KINEMATIC ANALYSIS OF WHEELED MOBILE ROBOTS

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Bachelor of Technology
In
Mechanical Engineering

By

ABHIJIT DATTA
&
SWARNENDU MANNA

Department of Mechanical Engineering
National Institute of Technology
Rourkela
2007

KINEMATIC ANALYSIS OF WHEELED MOBILE ROBOTS

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Bachelor of Technology
In
Mechanical Engineering

By

ABHIJIT DATTA
&
SWARNENDU MANNA

Under the Guidance of

Dr. D.R.K.Parhi

Department of Mechanical Engineering
National Institute of Technology
Rourkela
2007

National Institute of Technology
Rourkela

CERTIFICATE

This is to certify that the thesis entitled, “Kinematic analysis of wheeled mobile robots” submitted by Sri Abhijit Datta and Sri Swarnendu Manna in partial fulfillment of the requirements for the award of Bachelor of Technology Degree in Mechanical Engineering at the National Institute of Technology, Rourkela (Deemed University) is an authentic work carried out by them under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University / Institute for the award of any Degree or Diploma.

Date

Dr. D.R.K.Parhi
Dept. of Mechanical Engineering
National Institute of Technology
Rourkela - 769008

National Institute of Technology
Rourkela

ACKNOWLEDGEMENT

We would like to articulate our deep gratitude to our project guide Dr. D.R.K.Parhi who has always been our motivation for carrying out the project. .

It is our pleasure to refer Microsoft word 2003 of which the compilation of this report would have been impossible.

An assemblage of this nature could never have been attempted without reference to and inspiration from the works of others whose details are mentioned in reference section. we acknowledge our indebtedness to all of them.

Last but not the least to all of our friends who were patiently extended all sorts of help for accomplishing this undertaking.

Date

Abhijit Datta,
Swarnendu Manna
Dept. of Mechanical Engineering
National Institute of Technology
Rourkela - 769008

CONTENTS

ABSTRACT

The thesis deals with the problem associated with the path constraints and kinematic constraints of wheeled mobile robots. It also deals with the analysis of kinematic characteristics of the wheeled mobile robots and developing a kinematic model, including the path constraints and kinematic constraints equations.

From the analysis of established kinematic model some factors which affect the path tracking problem for a wheeled mobile robot, are taking into account.

Considering the analysis of Fuzzy Logic, the path followed by the autonomous wheeled mobile robot determine. And from the Fuzzy Logic graph the left wheel velocity and right wheel velocity are determined.

A robot arena has been devised by a MATLAB program to determine the path that the robot should follow avoiding all obstacles to reach its target.

LIST OF FIGURES

Chapter 1

INTRODUCTION

Introduction to wheeled mobile robots

Introduction to Kinematic Model of WMR

Introduction to Fuzzy Logic

INTRODUCTION TO WHEELED MOBILE ROBOTS:

The wheeled mobile robots(WMRs) are used in automating office, hospitals and factory floors operations, particularly in fully automatic factors.

We are considering a two degree of freedom mobile robot, capable of navigating without external guide track. This robot consists of two differently driven wheels, which have good maneuverability and simple mechanical design. This type of two wheeled robot has great advantage over omni directional robot because it requires fewer parts, it is easy to control, saves computation time and has less cost.

The most important aspect of WMR is the accuracy and reliability of its guidance system, which ensures that the robot goes from one position to another position following the specified path, taking in to account the posture of WMR and the tracking error, and for this, a suitable kinematic model is to be designed.

Robot manipulators have complex nonlinear dynamics that make their accurate and robust control
difficult. On the other hand, they fall in the class of Lagrangian dynamical systems, so that they have several extremely
nice physical properties that make their control straight forwarded

1. Different controllers had been developed for the
motion of robot manipulators, however, not until recently where there has been an interest in moving the robot itself, not
only its manipulators.

2.Shim and Sung proposed a WMR asymptotic control with drift less constraints based on empirical practice **using** the WMR **kinematic** equations. They showed that with the appropriate selection of the control parameters, the numerical performance of the asymptotic control could be effective.

3.The control algorithm consists of balance and velocity control, steering control, and straight line tracking control for navigation in a real indoor environments.

4. A computed torque control scheme for Cartesian velocity control of WMRs was developed.. Their control structure can be used to control any **mobile** robot if its inverse dynamic model exists. A

discontinuous stabilizing controller for WMRs with nonholonomic constraints where the state of the robot asymptotically converges to the target configuration with a smooth trajectory

5. A path tracking problem was formulated by Koh and Cho

6.For a **mobile** robot to follow a virtual target vehicle that is moved exactly along the path with specified velocity. The driving velocity control law was designed based on bang-bang control considering the acceleration bounds of driving wheels and the robot dynamic constraints in order to avoid wheel slippage or mechanical damage during navigation.

7. A dynamic modeling was done to design a tracking controller for a differentially steered **mobile** robot that is subject to wheel slip and external loads. A sliding mode control was used to develop a trajectory tracking control in the presence of bounded uncertainties

.8. A solution for the trajectory tracking problem for a WMR in the presence of disturbances that violate the
nonholonomic constraint is proposed later by the same authors based on discrete-time sliding mode control

9,10. An electromagnetic approach for path guidance of a **mobile**-robot-based automatic transport service system with a PD
control algorithm

11. Jiang, et al.

12. A model-based control design strategy was developed that
deals with global stabilization and global tracking control for the **kinematic** model with a nonholonomic WMR in the
presence of input saturations. An adaptive robust controller was proposed for the global tracking problem for the
dynamic of the non-holonomic systems with unknown dynamics

13. However, real time adaptive controls are not common
in practical applications due partly to the stability problems associated with them

.

14.A fuzzy logic controller had been tried for WMRs navigation. Montaner and Ramirez-Serrano
15. A fuzzy logic controller that can deal with the sensors inputs uncertainty and ambiguity for direction and velocity
maneuvers. A locomotion control structure was developed based on the integration of an adaptive fuzzy-net torque
controller with a **kinematic** controller to deal with unstructured unmodeled robot

dynamics for a non-holonomic **mobile robot cart.**

16.employed a sonar-based mapping of crop rows and **fuzzy** **logic** control-based steering for the

navigation of a WMR in an agricultural environment. They constructed a crop row map from the sonar readings and
transferred it to the **fuzzy** **logic** control system, which steers the robot along the crop row.

A local guidance control
method for WMR **using** **fuzzy** **logic** for guidance, obstacle avoidance and docking of a WMR was proposed by Vázquez
and Garcia
17.The method provide a smooth but not necessary optimal solution

Chapter 2

KINEMATIC MODEL OF WHEELED MOBILE ROBOTS

Path Constraints
Kinematic Constraints

KINEMATIC MODEL OF WMR:

The kinematic model of WMR must take into account the following two aspects:

i) One of the content is to build the relationship between the error posture and the commands which is used to control the driving system of the WMR. The method to build the relationship is to use the feedback from the error posture in Cartesian space. But this method has got a drawback, i.e., the system must perform many computations because of kinematics and co-ordinate transformations. This makes the system run at a lower sampling time compared to the wheel base control system.

So, using path error meter for correction of drive velocity and steering velocity with weighting constants and due to these, many weighting constants, the tracking error may not be corrected correctly, which is a drawback.

ii) The other content is that the structure, driving and steering system of the WMR must be taken into account in the kinematic model. The motions of the WMR are supposed to be only subject to constraints from the path which will be tracked and the constraints of the structure system of WMR are ignored.

As the structure constraints are ignored when the WMR is required to track a path, the possibility of finding path to be tracked is inexecutable, which is a major drawback.

Thus to overcome the drawbacks two kinds of constraints – i) path constraints, ii) kinematic constraints are taken into account.

THE PATH CONSTRAINTS:

In analyzing the path constraint of a wheeled mobile robot, the following assumptions are taken into account:-

1) There are no flexible parts in wheeled mobile robot.
2) There is no transitional slip between the wheel and the surface.
3) There is enough rotational friction between the wheels and the surface so, the wheels can rotate without disturbance.
4) The wheels should rotate without any disturbance.

The relationship between the two driving wheel velocities and the transition and angular velocities of the wheeled mobile robot is,

$$\omega = (V_l - V_{r)}) / B \ \ldots (1) \quad , \qquad V = (V_l + V_r) / 2 \ \ldots (2)$$

where, V_l and V_r are the velocities of left and right driving wheel, B is the wheel base.

The experimental wheeled mobile robot can be considered to have four wheels, two separate wheels on the behind for translation and rotation and two castors in the front for stability.

