A rule-based system for trajectory planning of an indoor mobile robot

Conference Paper · June 2010

DOI: 10.1109/SSD.2010.5585504

Siba Sharef
University of Technology, Iraq

Waladin Khairi Said
University of Technology, Iraq

A Rule-Based System for Trajectory Planning of an Indoor Mobile Robot

Siba M. Sharef, Waladin K. Sa'id, Farah S. Khoshaba

Control and Systems Engineering Department, UOT, Iraq.

Siba_sharef@yahoo.com
waladinksy@yahoo.com
farah_sami76@yahoo.com

Abstract— **In this paper, a software simulation model is developed for a two wheels driven mobile robot motion controller that can navigate the robot safely through an unknown environment. The work involves the design of a controller, which has four functions: motion control; obstacle avoidance; self-location; and path planning both global and local. The proposed controller is responsible for the mobile robot navigation after it generates a trajectory between start and goal points. Also it enables the robot to operate successfully in the presence of various obstacles present in any user built maps. The mobile robot is able to locate its position on any given map. The dynamic of the mobile robot is examined and the time constant of the two motors, which affects the direction of the mobile robot motion, is controlled. Obstacle avoidance is implemented with Fuzzy Logic Controller. The numerical experiments demonstrated that the indoor robot navigated successfully in tight corridors, avoided obstacles and dealt with a variety of world maps with various irregular wall shapes that were presented to it.**

I. INTRODUCTION

Automated Guided Vehicle (AGV) has been in existence since the 1950's. AGVs are driver-less industrial trucks, usually powered by electric motors and batteries and were applied mainly in warehouses, factories and mines [1]. Since then mobile robots were developed and applied in factories, military, planetary exploration, mining, woods, hospitals and for the disabled. The advancement in technology and in particular the development of on-board signal and data processing technology has added to the acceptance and wide spreading of its use. A wheeled mobile robot is a wheeled vehicle, which is capable of an autonomous motion (without external human driver) because it is equipped, for its motion, with sensors and actuators that are driven by an embarked computer [2]. They come in a variety of sizes, shapes, and capabilities, but they are all made up of the same important parts; each of these parts must be designed to work with other parts of the system as well as the other machines in a work cell [3].

One of the most important tasks in autonomous navigation is to create a suitable path planning in environments where robot navigates [4]. In moving between two points, the mobile robot must plan its path (globally and locally) as well as avoiding obstacles by temporarily deviating from the planned path. Furthermore, robot motion must be controlled which implies the strategy by which the platform approaches a desired location and the implementation of this strategy [5]. The robot should be capable of intelligent motion and action without requiring any external support while executing a given task. There are many algorithms for control of mobile robots described in literature [6]. However control of nonholonomic systems is a difficult problem. In tasks such as trajectory tracking or following a predetermined trajectory, it is necessary to control simultaneously the position and orientation of the robot, as well as its velocity.

This paper focuses on the application of FLC to move, orient and avoid obstacles of a mobile robot as it navigates in an indoor environment. For this end a differentially wheel driven mobile robot is considered and its kinematics and dynamics are modeled and the control action of the wheels speed is discussed. Global and local path planning and localization method are also examined and simulated.

II. MOBILE ROBOT DISCRIPTION

The mobile robot under study is made up of a rigid cart equipped with non-deformable wheels and it is moving on a horizontal plane, as shown in Fig. 1. The basic parts of the mobile robot responsible for the way it moves are the wheels. There are two basic classes of idealized wheels; namely the *conventional wheels* and the *Swedish wheels* [6]. The latter type of wheels is selected for the mobile robot shown in Fig. 1. The wheels are driven differentially by two DC motors.

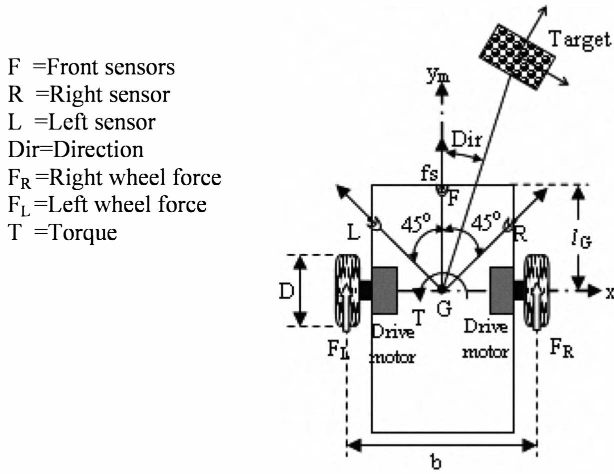


Fig. 1 Mobile robot with sensors

A sensory system is required to enable the robot to navigate in an unknown environment. A variety of sensors have been applied to mobile robots, this includes laser sensors, sonar range sensors, infrared proximity sensors, tactile sensors and cameras. Shaft encoders are also used to asses the relocation

978-1-4244-7534-6/10/$26.00 ©2010 IEEE

of the robot in comparison with the previous cycle. Furthermore, they are used to estimate the thereabouts of the robot position relative to the starting point or the borders of the environment while the robot navigates in the environment [7]. Fig. 1 shows the proposed sensor layout for the mobile robot under study. The figure shows three sonar sensing devices mounted in front, left and right of the robot's platform.

