an element may be a partial member of a set.
Fuzzy set theory introduces formalism for allowing set membership values to vary between 0 and 1, and not only restricted to these two values.

2.2.5 Member functions

These functions determine the membership value for input elements. In this way, the membership value does not required to be listed for every possible set member but defined as a function instead.
For instance, sonar(x) = {0 if x>=1, 1 if x<= 0.7, else (1-x)/0.3}
Set membership functions can be shown as graphs as illustrated in Figure 2.0.

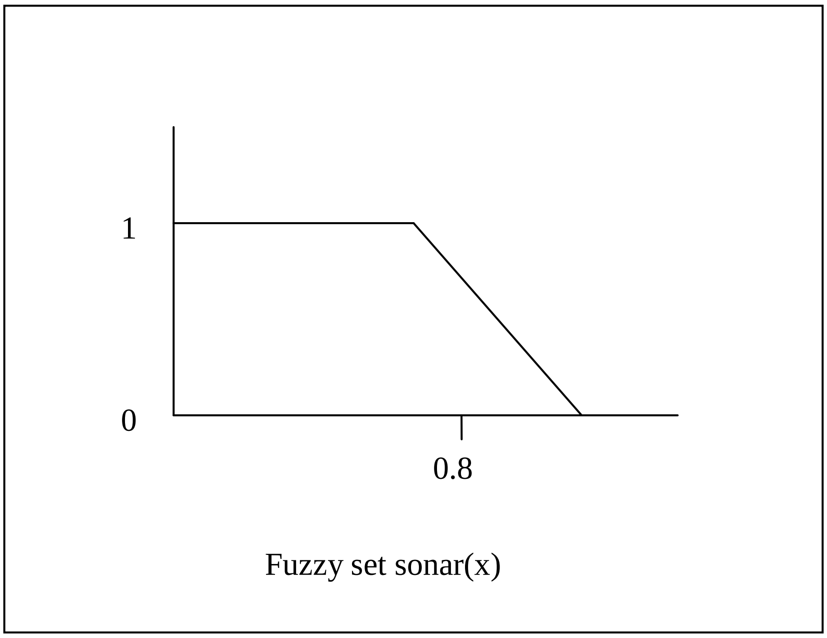


Figure 2.0: Set membership functions

2.2.6 Fuzzy Controller

A fuzzy controller takes inputs, converts the inputs to fuzzy values and then uses a fuzzy inference procedure to calculate what appropriate outputs should be.

The design consists of three steps; which are:

- Fuzzification;
- Fuzzy rules; and
- Defuzzification.

Figure 2.1 illustrates the block diagram of the fuzzy controller system.

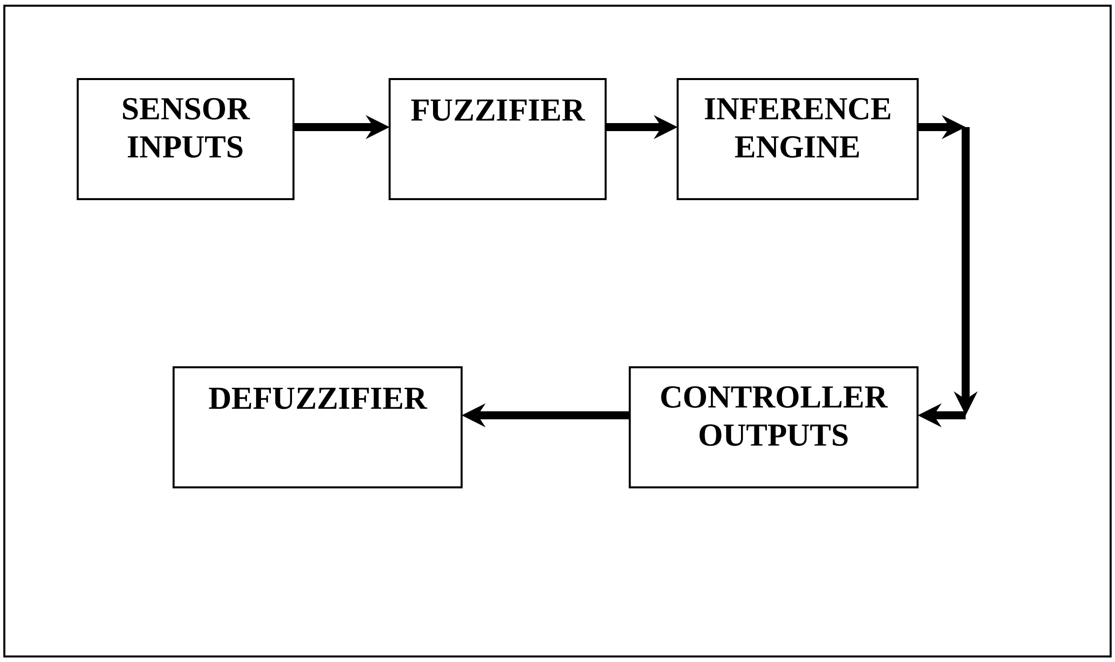


Figure 2.1: Fuzzy controller system

2.2.6.1 Fuzzification

Fuzzification is a process of retrieving the fuzzy set member value for a particular input variable.