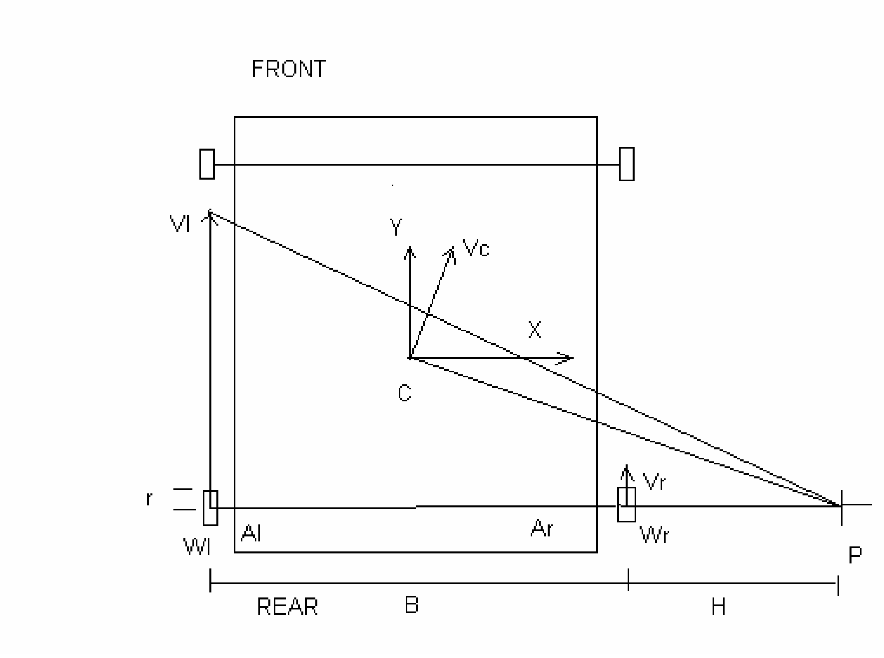


Fig:-2.1

C and M are centroid of the wheeled mobile robot (WMR) and mid point of wheel base

d = Distance between C and M

r = Radius of two driving wheels

A_l, A_r = Point of contact of the left and right wheel

W_l, W_r = Spinning speed of left and right wheel

P = Instant centre at this time

B = Wheel Base

$$\mathbf{V_c = \omega \times \overline{PC}} \ldots (3)$$

i, j and k are unit vectors of co – ordinate frame fixed at C along X, Y and Z axis.

$$\boldsymbol{\omega} = \frac{(\omega_l - \omega_r)\, r}{B} \times \mathbf{k} \ldots (4)$$

$$\mathbf{PC} = (B/2 + H).\mathbf{i} + d\mathbf{j} \ldots (5), \text{ where}$$

$$H = \left(\frac{V_r}{(V_l - V_r)}\right) . B = \left(\frac{\omega_r}{(\omega_l - \omega_r)}\right) . B$$

$$V_c = \frac{-(\omega_l - \omega_r)r - d}{B}\ \mathbf{i} + \frac{(\omega_l + \omega_r).r}{2}\ \mathbf{j} \ \ldots (6)$$

$$\left.\begin{aligned} Ex^R &= V_x = -(\omega_l - \omega_r).r.d / B \ \ldots(7) \\ Ey^R &= V_y = (\omega_l + \omega_r).r / 2 \quad \ldots(8) \\ Ez^R &= \omega = (\omega_r - \omega_r).r / B \quad \ldots(9) \end{aligned}\right\} \text{path constraints to find the angular velocity of the wheel.}$$

Path constraints used to eliminate the error posture.

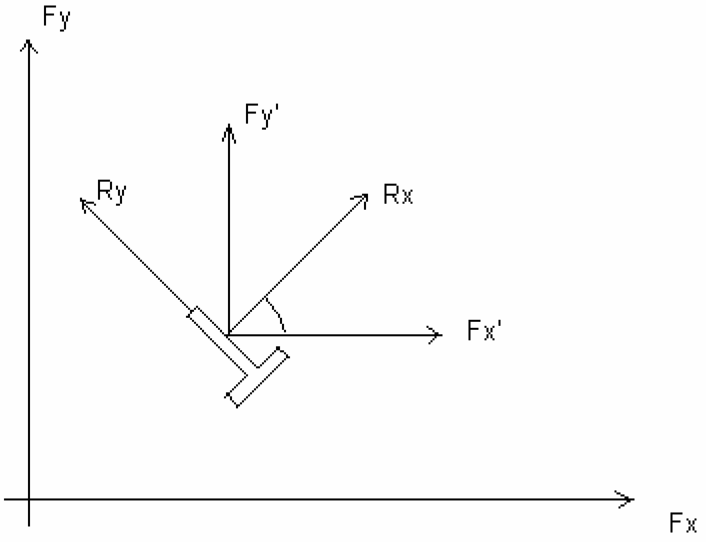


Fig:-2.2

Here F is absolute co – ordinate system.

The robot co – ordinate system R is assigned to the centroid of the robot body.

Using system $F^{/}$, small displacement change in each axis of system are shown.

The WMR's reference state $\mathbf{P_r = (X_r , Y_r , \Theta_r)}$ and the current state $\mathbf{P_c = (X_c , Y_c , \Theta_c)}$ can be derived then, the error posture $\mathbf{P_e = (E_x , E_y , E_d)}$ is calculated. $\mathbf{[P_e = P_r - P_c]}$. $\mathbf{P_e}$ can be treated as differential displacement.

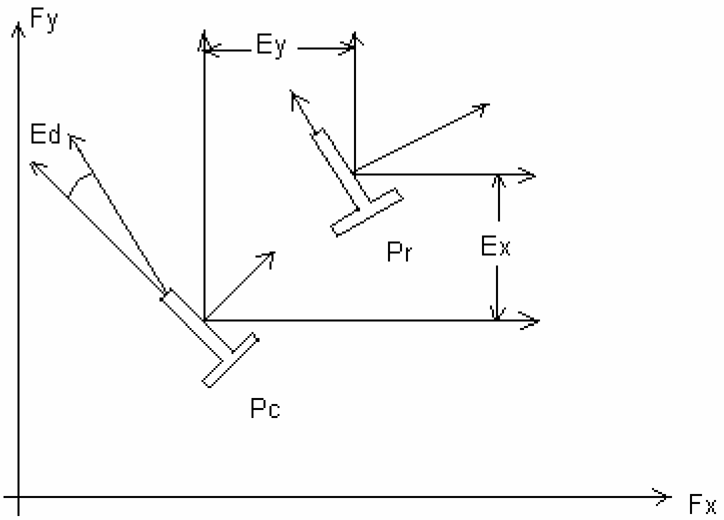


Fig:-2.3

$$\begin{pmatrix} \mathbf{Ex^R} \\ \mathbf{Ey^R} \\ \mathbf{Ed^R} \end{pmatrix} = \begin{pmatrix} \mathbf{\cos\Theta} & \mathbf{\sin\Theta} & \mathbf{0} \\ \mathbf{-\sin\Theta} & \mathbf{\cos\Theta} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{pmatrix} \begin{pmatrix} \mathbf{Ex^F} \\ \mathbf{Ex^F} \\ \mathbf{Ed^F} \end{pmatrix}$$

$\mathbf{Ex^R}$, $\mathbf{Ey^R}$ are the error componentsof the reference point **C** in X – axis and Y – axis in terms of robot co – ordinate system R respectively.

$\mathbf{Ed^R}$ => Header error of **‘C’**

Based on the equations (7) – (9) we obtain the angular velocity of right and left driving wheel which is used to eliminate the error posture.

THE KINEMATIC CONSTRAINTS:

The kinematic constraints for a wheeled mobile robot taken into account the limit imposed by the steering mechanism which defines the minimum radius of curvature of the reference points path Ex^R, Ey^R and Ex^R are the errors which are corrected by the driving and steering mechanism.

The spinning speed of left and right wheels ω_e, ω_r are

$Ex^R = V_x = -(\omega_e - \omega_r) \times r \times d/B,$ **B =** wheel base

$Ey^R = V_y = (\omega_e + \omega_r) \times r/2$

$\omega_l \in [\omega_{min}, \omega_{max}]$, $\omega_r \in [\omega_{min}, \omega_{max}]$

where ω_{min} and ω_{max} are minimum and maximum angular velocities of the two driving wheels.

The spinning speed of left and right wheels are obtained by the errors Ex^R and Ey^R.

While correcting the header error of 'ω' i.e., $\omega = (\omega_l - \omega_r) \times r/B$, along with ω_l and ω_r, the wheeled mobile robots allowable turning radius of curvature 'ρ_{min}' must be taken into account

$$\rho_{min} = \sqrt{d^2 + \frac{B^2(\omega_l + \omega_r)^2}{4(\omega_l - \omega_r)^2}}$$

The maximum steering angle $(\varphi_{max}) = \sin^{-1}(d/\rho_{min})$

If a known path is required to be followed by the reference point on the robot the minimum radius of curvature of the path must be larger than minimum radius of curvature of the reference point.

The steering angle of the reference point on the robot must be less than the maximum steering angle (φ_{max}) the robot to trace a known path, its kinematic model is composed of the path constraints and kinematic constraints.

Path Constraints:-	**or, Path Constraints:-**
$V_x = -(\omega_l - \omega_r) \mathrm{\ x\ } r \mathrm{\ x\ } d/B$	$V_x = -(\omega_l - \omega_r) \mathrm{\ x\ } r \mathrm{\ x\ } d/B$
$V_y = (\omega_l - \omega_r) \mathrm{\ x\ } r/2$	$V_y = (\omega_l + \omega_r) \mathrm{\ x\ } r/2$
$W = (\omega_l - \omega_r) \mathrm{\ x\ } r/B$	$\omega = (\omega_l - \omega_r) \mathrm{\ x\ } r/B$
Kinematic Constraints:-	**Kinematic Constraints:-**
$\omega_l \in [\omega_{min}, \omega_{max}]$	$\omega_l \in [\omega_{min}, \omega_{max}]$
$\omega_r \in [\omega_{min}, \omega_{max}]$	$\omega_r \in [\omega_{min}, \omega_{max}]$
$\rho_{min} = \sqrt{d^2 + \frac{B^2(\omega_l + \omega_r)^2}{4(\omega_l - \omega_r)^2}}$	$\varphi_{max} = \sin^{-1}(d/\rho_{min})$

i) Based on the path constraints, we can obtain the spinning speed of left and right driving wheels and eliminate the tracking error between the reference posture $(\mathbf{P_r})$ and current posture $(\mathbf{P_c})$.

ii) From the kinematic constraints, we obtain the relations of the spinning speed of left and right driving wheels with the maximum and minimum angular velocity of the driving wheels

$$\omega_l \in [\omega_{min} , \omega_{max}]$$
$$\omega_r \in [\omega_{min} , \omega_{max}]$$

And also the maximum steering angle,

$$(\varphi_{max}) = \sin^{-1}(d / \rho_{min})$$

Where, ρ_{min} = Allowable turning radius of curvature of the reference point

iii) If two WMRs have identical rigid body shape and driving system with reference point in different positions, their kinematic constraints and capability of tracking path are different. Also the limit imposed by the steering mechanism determines the minimum radius of curvature of the reference points.

iv) This kinematic model of the WMR can overcome the drawback that the radius of the path generalized for the WMR's reference point may be less than the minimum turning radius of curvature determined by the WMR's steering mechanism.

Chapter 3

FUZZY LOGIC BASED NAVIGATION CONTROLLER FOR AN AUTONOMOUS MOBILE ROBOT

Drive Configuration
Types of Maneuvers
Possible Collision Free Path

FUZZY LOGIC BASED NAVIGATION CONTROLLER FOR AN AUTONOMOUS MOBILE ROBOT

This controller combines the path planning and trajectory following as an integrated and coordinated unit so that it executes maneuvers such as docking and obstacle avoidance on – line.

This main issue in wheeled mobile robot includes mission planning, navigation, maneuvering and manipulation. After the wheeled mobile robot receives commands for submissions from mission planner, the navigation controller takes over the responsibility to automatically drive the vehicle towards the goal.

These are two driving configurations in wheeled mobile robot

i) Steer drive
ii) Differential drive

STEER DRIVE CONFIGURATION – The steer drive uses two driving wheels to make the vehicle move forward and backward. The heading angle is controlled by an independent steering mechanism but due to physical constraints, this configuration cannot make turning in a a small radius. So, this is a shortcoming.

DIFFERENTIAL DRIVE CONFIGURATION – The differential drive configuration has two independent driving wheel parallel to each other. Their speed can be controlled separately. It is able to drive the vehicle forward and backward, as well as steer its heading angle by differentiating their speed. It has a capability of making a turning on the spot, i.e., small radius turning.

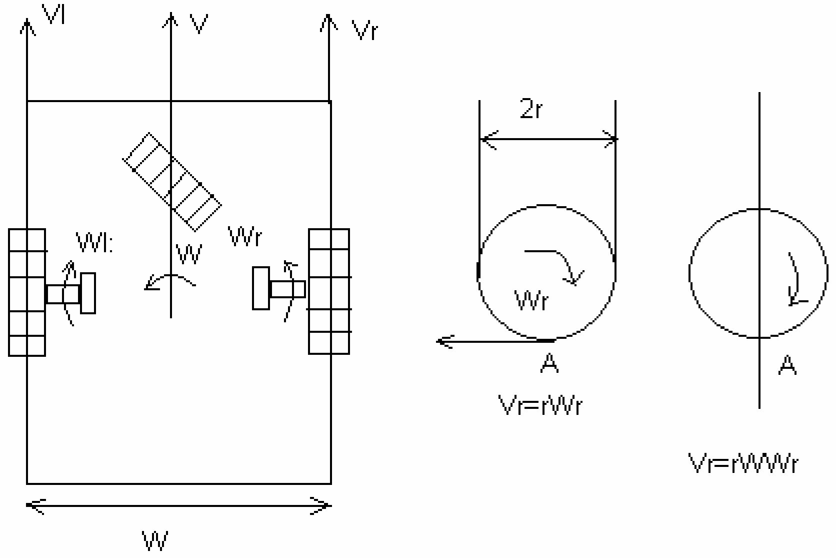


Fig:-3.1

$$\mathbf{V(t) = \tfrac{1}{2}\,(V_r + V_l)} \qquad \& \qquad \mathbf{\omega(t) = 1/r} \times \mathbf{(V_r - V_l)}$$

V(t) => Linear velocity of the vehicle

Since the differential drive configuration is symmetric about both horizontal and vertical axis, navigation control only requires design in one moving direction and should be applied to either moving forward or moving backward.

If we consider only the situation of moving forward, the following **maneuvers** are essential for successful navigation.

(i)APPROACH THE GOAL (AG) MANEUVER:-

The purpose of this maneuver is to locate the goal with respect to the mobile base.

The radius of curvature **'R'** is known that when mobile robot moving along a curve, there will be a centrifugal force, $\mathbf{F_c}$ associated with its turning movement.

$\mathbf{F_c = MV^2 / G}$ where **M** = Mass of mobile robot, **V** = Linear velocity

$\mathbf{F_c = MV^2 / R \leq G_{mf}}$ $\mathbf{G_{mf}}$ = Frictional force

If the mobile robot moves with its highest speed , then minimum radius of curvature $\mathbf{(R_{min}) \geq M / G_{mf} V_m^2}$

(ii)DOCKING MANEUVER (DK):-

Docking angle = $\boldsymbol{\beta}$, $\boldsymbol{\alpha}$ = Heading angle

The purpose of this maneuver is to approximately steer the mobile base such that it can move along the turning circle to approach the goal.

INWARD STEERING **OUTWARD STEERING**

$$L_c = \sqrt{R^2 + l^2 - 2Rl \sin\varphi} \quad , \quad \psi = \cos^{-1}\left(\frac{l - R\sin\varphi}{l_c}\right)$$

$$\rho = \sin^{-1}(R / l_c)$$

If the mobile base is correctly driven, then $(\Theta + \psi)$ or $(|\Theta| - \psi)$ should be 90^0 and l_c should be equal to R.

If $(\Theta + \psi) > 90^0$ or $l_c > R$ then mobile base must be steered inward.

If $(\Theta + \psi) < 90^0$ or $l_c < R$ then mobile base must be steered outward.

If $(|\Theta| - \psi) > 90^0$ or $l_c > R \Rightarrow$ Inward steering

If $(|\Theta| - \psi) < 90^0$ or $l_c < R \Rightarrow$ Outward steering

(iii)STAY ON THE PATH MANEUVER (SP):-

The wheeled mobile robot (WMR) needs moving along a physically constrained path, these physical constraints limit the motion from one side or from both sides.

The three sub – maneuvers in it are "**Stay – to – the – Right – side – of – the - sidewalk" (SRS), "Stay – to – the – Left – side – of – the – sidewalk" (SLS), "Keep – off – Both – sides" (KOB)**.

Their common feature is using sensor data obtained from certain sides of mobile base to steer its motion.

(iv)AVOID OBSTACLES (AO) MANEUVER:-

This maneuver is used to detect whether or not there is an obstacle on the road. Denote a safe distance between the possible obstacle and the front of the mobile base by $\mathbf{R_s}$ and a restricted distance by $\mathbf{R_r}$. Between $\mathbf{R_r}$ and $\mathbf{R_s}$ is a transition zone. If there is no object in front of the mobile base detected within the range of $\mathbf{R_s}$, then the mobile base will move straight ahead; otherwise the **AO** maneuver will take charge.

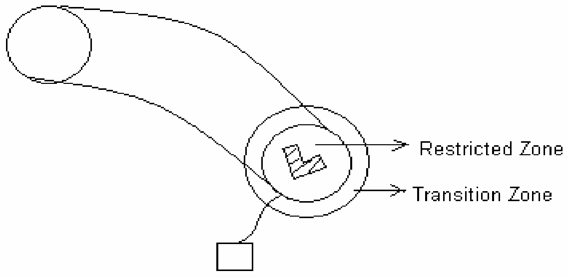


Fig:-3.4

POSSIBLE COLLISION FREE PATH

Thus a fuzzy – logic based controller plans the path trajectory for a wheeled mobile robot (WMR). The driving mechanism reacts immediately to perceived sensor data as the mobile robot navigates through the world.

We have applied this fuzzy – logic analysis for the path planning of our wheeled mobile robot (WMR). From the fuzzy – logic analysis, we got the velocities of the left and right wheels of the wheeled mobile robot (WMR) and from that we can calculate the time required to complete a U – turn on a given path configuration by the wheeled mobile robot (WMR).

Chapter 4

FUZZY LOGIC APPLICATION ON WHEELED MOBILE ROBOT

Traveling of Robot in an arena
Left Wheel **Velocity**
Right Wheel Velocity

FUZZY LOGIC APLLICATION IN WHEELED MOBILE ROBOT

Let, in a certain path, the front obstacle, left obstacle and right obstacle are at a distance of 5.8 m, 6m and 2.5 m from a wheeled mobile robot respectively.

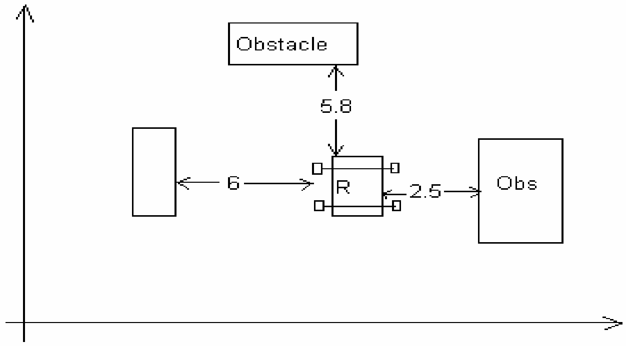


Fig:-4.1

For that particular path the left obstacle distance, right obstacle distance and front obstacle distance curves are given below.