Figure 2.2 illustrates the fuzzification of the input value.

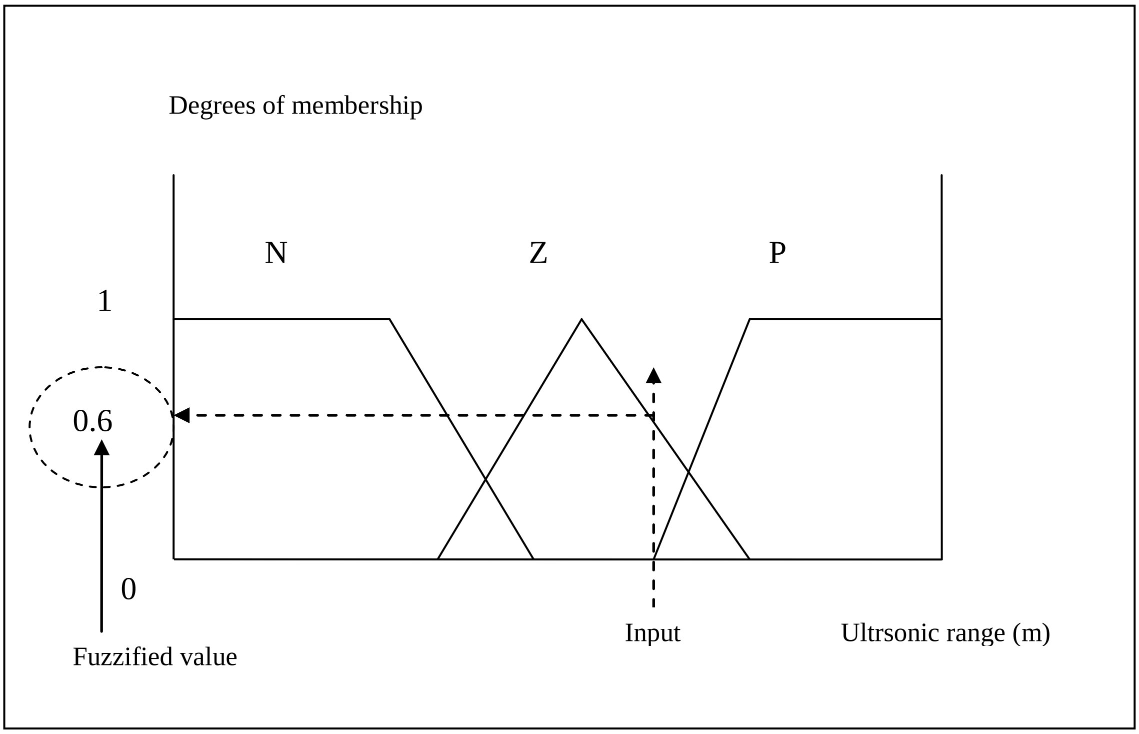


Figure 2.2: Fuzzification Process

2.2.6.2 Inference Engine

Fuzzified input values are used as values that are acted upon by the fuzzy inference engine. The fuzzy inference engine has rules that manipulate input values to produce fuzzy rules for the output.
There are several methods that can be used.

The **MAX-MIN** method tests the magnitudes of each rule and selects the highest one. The horizontal coordinate of the "Fuzzy centroid"of the area under that function is taken as the output. This method does not combine the effects of all applicable rules but does produce a continuous output function and is easy to implement.

The **MAX-DOT or MAX-PRODUCT** method scales each member function to fit under its respective peak value and takes the horizontal coordinate of the "fuzzy" centroid of the composite area under the function (s) as the output. Essentially, the member function(s) are shrunk so that their peak equals the magnitude of their respective function ("negative"," zero", "positive"). This method combines the influence of all active rules and produces a smooth, continuous output.

The **AVERAGING** method is another approach that works but fails to give increased weighting to more rule votes per output member function. For example, if three "negative"rules fire, but only one"zero rules does, averaging will not reflect this difference since both averages will equal 0.5. Each function is clipped at the average and the "fuzzy" centroid of the composite area is computed.

The **ROOT-SUM-SQUARE(RSS**) method combines the effects of all applicable rules, scales the functions at their respective magnitudes, and computes the" fuzzy" centroid of the composite area. This method is more complicated mathematically than other methods, but was selected for this example since it seemed to give the best weighted influenced to all firing rules.