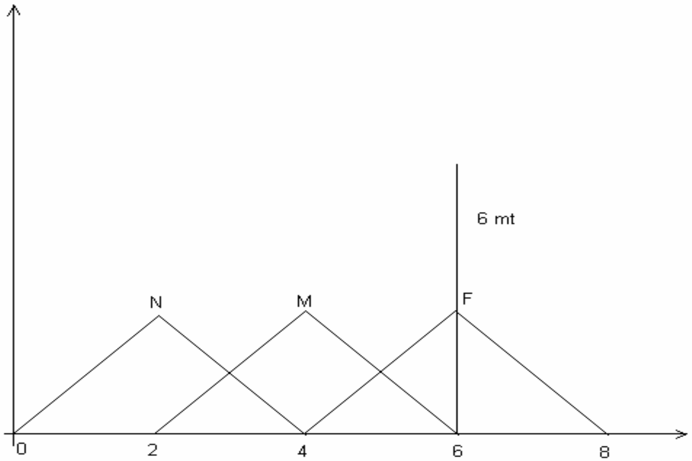


Fig:-4.2

Left Obstacle Distance Curve

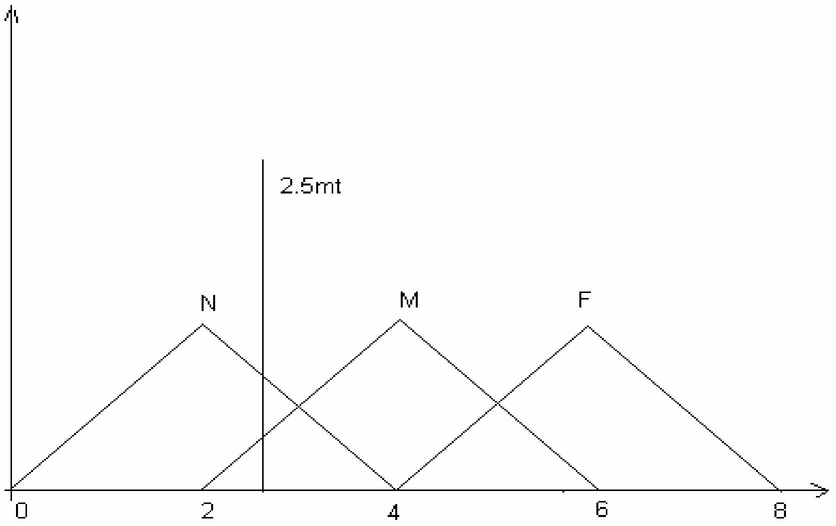


Fig:-4.3

Right Obstacle Distance Curve

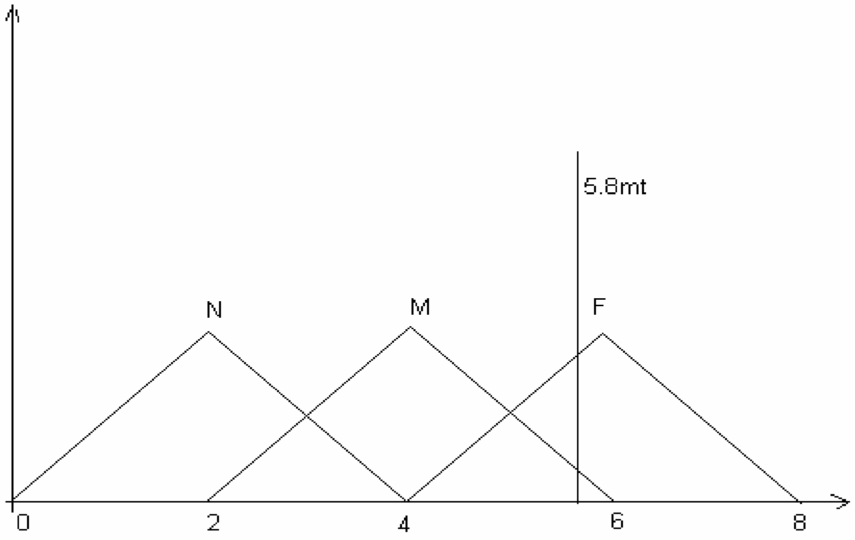


Fig:-4.4

Front Obstacle Distance Curve

From those there are several probable combination of velocity we can find,

	(1)	**(2)**	**(3)**	**(4)**
LOD ->	**F**	**F**	**F**	**F**
ROD ->	**N**	**M**	**M**	**N**
FOD ->	**F**	**F**	**M**	**M**

From the combination, we can find out whether the wheels should move slow, medium or fast.

	(1)	**(2)**	**(3)**	**(4)**
LWV ->	**M**	**F**	**M**	**S**
RWV ->	**M**	**F**	**F**	**F**

From the above combination of left wheel and right wheel velocity of wheeled mobile robot we can obtain the Left Wheel Velocity and Right Wheel Velocity curves.

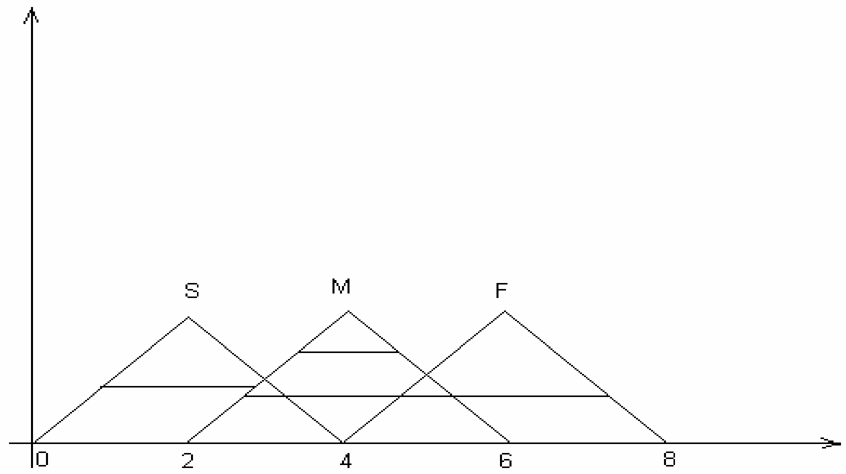


Fig:-4.5

Left Wheel Velocity Curve

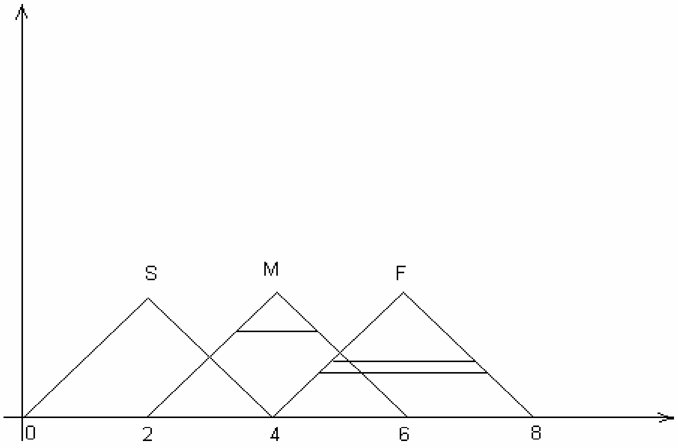


Fig:-4.6

Right Wheel Velocity Curve

The area under the curve shown, the left and right wheel velocity of wheeled mobile robot for different combination, we can obtain the right and left wheel velocity by obtaining the centre of gravity of the area under the curve.

Left Wheel Velocity (LWV):-

$$V_l = \frac{0.4/2\text{x}(4+2.4)\text{x}2 + 0.7/2\text{x}(4+1.2)\ \text{x}\ 4 + 0.3/2\text{x}(4+2.8)\text{x}4 + 0.3/2\text{x}0.3/2\text{x}(4+2.8)\text{x}6}{0.4/2\text{x}(4+2.4) + 0.7/2\text{x}\ (4+1.2) + 0.3/2\text{x}(4+2.8) + 0.3/2\text{x}(4+2.8)}$$

= 3.84 m/s

= 3.84 x 3.6 = 15 km/hr

Right Wheel Velocity (RWV):-

$$V_r = \frac{0.4/2\text{x}(4+2.4) \text{ x } 6 + 0.7/2\text{x}(4+1.2) \text{ x } 4 + 2 \text{ x } 0.3/2\text{x}(4+2.8) \text{ x } 6}{0.4/2\text{x}(4+2.4) + 0.7/2\text{x}(4+1.2) + 2 \text{ x } 0.3/2\text{x}(4+2.8)}$$

= 5.3 m/s

= 5.3 x 3.6 = 20 km/hr

So, V_l = 15 km/hr and V_r = 20 km/hr

Now if we consider a whole U – turn

Θ	**LWV**	**RWV**	**Time required**
10^0	**15 km/hr**	**20 km/hr**	**1.24 secs**
20^0	**15 km/hr**	**20 km/hr**	**2.5 secs**
30^0	**15 km/hr**	**20 km/hr**	**3.75 secs**
60^0	**15 km/hr**	**20 km/hr**	**7.5 secs**
90^0	**15 km/hr**	**20 km/hr**	**11.25 secs**
120^0	**15 km/hr**	**20 km/hr**	**15 secs**
180^0	**15 km/hr**	**20 km/hr**	**22.5 secs**

Calculation:-

We know, $\omega = (V_l - V_r) / B$ **[B = 10 m = wheel track]**

$= (5 \times 1000) / (3600 \times 10)$

$= 0.14$ **rad/sec**

Again, we know,

$\Theta = \omega l$

$\Rightarrow \quad t = \Theta / \omega$

For, $10^0 \Rightarrow \quad \Theta = \pi/180 \times 10$

$= 0.1745$ **rad**

So $\quad t = \Theta / \omega$

$= 0.1745/0.14$

$$\boxed{t = 1.24 \text{ secs}}$$

In this way, we can calculate all the time required for different angle of wheel mobile robot in turns.

Chapter 5

TYPES OF DRIVES

Differential Drive
Synchronous Drive

TYPES OF DRIVES

DIFFERENTIAL DRIVE:-

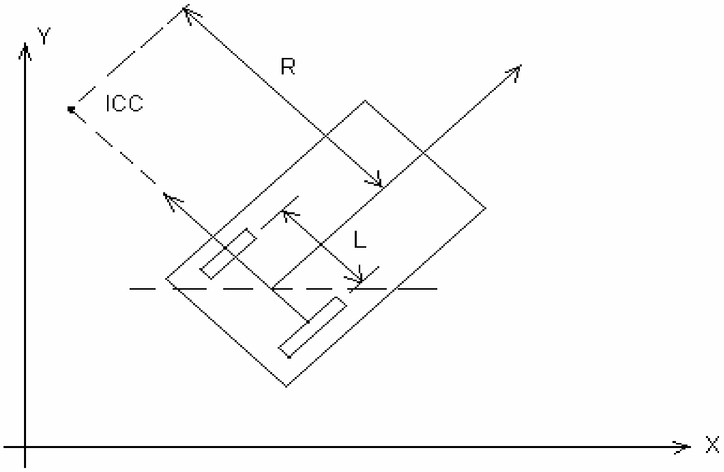


Fig:-5.1

Here ICC=Instantaneous centre of curvature

ICC=(x-Rsin Θ,y+R cos Θ)

V(r,t)=linear velocity of right wheel

V(l,t)= linear velocity of left wheel

r=nominal radius of each wheel

R=instantaneous curvature radius of robot trajectory relative to midpoint axis

(R-L/2)=curvature radius of trajectory described by left wheel

(R+L/2)=curvature radius of trajectory described by right wheel

$W(t)=V(r,t)/(R+L/2)$

$W(t)=V(l,t)/(R-L/2)$

So, $W(t)=(V(r,t)-V(l,t))/L$

$R=L/2((V(r,t)+V(l,t))/((V(l,t)-V(r,t))$

$V(t)=W(t)R=1/2(V(r,t)+V(l,t))$

Kinematic model in robot frame

$$\begin{matrix} V(x,t) \\ V(y,t) \\ \dot{\Theta}(t) \end{matrix} = \begin{matrix} r/2 & r/2 \\ 0 & 0 \\ -r/L & -r/L \end{matrix} \begin{matrix} W(l,t) \\ W(r,t) \end{matrix}$$

where $W(r,t)$ & $W(l,t)$ are the angular velocities of right & left wheel

It is useful for velocity control.

Particular cases:-

1. V(l,t)=V(r,t)

   Then a straight line trajectory is obtained.

   V(l,t)=V(r,t)=V(t)

   W(t)=0, $\dot{\Theta}(t)=0$

2. V(l,t)=-V(r,t)

   Then a circular path with instantaneous centre of curvature(ICC) on the midpoint between drive wheels.

   V(t)=0

   W(t)=2/L * V(r,t)

SYNCHRONOUS DRIVE

In a synchronous drive robot, each wheels are capable of being driven & steered.

Typical configurations:-

1. 3 steered wheel arranged as vertices of an equilateral triangle often surmounted by a cylindrical platform.
2. All the wheels turn and drive in unison.

Steered wheel:-

The orientation of the rotation axis can be controlled .all the three wheels point in the same direction & turn at the same rate and it is achieved by use of complex collection of belts that physically link the wheels together.

The vehicles control the direction in which the wheels point and the rate at which they roll

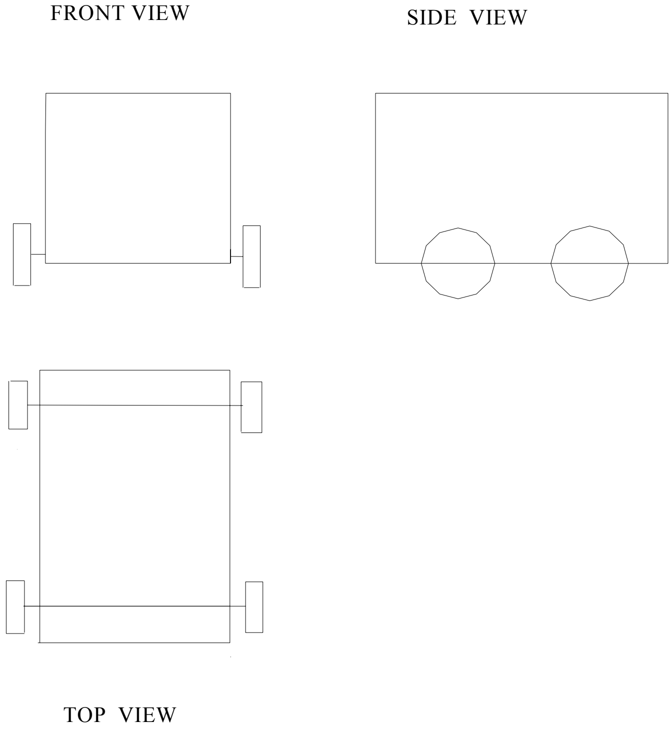
Because all the wheels remain parallel, the synchrodrive always rotate about the centre of robot.

Chapter 6

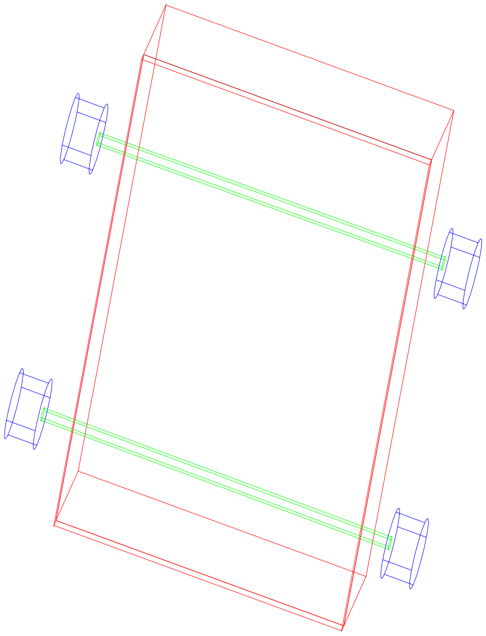
AUTOCAD DESIGN OF W.M.R

2-D MODEL OF W.M.R
3-D MODEL OF W.M.R

AUTOCAD DESIGN OF 2-D MODEL OF WMR:-



AUTOCAD DESIGN OF 3-D MODEL OF WMR:-



Chapter 7

LITERATURE SURVEY

Details of previous work on W.M.R

LITERATURE SURVEY

One paper presents the theoretical development of a complete navigation problem of an autonomous mobile robot. The situation for which the vehicle tries to reach the endpoint is treated using a fuzzy logic controller. The problem of extracting the optimized IF–THEN rule base is solved using an evolutionary algorithm. A new approach based on fuzzy concepts is presented in this paper to avoid any collision with the surrounding environment when this latter becomes relatively complex. Simulation results show that the designed fuzzy controller achieves effectively any movement control of the vehicle from its current position to its end motion and without any collision.

As regards the corridor and wall-following navigation problem, some control algorithms based on artificial vision have been proposed. In [1], image processing is used to detect perspective lines to guide the robot along the centre axis of the corridor. In [2], two lateral cameras mounted on the robot are used, and the optical flow is computed to compare the apparent image velocity on both cameras in order to control robot motion. In [3, 4], one camera is used to drive the robot along the corridor axis or to follow a wall, by **using** optic flow computation and its temporal derivatives. In [5], a globally stable control algorithm for wall-following based on incremental encoders and one sonar sensor is developed. In [6], a theoretical model of a **fuzzy** based reactive controller for a non holonomic **mobile** robot is developed. In [7], an ultrasonic sensor is used to steer an autonomous robot along a concrete path **using** its edged as a continuous landmark

Some other paper describes the development of PD, PID Computed-Torque (CT), and a PD digital motion controller
for the autonomous navigation of a **Wheeled** **Mobile** Robot (WMR) in outdoor environments. The controllers select the
suitable control torques, so that the WMR follows the desired path produced from a navigation

algorithm described in a previous paper. PD CT, PID CT, and PD digital controllers were developed using a linear system design procedure to select the feedback control signal that stabilizes the tracking error equation. The torques needed for the motors were computed by using the inverse of the dynamic equation for the WMR.

Simulation software was developed to simulate the performance and efficiency of the controllers. Simulation results verified the effectiveness of the controllers under different motion trajectories, comparing the performance of the three controllers shows that the PD digital controller was the best where the tracking error did not exceed .05 using 20 msec sample period. The significance of this work lies in the development of CT and digital controllers for WMR navigation, instead of robot manipulators. These CT controllers will facilitate the use of WMRs in many applications including defense, industrial, personal, and medical

Chapter 8

CONCLUSION

Results & Discussions
Conclusion

RESULTS & DISCUSSIONS

We got the 2-D and 3-D model of the wheeled mobile robot. We found out the left wheel velocity & right wheel velocity of robots in the arena where the robot is supposed to maneuver using fuzzy logic with the help of a c program.

We devised an arena for the robot to move avoiding all obstacles in its way to reach the target using MATLAB.Using this program we have been able to find the obstacle distances in the arena and the path to be tracked by the robot.

We also developed a kinematic model of the wheeled mobile robot, taking its path constraints & kinematic constraints into account.

CONCLUSION

We were dealing with the kinematic analysis of wheeled mobile robots.Our aim was to develop a kinematic model by path constraint & kinematic constraint.From the analysis of the established kinematic model,some factors which affect the path trekking problem of a wheeled mobile robot are taken into account.

Fuzzy logic navigation controller is used to determine the path followed by the autonomous wheeled mobile robot.this controller combines the path planning & trajectory following as an integrated & co-ordinated unit so that it executes maneuvers such as docking & obstacle avoiding maneuver.

A C program has been written to find the right & left wheel velocity of wheeled mobile robot in a particular arena.It is done with various possible combination of left obstacle distance,right obstacle distance & front obstacle distance in the arena where the robot moves to achieve the target.