Below is the example of how the inference engine works:-

Fuzzified values for input 1 and 2 are as below:

- Range :- N(3.3) =0.0, Z(3.3)=0.5, P(3.3)=0.3
- Position :- N(2.6)=0.6, Z(2.6)=0.4, P(2.6)=0.0

Rule base (example):-

- If range =Z AND position=N Then angle = MP
- If range=Z AND position=Z Then angle = NC
- If range = P AND position=N Then angle = MN
- If range = P AND position=Z Then angle = P

Therefore, the fuzzy output values are

- MP = min(0.5,0.6)=0.5
- NC = min(0.5,0.4)=0.4
- MN = min(0.3,0.6)=0.3
- SP = min(0.3,0.4)=0.1

To find the output value, it must be defuzzified first.

2.2.6.3 Defuzzification

Normal rule bases have several rules, so when the input has been applied to the rules, several output fuzzy sets can exist. In this case the outputs are combined and defuzzified by a centre of gravity approach.

The defuzzified method that being used is called the centre of gravity approach. For each fuzzy set, we define a mid value called its centre of gravity. The defuzzified values are then calculated as centre of gravity of the output fuzzy sets.

Based on the example before, the defuzzified values are as below,

Centre values of the output sets

- MN = 2.5/2 = 1.25 degree
- NC = 3.0 degree
- MP = 6 – (2.5/2) =4.75 degree
- SP = 6 – 2.5 = 3.5 degree

Thus,

Using the centre of gravity method

Angle = (1.25*0.3 + 3.0*0.4 + 4.75*0.5 +3.5*0.1) / (0.5 + 0.4 + 0.3 + 0.1)

= 3.31 degrees.

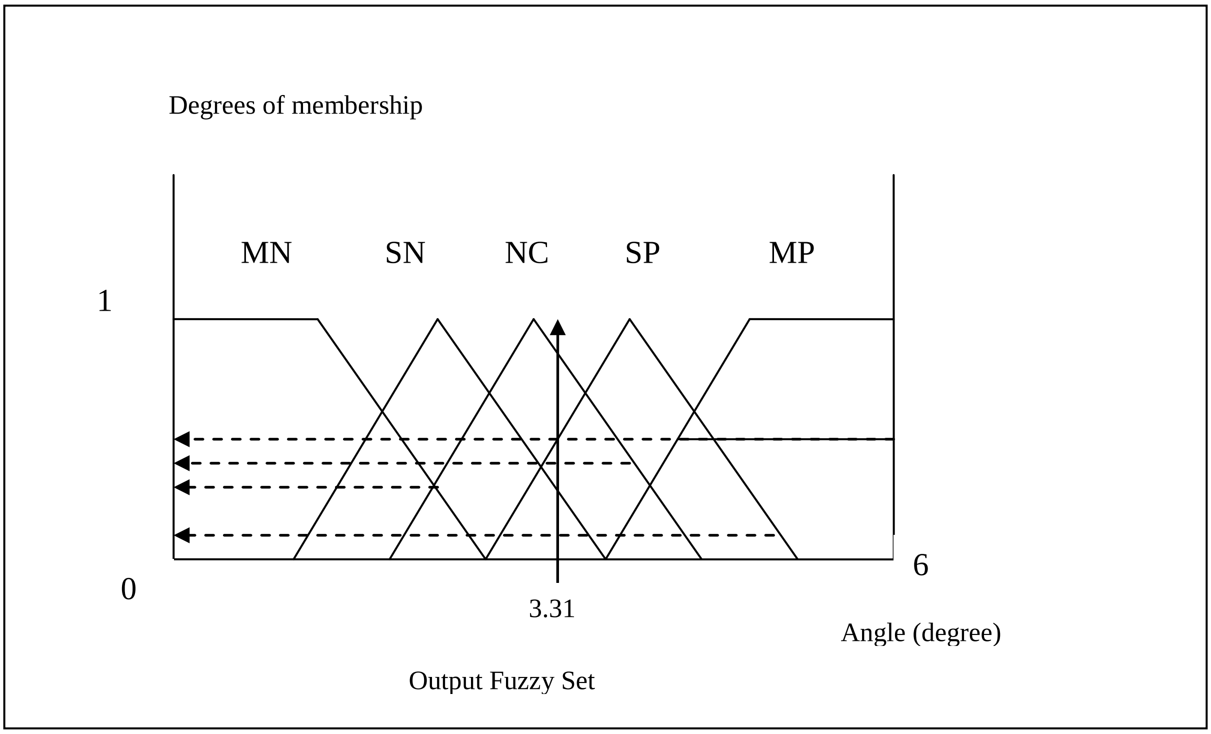


Figure 2.3: Defuzzification Process

2.3 Odometry Technique

A widely used technique for deductive calculus is the ‘odometry’. Odometry is a basic navigation method, using knowledge of wheel’s motion to estimate vehicle motion. A typical odometry system consists of a data acquisition system added to the mobile robot's wheels, and a group of equations for the calculus of its position. Figure 2.4 shows the interior view of a mobile robot platform.