A MATLAB program has been developed to make the arena within which the robot moves.in the arena ,the robot has to achieve the target avoiding the obstacles in its way.Like,if the left obstacle distance is less compared to the other obstacles,then the left wheel velocity is much higher than the right wheel velocity to turn the vehicle on the right side to achieve the target.

If two WMR's have the identical rigid body shape & driving system,but the reference point is selected in different positions,therefore their kinematic constraints & the capability of tracking path are totally different .In addition,the limit imposed by the steering mechanism is important ,which determines the minimum radius of curvature of the path of the reference point on WMR.

This thesis's kinematic model of the WMR can overcome the drawback that the radius of curvature of the path generated for the WMR's reference point may be less than the minimum turning radius of curvature which is determined by the WMR's steering mechanism ,& the kinematic model will be more practical & effective model for WMR's path following.

Chapter 9

APPENDIX

Program in C to find out LWV & RWV of the WMR in an arena

Program in MATLAB to find out the obstacle & avoid it in an arena

Program in C to find out LWV & RWV of the WMR in an arena

```
#include<stdio.h>
#include<conio.h>
main()
{
 float
fod,lod,rod,fn,ff,ln,lf,rn,rf,f1,f2,f3,f4,f5,f6,f7,f8,alf1,ars1,als2,arf2,alf3,arf3,als4,arf4,alf5,ars5,alf6,ars6,als7,arf7,als8,arf8,balf,bals,bars,barf,a1,a2,ylf,yls,yrs,yrf,A1,A2,A3,A4,A5,A6,LWV,RWV,VR;


alf1=0;ars1=0;als2=0;arf2=0;alf3=0;arf3=0;als4=0;arf4=0;alf5=0;ars5=0;alf6=0;ars6=0;als7=0;arf7=0;als8=0;arf8=0;
  printf("Enter the value of Velocity Range VR=");
  scanf("%f",&VR);
  printf("Enter the value of FOD=");
  scanf("%f",&fod);
if(fod<=25)
{ fn=1; ff=0; }
  else if(25<fod && fod<50)
    {
     fn=(-fod/25)+2;
     ff=(fod/25)-1;
    }
      else if(fod>=50)
      {  fn=0; ff=1;     }
   printf("The value of fn=%f\n",fn);
   printf("The value of ff=%f\n",ff);
 printf("Enter the value of LOD=");
```

```
scanf("%f",&lod);

if(lod<=25)
{ ln=1; lf=0; }
  else if(25<lod && lod<50)
  {
    ln=(-lod/25)+2;
    lf=(lod/25)-1;
   }
   else  if(lod>=50)
   { ln=0; lf=1; }
 printf("The value of ln=%f\n",ln);
 printf("The value of lf=%f\n",lf);

 printf("Enter the value of ROD=");
 scanf("%f",&rod);
 if(rod<=25)
{ rn=1; rf=0; }
 else if(25<rod && rod<50)
 {
    rn=(-rod/25)+2;
    rf=(rod/25)-1;
  }
 else  if(rod>=50)
 { rn=0; rf=1; }
 printf("The value of rn=%f\n",rn);
 printf("The value of rf=%f\n",rf);

  if(fn>0 && lf>0 && rf>0)
 {
  if((fn<lf)&&(fn<rf))
```

```
{

 printf("the smallest among (fn,lf and rf),fn=%f\n",fn);
  f1=fn; }
  else
   if(lf<rf)
  { printf("The smallest among (fn,lf and rf),lf=%f\n",lf);
    f1=lf; }
    else  {printf( "The smallest among (fn,lf and rf) ,rf=%f\n",rf);
     f1=rf;  }
     printf("f1=%f\n",f1);


      a1=((VR/2)*(2-f1))*f1;
      a2=f1*f1/2;
      alf1=ars1=a1+a2;
    printf(" Area of rule 1   alf=%f\n,ars=%f\n",alf1,ars1);
        }
  if(fn>0 && lf>0 && rn>0)
  {
    { if((fn<lf)&&(fn<rn))
 {
 printf("the smallest among (fn,lf and rn ),fn=%f\n",fn);
  f2=fn; }
  else
   if(lf<rn)
  { printf("The smallest among (fn,lf and rn),lf=%f\n",lf);
    f2=lf; }
    else  {printf( "The smallest among (fn,lf and rn),rn=%f\n",rn);
          f2=rn;  }
     printf("f2=%f\n",f2);   }
```

```
       a1=((VR/2)*(2-f2))*f2;
       a2=f2*f2/2;
      als2=arf2=a1+a2;
     printf(" Area of rule 2   als=%f\n,arf=%f\n",als2,arf2);
        }
      if(ff>0 && lf>0 && rf>0)
      {
{ if((ff<lf)&&(fn<rf))
{
  printf("the smallest among (ff,lf,rf),ff=%f\n",ff);
   f3=ff; }
   else
    if(lf<rf)
   { printf("The smallest among (ff,lf,rf),lf=%f\n",lf);
     f3=lf; }
     else  {printf( "The smallest among (ff,lf,rf),rf=%f\n",rf);
      f3=rf;  }
      printf("f3=%f\n",f3);
       }
        a1=((VR/2)*(2-f3))*f3;
        a2=f3*f3/2;
      alf3=arf3=a1+a2;
       printf(" Area of rule 3   alf=%f\n,arf=%f\n",alf3,arf3);
       }
      if(ff>0 && lf>0 && rn>0)
       {
{ if((ff<lf)&&(fn<rn))
{
  printf("the smallest among (ff,lf and rn),ff=%f\n",ff);
   f4=ff; }
   else
```

```
 if(lf<rn)
   { printf("The smallest among (ff,lf and rn),lf=%f\n",lf);
    f4=lf; }
    else  {printf( "The smallest among (ff,lf and rn),rn=%f\n",rn);
     f4=rn;  }
     printf("f4=%f\n",f4);
   }
     a1=((VR/2)*(2-f4))*f4;
     a2=f4*f4/2;
     als4=arf4=a1+a2;
     printf(" Area of rule 4   als=%f\n,arf=%f\n",als4,arf4);
          }
        if(fn>0 && ln>0 && rf>0)
       {
{ if((fn<ln)&&(fn<rf))
{
 printf("the smallest among (fn,ln and rf) ,fn=%f\n",fn);
  f5=fn; }
  else
   if(ln<rf)
   { printf("The smallest among (fn,ln and rf),ln=%f\n",ln);
    f5=ln; }
    else  {printf( "The smallest among (fn,ln and rf),rf=%f\n",rf);
     f5=rf;  }
     printf("f5=%f\n",f5);
     }
      a1=((VR/2)*(2-f5))*f5;
      a2=f5*f5/2;
     alf5=ars5=a1+a2;
     printf(" Area of rule 5   alf=%f\n,ars=%f\n",alf5,ars5);
      }
```

```
 if(ff>0 && ln>0 && rf>0)
      {
{ if((ff<ln)&&(ff<rf))
{
  printf("the smallest among (ff,ln,rf),ff=%f\n",ff);
   f6=ff; }
   else
    if(ln<rf)
   { printf("The smallest among (ff,ln,rf),ln=%f\n",ln);
     f6=ln; }
     else  {printf( "The smallest among (ff,ln,rf),rf=%f\n",rf);
      f6=rf;  }
      printf("f6=%f\n",f6);
      }
      a1=((VR/2)*(2-f6))*f6;
       a2=f6*f6/2;
      alf6=ars6=a1+a2;
      printf(" Area of rule 6   alf=%f\n,ars=%f\n",alf6,ars6);
       }
       if(fn>0 && ln>0 && rn>0)
   {
{ if((fn<ln)&&(fn<rn))
{
  printf("the smallest among (fn,ln and rn),fn=%f\n",fn);
   f7=fn; }
   else
    if(ln<rn)
   { printf("The smallest among (fn,ln and rn),ln=%f\n",ln);
     f7=ln; }
     else  {printf( "The smallest among (fn,ln and rn ),rn=%f\n",rn);
      f7=rn;  }
```

```
printf("f7=%f\n",f7);
      }
       a1=((VR/2)*(2-f7))*f7;
       a2=f7*f7/2;
      als7=arf7=a1+a2;
      printf(" Area of rule 7   als=%f\n,arf=%f\n",als7,arf7);
       }
      if(ff>0 && ln>0 && rn>0)
    {
  { if((ff<ln)&&(ff<rn))
  {
  printf("the smallest among (ff,ln and rn),ff=%f\n",ff);
   f8=ff; }
   else
    if(ln<rn)
   { printf("The smallest among (ff,ln and rn),ln=%f\n",ln);
     f8=ln; }
     else  {printf( "The smallest among (ff,ln and rn),rn=%f\n",rn);
      f8=rn;  }
      printf("f8=%f\n",f8);
     }
   {  a1=((VR/2)*(2-f8))*f8;
       a2=f8*f8/2;
      als8=arf8=a1+a2;    }
      printf(" Area of rule 8   als=%f\n,arf=%f\n",als8,arf8);
      }

    if((alf1>alf3)&&(alf1>alf5)&&(alf1>alf6))
          { balf=alf1;ylf=f1; }
             else
              if((alf3>alf5)&&(alf3>alf6))
```

```
{balf=alf3;ylf=f3;  }

                else
                  if(alf5>alf6)
                      {balf=alf5;ylf=f5 ; }
                         else
                          { balf=alf6;ylf=f6;}
    printf("Biggest area for left wheel velocity Fast=%f\n",balf);

      if((als2>als4)&&(als2>als7)&&(als2>als8))
         { bals=als2;yls=f2;}
           else
            if((als4>als7)&&(als4>als8))
              {bals=als4;yls=f4;  }
                else
                  if(als7>als8)
                      {bals=als7;yls=f7; }
                         else
                          { bals=als8; yls=f8; }
    printf("Biggest area for left wheel velocity Slow=%f\n",bals);
        if((ars1>ars5)&&(ars1>ars6))
              {bars=ars1; yrs=f1;  }
                else
                  if(ars5>ars6)
                      {bars=ars5; yrs=f5; }
                         else
                            {bars=ars6; yrs=f6;}
    printf("Biggest area for Right wheel velocity Slow=%f\n",bars);
       if((arf2>arf3)&&(arf2>arf4)&&(arf2>arf7)&&(arf2>arf8))
         {barf=arf2; yrf=f2; }
         else
```

```
           if((arf3>arf4)&&(arf3>arf7)&&(arf3>arf8))
                   { barf=arf3; yrf=f3;}
                    else
                     if((arf4>arf7)&&(arf4>arf8))
                       {barf=arf4; yrf=f4; }
                        else
                          if(arf7>arf8)
                               {barf=arf7;yrf=f7; }
                                  else
                                     {barf=arf8; yrf=f8;}
             printf("Biggest area for Right wheel velocity Fast=%f\n",barf);
                 if(bals>balf)
                 { if(ylf<=0.5)
                   { A1=yls*(VR/2)*(2-yls);
                     A2=((VR/2)/2)*yls*yls;
                     A3=(VR/2)*ylf;
                     A4=((VR/2)/2)*ylf*ylf;
                     LWV=(A1*((VR/2)/2)*(2-yls)+A2*((VR/2)/3)*(6-
2*yls)+A3*5*((VR/2)/2)+A4*(VR/2)*((3/2)+(2/3)*ylf))/(A1+A2+A3+A4);
                    }  else
                    { A1=yls*(VR/2)*(2-yls);
                     A2=((VR/2)/2)*yls*yls;
                     A3=(VR/2)*ylf;
                     A4=((VR/2)/2)*ylf*ylf;
                     A5=(VR/2)*(2-ylf)*((ylf-0.5));
                     A6=(1/2)*(ylf-((VR/2)/2))*(ylf-0.5);
                     LWV=(A1*((VR/2)/2)*(2-yls)+A2*((VR/2)/3)*(6-
2*yls)+A3*5*((VR/2)/2)+A4*((VR/2)*((3/2)+(2/3)*ylf))+A5*((VR/2)/2)*(4-
ylf)+A6*(VR/2)*((3/2)+(2/3)*(ylf-0.5)))/(A1+A2+A3+A4+A5+A6);}}
                     else
                       { if(yls<=0.5)
```

```
        { A1=ylf*(VR/2)*(2-ylf);
            A2=((VR/2)/2)*ylf*ylf;
            A3=(VR/2)*yls;
            A4=((VR/2)/2)*yls*yls;

LWV=(A1*((VR/2)/2)*(4+ylf)+A2*((VR/2)/3)*(3+2*ylf)+A3*((VR/2)/2)+A4*((VR/2)/3)*(3+yls))/(A1+A2+A3+A4);
            }  else
            { A1=ylf*(VR/2)*(2-ylf);
             A2=((VR/2)/2)*ylf*ylf;
             A3=(VR/2)*yls;
             A4=((VR/2)/2)*yls*yls;
             A5=(VR/2)*(2-yls)*((yls-0.5));
             A6=(1/2)*(yls-((VR/2)/2))*(yls-0.5);

LWV=(A1*((VR/2)/2)*(4+ylf)+A2*((VR/2)/3)*(3+2*ylf)+A3*((VR/2)/2)+A4*((VR/2)/3)*(3+yls)+A5*(VR/2)*(2-yls)+A6*((VR/2)/6)*(11-4*yls))/(A1+A2+A3+A4+A5+A6); }}

             if(bars>barf)
           { if(yrf<=0.5)
            { A1=yrs*(VR/2)*(2-yrs);
             A2=((VR/2)/2)*yrs*yrs;
             A3=(VR/2)*yrf;
             A4=((VR/2)/2)*yrf*yrf;
             RWV=(A1*((VR/2)/2)*(2-yrs)+A2*((VR/2)/3)*(6-2*yrs)+A3*5*((VR/2)/2)+A4*(VR/2)*((3/2)+(2/3)*yrf))/(A1+A2+A3+A4);
            }  else
            { A1=yrs*(VR/2)*(2-yrs);
             A2=((VR/2)/2)*yrs*yrs;
             A3=(VR/2)*yrf;
             A4=((VR/2)/2)*yrf*yrf;
```

```
            A5=(VR/2)*(2-yrf)*((yrf-0.5));
                      A6=(1/2)*(yrf-((VR/2)/2))*(yrf-0.5);
                      RWV=(A1*((VR/2)/2)*(2-yrs)+A2*((VR/2)/3)*(6-
2*yrs)+A3*5*((VR/2)/2)+A4*((VR/2)*((3/2)+(2/3)*yrf))+A5*((VR/2)/2)*(4-
yrf)+A6*(VR/2)*((3/2)+(2/3)*(yrf-0.5)))/(A1+A2+A3+A4+A5+A6);}}
                      else
                         { if(yrs<=0.5)
                     { A1=yrf*(VR/2)*(2-yrf);
                      A2=((VR/2)/2)*yrf*yrf;
                      A3=(VR/2)*yrs;
                      A4=((VR/2)/2)*yrs*yrs;

RWV=(A1*((VR/2)/2)*(4+yrf)+A2*((VR/2)/3)*(3+2*yrf)+A3*((VR/2)/2)+A4*((VR/2)/3)*(3+
yrs))/(A1+A2+A3+A4);
                      }  else
                      { A1=yrf*(VR/2)*(2-yrf);
                      A2=((VR/2)/2)*yrf*yrf;
                      A3=(VR/2)*yrs;
                      A4=((VR/2)/2)*yrs*yrs;
                      A5=(VR/2)*(2-yrs)*((yrs-0.5));
                      A6=(1/2)*(yrs-((VR/2)/2))*(yrs-0.5);

RWV=(A1*((VR/2)/2)*(4+yrf)+A2*((VR/2)/3)*(3+2*yrf)+A3*((VR/2)/2)+A4*((VR/2)/3)*(3+
yrs)+A5*(VR/2)*(2-yrs)+A6*((VR/2)/6)*(11-4*yrs))/(A1+A2+A3+A4+A5+A6); }}

              printf("LWV=%f\nRWV=%f",LWV,RWV);

               getch();
              }
```