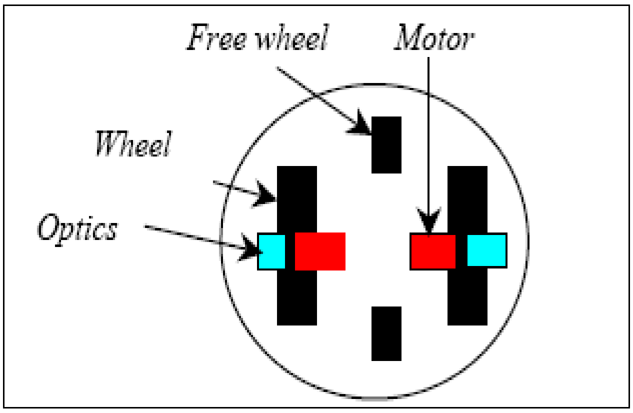


Figure 2.4: Interior view of a mobile robot platform

The data acquisition will be done using the optical encoder for each wheel. The encoder will detect the number of pulses generated by the rotation of the motor. Figure 2.5shows the encoder implementation with the motor.

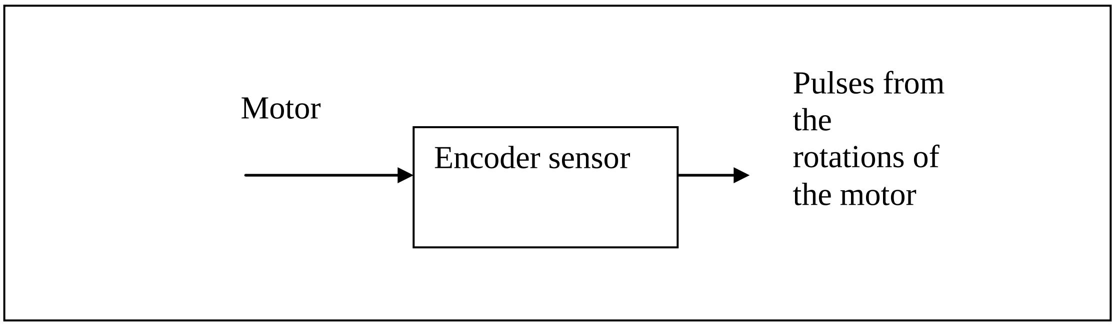


Figure 2.5: Encoder Implementation

The pulses generated by the motor will determine the movement of each motor and eventually determine the position of the robot.

Optical encoders create pulses of the moved angle by each one of the motorized wheels, acquiring indirectly in this way the robot displacement. The distance moved by the robot in calculated by means of:

$$Cm = \pi\ Dn / n\ Ce \qquad (2.1)$$

where Cm is the conversion factor that translates the pulses from the encoder to linear displacement; Dn is the nominal diameter of the wheel in millimetres, Ce is the resolution of the codifier and n is the reduction factor between the motor and gear of the wheel axle. In this case, n equals to one. From Figure 2.6, the travelled distance is calculated by taking into account the relative movement of the left and right wheels, ΔULi and ΔURi respectively, then:

$$\Delta ULi = Cm\ N_L \qquad (2.2)$$

$$\Delta URi = Cm\ N_R \qquad (2.3)$$

where N represents the increment pulse with a I sample rate. The incremental linear displacement ΔUCi of the robot's center is given by:

$$\Delta UCi = (\Delta URi - \Delta ULi) / 2 \qquad (2.4)$$

and the orientation is computed with:

$$\Delta \Theta_i = (\Delta URi - \Delta ULi) / b \qquad (2.5)$$

where b is the diameter of the mobile base and ideally is the distance between the two points of contact of the wheels with the floor. The new relative orientation of the robot θ i can be calculated from:

$$\Theta_i = \Theta_{i-1} + \Delta\Theta_i \qquad (2.6)$$

and the new relative position of the robot's center is given by:

$$X = X_{i-1} + \Delta C_i \mathrm{Cos}\ \Theta_i \qquad (2.7)$$

$$Y = Y_{i-1} + \Delta C_i \sin\ \Theta_i \qquad (2.8)$$

where xi and yi are the relative coordinates, accumulate the center C of the robot at instant i .

Figure 2.6 illustrates the odometry technique applied to the movement of the robot to determine the current position of the robot relative to its old position.

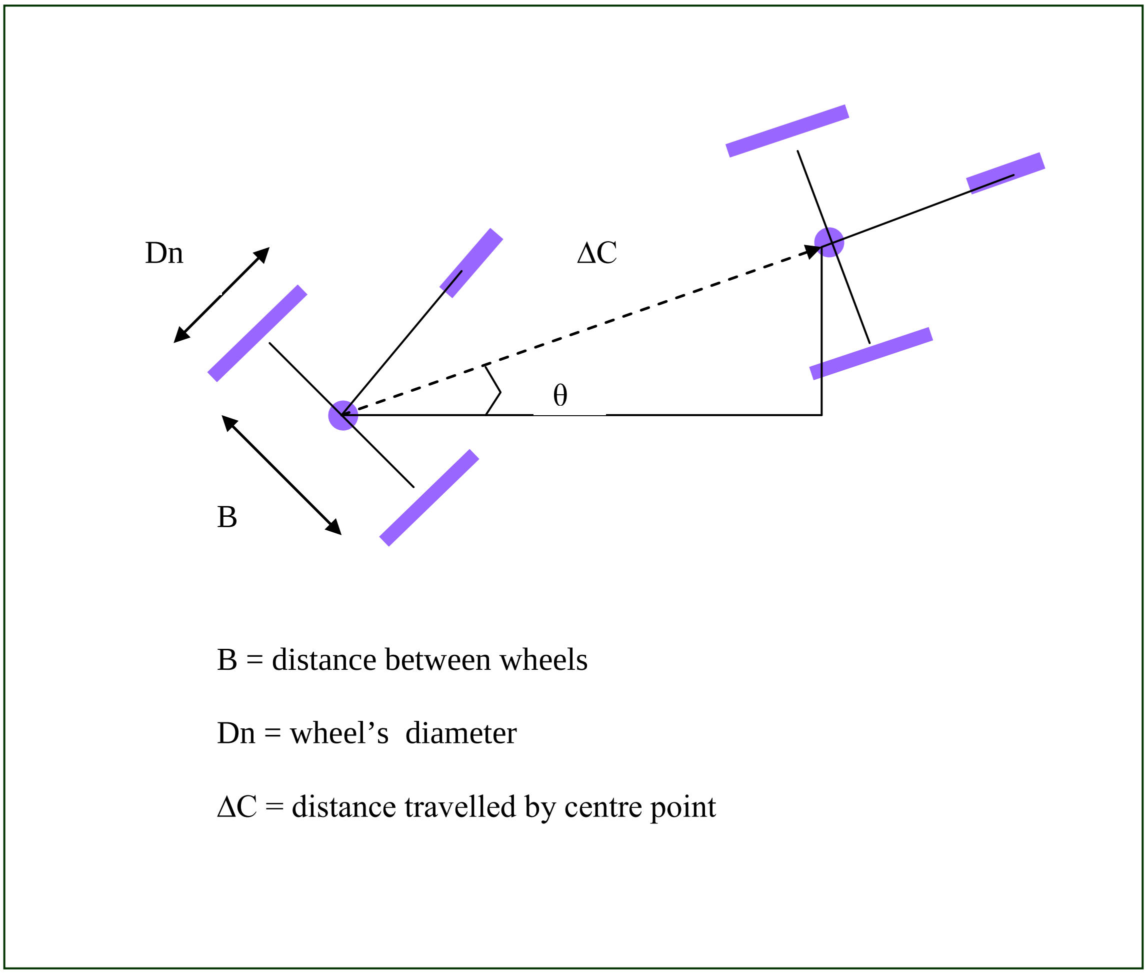
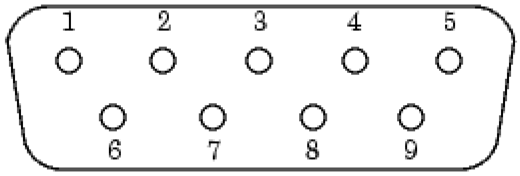