INPUT OF THE PROGRAM:-

```
Enter the value of Velocity Range VR=50
Enter the value of FOD=40
The value of fn=0.400000
The value of ff=0.600000
Enter the value of LOD=20
The value of ln=1.000000
The value of lf=0.000000
Enter the value of ROD=40
The value of rn=0.400000
The value of rf=0.600000
```

OUTPUT OF THE PROGRAM:-

```
the smallest among (fn,ln and rf) ,fn=0.400000
f5=0.400000
 Area of rule 5   alf=16.080000
,ars=16.080000
The smallest among (ff,ln,rf),rf=0.600000
f6=0.600000
 Area of rule 6   alf=21.180000
,ars=21.180000
The smallest among (fn,ln and rn ),rn=0.400000
f7=0.400000
 Area of rule 7   als=16.080000
,arf=16.080000
The smallest among (ff,ln and rn),rn=0.400000
f8=0.400000
```

```
Area of rule 8   als=16.080000
,arf=16.080000
Biggest area for left wheel velocity Fast=21.180000
Biggest area for left wheel velocity Slow=16.080000
Biggest area for Right wheel velocity Slow=21.180000
Biggest area for Right wheel velocity Fast=16.080000
LWV=41.244446
RWV=32.599998
```

Program in MATLAB to find out the obstacle & avoid it in an arena

```
% ---  imaqhwinfo
%------- info = imaqhwinfo('winvideo')
%------------ dev_info = imaqhwinfo('winvideo')

 for kt1 = 1:2

vid = videoinput('winvideo');
%preview(vid)
%closepreview(vid)

set(gcf,'doublebuffer','on');

start(vid)

while(vid.FramesAcquired<=3)
    data = getdata(vid,2);
    diff_im = imabsdiff(data(:,:,:,1),data(:,:,:,2));
    I=data(:,:,:,1);
   % imshow(diff_im);
   % imshow(I);
end

stop(vid)

%I = imread('d:\t1\t2\test3.jpg');
% imshow(I);
```

```
whos;
J = rgb2gray(I);
%figure, imshow(I), figure, imshow(J);
for k = 1:288
   for l=1:352
   N(k,l)= J(k,l);

     if N(k,l)== 255
         N(k,l)=0;
     else
        N(k,l)=255;
     end

   end
end
%imshow((J));

BW_filled = im2bw(N, graythresh(N));
bw = imfill(BW_filled,'holes');

%imshow(I);
L = bwlabel(bw);
%--- imshow(L == 1)
%--- title('Object 8')
s = regionprops(L, 'Centroid');
ss= regionprops(L, 'Area');
%--- imshow(bw)
hold on
pp=1
% savefile = 'd:\t1\t2\dal1.txt';
```

```
% save(savefile, 'pp', '-ASCII');
fid = fopen('d:\t1\t2\exp.txt','w');
for k = 1:numel(s)
   c = s(k).Centroid;
   d = ss(k).Area;
   if  d > 100
   fprintf(fid,'%6.2f \n',pp);
   text(c(1), c(2), sprintf('%d', pp), ...
       'HorizontalAlignment', 'center', ...
       'VerticalAlignment', 'middle');
   pp=pp+1;

   end
end
fclose(fid)
hold off

% fid = fopen('d:\t1\t2\exp.txt','w');
% fprintf(fid,'%6.2f \n',pp);
% fclose(fid)

% savefile = 'd:\t1\t2\dal.txt';
% % pp=2;
% save(savefile, 'pp', '-ASCII')
% ----------imshow(J);
J1 = imfill(J);
% imshow(J);
% J(0,0)=0;
% J(0,1)=0;
% J(1,0)=0;
```

```
for k = 2:287   %start number(1+1)one more and end number one less (288-1)
   for l=2:351 %start number(1+1)one more and end number one less (352-1)
   N(k,l)= J(k,l);
     dd1=abs(J(k,l)-J((k-1),l));
     dd2=abs(J(k,l)-J((k-1),(l-1)));
     dd3=abs(J(k,l)-J(k,(l-1)));
     dd4=abs(J(k,l)-J((k+1),(l-1)));
     dd5=abs(J(k,l)-J((k+1),l));
     dd6=abs(J(k,l)-J((k+1),(l+1)));
     dd7=abs(J(k,l)-J(k,(l+1)));
     dd8=abs(J(k,l)-J((k-1),(l+1)));
     dd9=(dd1+dd2+dd3+dd4+dd5+dd6+dd7+dd8)/8;
     if (dd9 < 3)
        N(k,l)=0;
        NB1(k,l)=0;
     else
        N(k,l)=255;
         NB1(k,l)=255;
     end

   end
end


   for l=1:352
%        if (N(2,l)== 0)
%        N(1,l)=0;
%        end
      if (N(286,l)== 0)
      N(288,l)=0;
      end
   end
```

```
for k=1:288
%         if (N(k,2)== 0)
%         N(k,1)=0;
%         end
%         if (N(k,350)== 0)
%         N(k,352)=0;
%         end
    end

% BW_filled = im2bw(N, graythresh(N));
% bw = imfill(BW_filled,'holes');
% --------imshow(N);
BW_filled = im2bw(N, graythresh(N));
bw = imfill(BW_filled,'holes');
whos

 for l=1:352
%         if (N(2,l)== 0)
%         N(1,l)=0;
%         end


        bw(288,l)=1;
    end
% N1 = edge(J,'sobel');
% N1 =edge(J,'sobel',thresh);
% N1 = edge(J,'prewitt', thresh);
for l=1:352
        if (bw(2,l)== 0)
        bw(1,l)=0;
        end
%         if (N(286,l)== 0)
```

```
%         N(288,l)=0;
%         end
end
bw1 = imfill(bw,'holes');
for l=1:352

        bw1(1,l)=1;


end


    for l=1:352
        if (bw1(2,l)== 0)
     bw1(1,l)=0;
        end
        if (bw1(286,l)== 0)
        bw1(288,l)=0;
        end
    end
    for k=1:288
        if (bw1(k,2)== 0)
        bw1(k,1)=0;
        end
        if (bw1(k,350)== 0)
        bw1(k,352)=0;
        end
    end
L = bwlabel(bw1);
% -----imshow(L == 1)
% -------title('Object 8')
s = regionprops(L, 'Centroid');
ss= regionprops(L, 'Area');
```

```
imshow(bw1)
hold on
pp=1
% savefile = 'd:\t1\t2\dal1.txt';
% save(savefile, 'pp', '-ASCII');
fid = fopen('d:\t1\t2\exp.txt','w');
for k = 1:numel(s)
   c = s(k).Centroid;
   d = ss(k).Area;
   if  d > 100
   fprintf(fid,'%6.2f, %6.2f, %6.2f, %6.2f  \n',pp,c(1),c(2),d(1));
   text(c(1), c(2), sprintf('%d', pp), ...
      'HorizontalAlignment', 'center', ...
      'VerticalAlignment', 'middle');
   pp=pp+1;

   end
end
fclose(fid)
hold off

% imshow(bw1);

% imshow(NB);
pause(0.01);
 end
% imshow(I);
```

OUTPUT OF THE PROGRAM:-

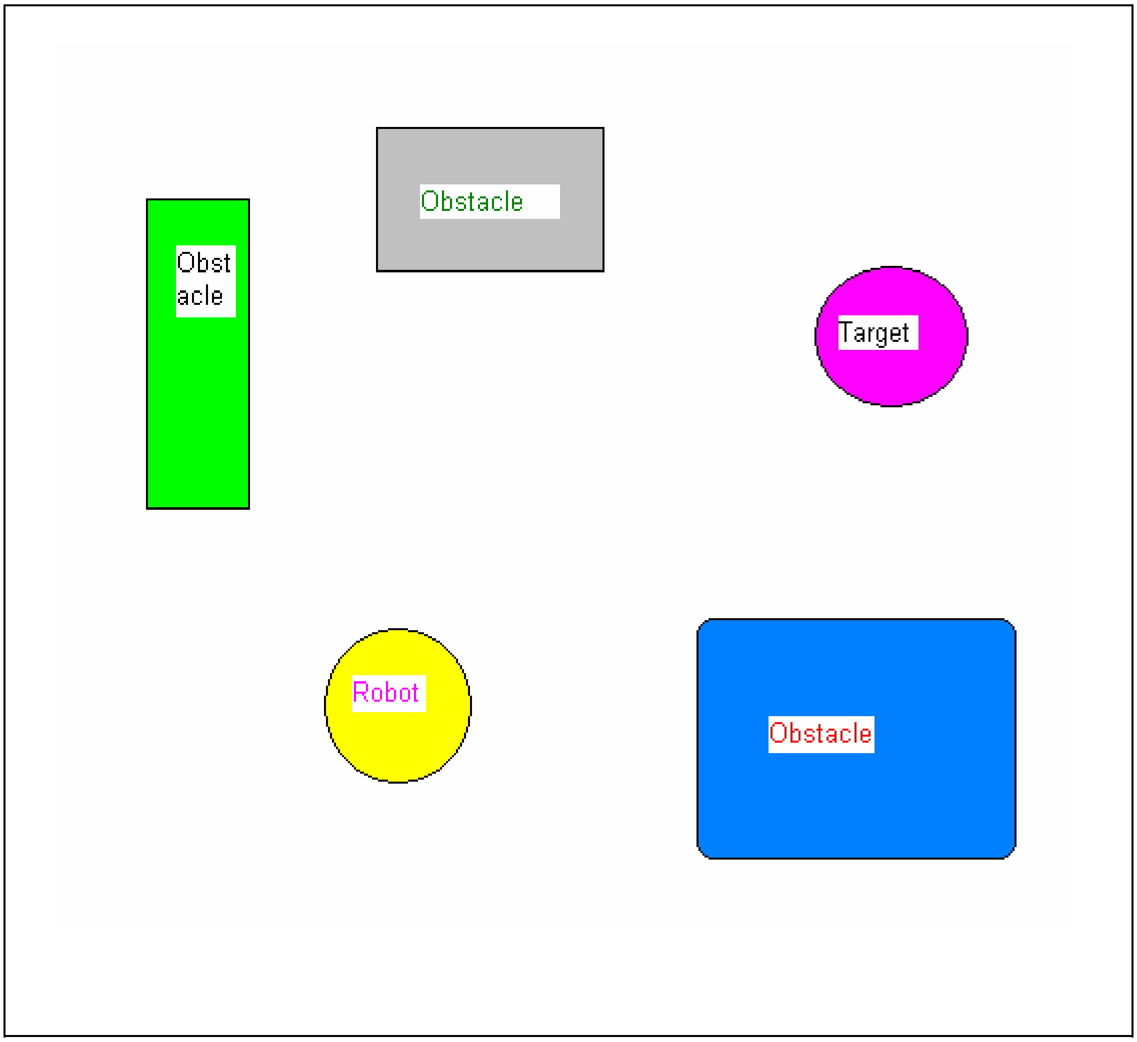


Fig:-9.1

REFERENCES:-

1. Y.Kanayama et al "A Locomotion Control Method for Autonomous Vehicles" IEEE conf. on Robotics and automation,1988:1315-1317.

2. W.L Nelson& I.J.Cox "Local Path Control for an Autonomous Wheeled Vehicle" IEEE conf. on Robotics and automation,1988:1504-1510

3. Zhang Minglu,Pang Shang Xian "Study on the Kinematic Model for a Wheeled Mobile Robot"Intelligence Machine Institute,PR China University

4. Vamsi Mohan Peri " Fuzzy LogicController for an Autonomous Mobile Robot"
Bachelor of Technology in Electrical and Electronics Engineering
Jawaharlal Nehru Technological University, India May, 2002

5. M. L. Corradini and G. Orlando, "Robust tracking control of **mobile robots** in the presence of uncertainties in the dynamical model," Journal of Robotic Systems, Vol. **18**, Issue 6, pp. 317-323, 2001

6. S.-F. Wu, J.-S. Mei, and P.-Y. Niu, "Path guidance and control of a guided **wheeled mobile** robot," Control Engineering Practice, Vol. **9**, Issue 1, pp. 97-105, 2001

7. M. B. Montaner and A. Ramirez-Serrano, "**Fuzzy** knowledge-based controller design for autonomous robot navigation," Expert Systems with Applications, Vol. **14**, Issue 1-2, pp. 179-186, 1998

8. M. Sugeno, M. Nishida, Fuzzy control of a model car, in: Fuzzy Sets and Systems, Elsevier, North Holland, 1985, pp. 103–113

9. O. Khatib, Real time obstacle avoidance for manipulators and mobile robots, International Journal of Robotics and Research 5 (1) (1986)

10. A. Safiotti, Fuzzy logic in autonomous robotics: behavior coordination, in: Proceedings of the Sixth IEEE International Conference on Fuzzy Systems, Barcelona, Spain, 1997, pp. 573–578.

11. F. Abdessemed, K. Benmahammed, E. Monacelli, Fuzzy image method to obstacle avoidance control, in: Proceedings of the IASTED International Conference on Artificial Intelligence and Soft Computing, Banff, Canada, July 17–19, 2002, pp. 603–608