Figure 2.6: Mobile Robot Odometry System

2.4 Serial Port Signals and Pin Assignments

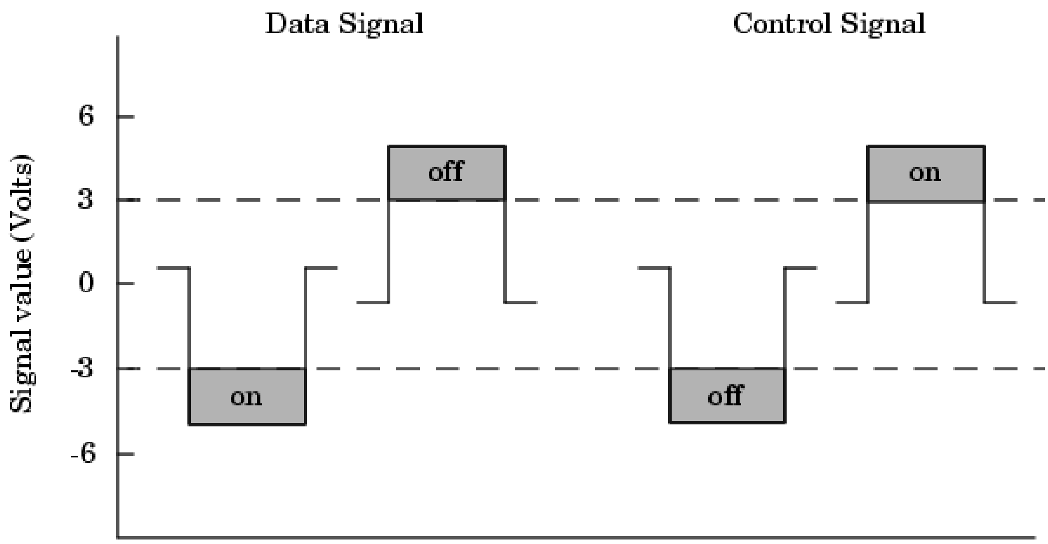
Serial ports consist of two signal types: data signals and control signals. To support these signal types, as well as the signal ground, the RS-232 standard defines a 25-pin connection. However, most PC's and UNIX platforms use a 9-pin connection reference. In fact, only three pins are required for serial port communications: one for receiving data, one for transmitting data, and one for the signal ground. The pin assignment scheme for a 9-pin male connector on a DTE is given below.



Pin	Label	Signal Name	Signal Type
1	CD	Carrier Detect	Control
2	RD	Received Data	Data
3	TD	Transmitted Data	Data
4	DTR	Data Terminal Ready	Control
5	GND	Signal Ground	Ground
6	DSR	Data Set Ready	Control
7	RTS	Request to Send	Control
8	CTS	Clear to Send	Control
9	RI	Ring Indicator	Control

2.4.1 Signal States

Signals can be in either an active state or an inactive state. An active state corresponds to the binary value 1, while an inactive state corresponds to the binary value 0. An active signal state is often described as logic 1, on, true, or a mark. An inactive signal state is often described as logic 0, off, false, or a space. For data signals, the "on" state occurs when the received signal voltage is more negative than -3 volts, while the "off" state occurs for voltages more positive than 3 volts. For control signals, the "on" state occurs when the received signal voltage is more positive than 3 volts, while the "off" state occurs for voltages more negative than -3 volts. The voltage between -3 volts and +3 volts is considered a transition region, and the signal state is undefined. To bring the signal to the "on" state, the controlling device unasserts (or lowers) the value for data pins and asserts (or raises) the value for control pins. Conversely, to bring the signal to the "off" state, the controlling device asserts the value for data pins and unasserts the value for control pins. The "on" and "off" states for a data signal and for a control signal are shown below.



2.4.2 The Data Pins

Most serial port devices support full-duplex communication meaning that they can send and receive data at the same time. Therefore, separate pins are used for transmitting and receiving data. For these devices, the TD, RD, and GND pins are used. However, some types of serial port devices support only one-way or half-duplex communications. For these devices, only the TD and GND pins are used. In this guide, it is assumed that a full-duplex serial port is connected to your device. The TD pin carries data transmitted by a DTE to a DCE. The RD pin carries data that is received by a DTE from a DCE.

2.4.3 The Control Pins

9-pin serial ports provide several control pins that: Signal the presence of connected devices Control the flow of data The control pins include RTS and CTS, DTR and DSR, CD, and RI. The RTS and CTS Pins. The RTS and CTS pins are used to signal whether the devices are ready to send or receive data. This type of data flow control - called hardware handshaking - is used to prevent data loss during transmission. When enabled for both the DTE and DCE, hardware handshaking using RTS and CTS follows these steps: The DTE asserts the RTS pin to instruct the DCE that it is ready to receive data. The DCE asserts the CTS pin indicating that it is clear to send data over the TD pin. If data can no longer be sent, the CTS pin is unasserted. The data is transmitted to the DTE over the TD pin. If data can no longer be accepted, the RTS pin is unasserted by the DTE and the data transmission is stopped. To enable hardware handshaking in MATLAB, refer to Controlling the Flow of Data: Handshaking.

The DTR and DSR Pins.

Many devices use the DSR and DTR pins to signal if they are connected and powered. Signaling the presence of connected devices using DTR and DSR follows these steps: The DTE asserts the DTR pin to request that the DCE connect to the communication line. The DCE asserts the DSR pin to indicate it's connected. DCE unasserts the DSR pin when it's disconnected from the communication line. The DTR and DSR pins were originally designed to provide an alternative method of hardware handshaking. However, the RTS and CTS pins are usually used in this way, and not the DSR and DTR pins. However, you should refer to your device documentation to determine its specific pin behavior.

The CD and RI Pins.

The CD and RI pins are typically used to indicate the presence of certain signals during modem-modem connections. CD is used by a modem to signal that it has made a connection with another modem, or has detected a carrier tone. CD is asserted when the DCE is receiving a signal of a suitable frequency. CD is unasserted if the DCE is not receiving a suitable signal. RI is used to indicate the presence of an audible ringing signal. RI is asserted when the DCE is receiving a ringing signal. RI is unasserted when the DCE is not receiving a ringing signal (for example, it's between rings).

Chapter 3: Design and Construction

3.1 Mechanical Design

The mechanical design would determine directly or indirectly the rest of the design processes - electrical, electronics, and software. The design should minimize weight to reduce power required, as high power drives are not easily available and portable. The basic design requirements for the mobile robot are forward and reverse drive. The drive could be either electrical or fuel. The fuel engine alternative is not feasible under the presumed vehicle dimension and weight. An electrical drive system is more practical for the mobile robot.

3.1.2 Brief Description of the Mechanical Components

3.1.2.1 Wheels

The importance of wheels and tyres in the automobile is obvious. Wheels are the important parts of the vehicle as they must support the weight of the vehicle and help to protect it from the road shocks. In addition, the rear wheels must transmit the power and steer the vehicle to the left and right direction. Wheels must also be perfectly balanced.

The various requirements of an automobile wheel are:
i) It should be balanced both statically and dynamically.
ii) It should be as light as possible so that the up sprung weight is least.
iii) Its material should not deteriorate with weathering and age.

3.1.2.2 Tyres

A tyre is a cushion provided with an automobile wheel. It consists of mainly the outer cover, i.e., the tyre proper and the tube inside. The tyre tube assembly is mounted over the wheel rim. It is the air inside the tube that carries the entire load and provides the cushion.

The tyre performs the following functions;
o To support the vehicle load;
o To provide cushion against shocks;

o To transmit driving and braking forces to the road;
o To provide cornering power for smooth steering.

3.1.2.2.1 Types of Tyres

The different types of tyres are:
a) Pneumatic tyre;
b) Tube tyre;
c) Tubeless tyre.

3.1.2.2.2 Desirable tyre properties:

Non-skidding

This is one of the most important tyre properties. The tread pattern on the tyre must be suitably designed to permit least amount of skidding even on wet road.

Uniform Wear

To maintain the non skidding property, it is very essential that the wear on the tyre tread must be uniform. The ribbed tread patterns help to achieve this.

Load Carrying

The tyre is subjected to alternating stresses during each revolution of the wheel. The tyre material and design must be able to ensure that the tyre is able to sustain these stresses.

.**Cushioning**

The tyre should be able to absorb small, high frequency vibrations set up by the road surface and thus provide cushioning effect.

Power Consumption

The automotive tyre absorbs some power which is due to friction between the tread rubber and the road surface and also due to hysterisis loss on account of the tyre being continuously flexed and released. This power comes from the engine fuel and should be the least possible. It is seen that the synthetic tyres comsume more power while rolling than the ones made out of natural rubber.

Tyre Noise

The tyre noise may be in the form of definite pattern sing. In all these cases, it is desirable that the noise would be minimum.

Balancing

This is a very important consideration. The tyre being a rotating part of the automobile, hence it must be balanced statically as well as dynamically. The absence of balancing gives rise to peculiar oscillations

3.1.2.3 Wheel Placement and Turning Circle

The position where the wheels are located on the robot base affects the turning circle, or radius, of the robot. Thus, the wheels have to be placed at the base of the robot rather than outside of the base. This decreases the effective size of the robot and allows the robot to turn in a tighter circle.

Note that when the wheels are outside, the effective diameter of the wheel increased. Assuming the wheels are mounted on the center line in the base, the turning circle of the robot is defined as the distance between the wheels. For example, if the wheels are placed eight inches apart, even if the robot is smaller, the turning circle is eight inches.

3.1.2.4 Castor Wheel

To be effective, the caster on the robot must not impede the direction or speed of the machine's travel. Cheap swivel casters can catch and not swivel properly when the robot changes its direction. This can definitely cause the robot to veer off course because the robot wants to go one way, but the caster is still pointing in another direction.

Casters with ball bearings—as opposed to a simple plastic insert—tend to provide better results. Even with ball bearings, the swivel action may be stiff or sporadic, so always test first. Casters can be tested by rolling them on the floor and making them turn in different directions. If the caster does not quickly and effortlessly reorient itself in the direction of new travel, different castor has to be selected.

3.1.2.4.1 Caster Wheel Material

There are literally thousands of variations in casters. For robotics, only a few critical details are concerned: cost, size, availability, reliability, traction, and load capacity. Cost, size, and availability are obvious.

- **Roll ability** is the ease with which the caster wheel turns when underload. Casters meant for a heavy-duty application may not roll well when used with a light robot. Roll ability is determined by the design of the caster, as veil as the caster wheel material, size and width, tread, and other factors. There is no magical formula for determining whether a given caster will operate well with a given robot.

- **Traction** is the ability of the caster wheel to grip the surface it is rolling over. This is greatly determined by both the weight on the caster and the wheel material. The harder the material (e.g., as steel or hard rubber), the less traction of the caster wheel. For high-traction applications, select a soft rubber (which tends to be gray in industrial casters), Buna rubber, or a similar material.
- Load capacity is the weight the castor can carry. Heavier loads than that listed will reduce the functionality and the life of the castor. For a robot with two drive wheels and one castor, the castor is supporting one-third of the robot's weight.

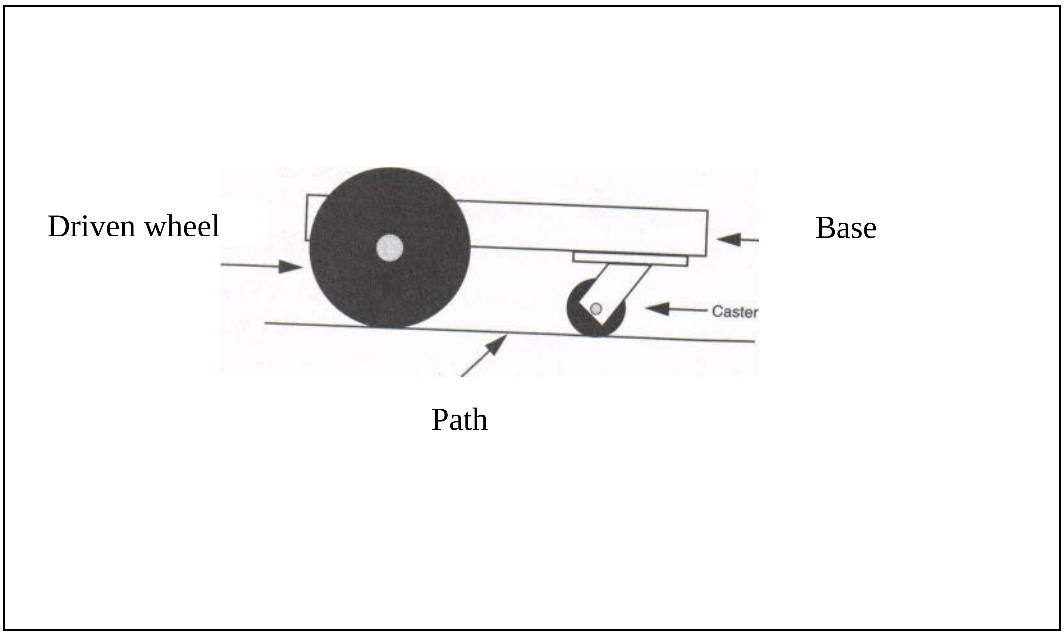


Figure 3.1: Diagram of mobile robot platform with a castor wheel

3.1.2.5 Frame

Frame is made up of galvanized iron pipe and sheet. The functions of frame are as follows:

A. To support the components, motor etc; and
B. To withstand static and dynamic load.

.

3.1.3 Design Criteria

The final goal of the mechanical design is to fabricate a vehicle that is safe, economic and practical to manufacture. Different design approaches must be adjusted to be compatible with the market. In approaching the final design some criteria should be established to guide decision making processes. The following are the design criteria that should be established in designing the project.

The design of the vehicle and its various components for efficient operation are:

- Selection of proper material;
- Safety of operation; and
- Economic status.

.

3.1.4 Design Options

3.1.4.1 Model 1

The vehicle can be made of four wheels. In this type of vehicle power is given to the shaft by means of motor and the differential gear is used in rear wheel for the turning of the wheels and another motor is connected with the steering which is used for turning front wheels.

3.1.4.2 Model 2

This model consists of three wheels. Two of them as rear wheel and one as front wheel. The two rear wheels would be driven by the servo motors. In this type of vehicle, motor is connected to each wheel directly. The ball castor is placed as the front wheel which can rotate in any direction to provide more degree of freedom to the vehicle.