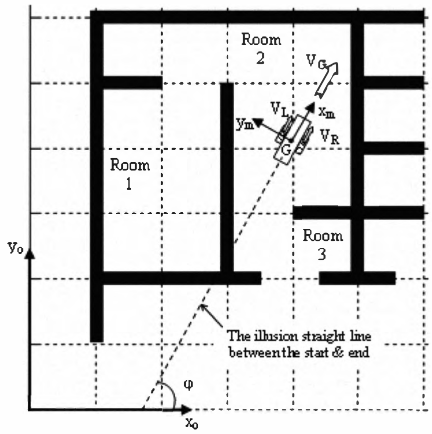


Fig. 2 Mobile robot in a typical indoors space

The position of the robot in the plane is described by the fixed frame (x_o,y_o,z_o) and the moving coordinates system (x_m,y_m,z_m), as shown in Fig. 2. The frame (x_o,y_o,z_o) is located at any convenient point in the building and the moving frame (x_m,y_m,z_m) is fixed at point G in the platform. The robot posture can be described in terms of the origin of frame (x_m,y_m,z_m) and its orientation angle φ (Fig. 2) both with respect to the base frame with origin at O.

III. MODELING of the MOBILE ROBOT

The kinematics and dynamics of the 2-DOF differentials drive vehicle shown in Fig. 1 is given in this section. The analysis assumes that the contact between the wheel and the ground is reduced to a single point of the plane.

The purpose of kinematics is to define the relationship between all known or measurable positions and velocities, and all quantities, which are computed by kinematics. With reference to Fig. 1 and Fig. 2, the linear velocity vector for the centre point of the differential-drive mobile robot is given by;

$$\begin{bmatrix}\dot{x}_G\\ \dot{y}_G\end{bmatrix}=\begin{bmatrix}V_G\cos\varphi\\ V_G\sin\varphi\end{bmatrix}=\frac{1}{2}\begin{bmatrix}\cos\varphi & \cos\varphi\\ \sin\varphi & \sin\varphi\end{bmatrix}\begin{bmatrix}V_L\\ V_R\end{bmatrix}$$
$$=\frac{D}{4}\begin{bmatrix}\cos\varphi & \cos\varphi\\ \sin\varphi & \sin\varphi\end{bmatrix}\begin{bmatrix}\omega_L\\ \omega_R\end{bmatrix} \quad (1)$$

As shown in Fig. 2, V_G is the linear velocity of the robot's centre (point G) along axis x_m. V_L and V_R are the velocities of the left and right side wheels, respectively. Similarly, the wheels angular velocities are ω_L and ω_R and D is the wheel diameter. $\dot{x}_G$ and $\dot{y}_G$ are the robot's velocity components along the fixed frame coordinates (x_o,y_o). The orientation of the robot expressed by φ is obtained by dividing the rows of equation (1), i.e., $\varphi=\tan^{-1}(\dot{x}_G/\dot{y}_G)$.

To construct workable system trajectories, the differential kinematics are required for point "fs" on the mobile platform. This point is the base of the front sonar sensor. Simple analysis shows that this differential kinematics is described by the following position/orientation vector [8];

$$\begin{bmatrix}\dot{x}_{fs}\\ \dot{y}_{fs}\\ \dot{\varphi}\end{bmatrix}=\frac{D}{4b}\begin{bmatrix}b\,c\varphi+2L_G\,s\varphi & b\,c\varphi-2L_G\,s\varphi\\ b\,s\varphi-2L_G\,c\varphi & b\,s\varphi+2L_G\,c\varphi\\ -2 & 2\end{bmatrix}\begin{bmatrix}\omega_L\\ \omega_R\end{bmatrix} \quad (2)$$

And the position vector is,

$$\begin{bmatrix}\dot{x}_{fs}\\ \dot{y}_{fs}\end{bmatrix}=\frac{D}{4b}\mathbf{M}\begin{bmatrix}\omega_L\\ \omega_R\end{bmatrix} \quad (3)$$

The value of the elements of matrix **M** follows directly from equation (2). The symbols c and s have been used instead of cos and sin. Equation (2) shows that the output velocities are nonzero even if only one wheel is rotating. For this reason this type of platform has the ability to change its orientation on the spot.

Mobile robot dynamics refer to the relationship between forces, torques and acceleration. Applying Newton's second law of motion to the differential-drive mobile robot shown in Fig. 1, the following is obtained;

$$\begin{bmatrix}\dot{V}_G\\ \ddot{\varphi}\end{bmatrix}=\begin{bmatrix}\frac{1}{m} & \frac{1}{m}\\ \frac{b}{J_r} & -\frac{b}{J_r}\end{bmatrix}\begin{bmatrix}F_R\\ F_L\end{bmatrix} \quad (4)$$

Where, F_R is the force exerted on the robot by the right wheel, F_L is the force exerted by the left wheel and b is the distance between the two wheels (Fig. 1). J_r and m are moment of inertia and mass of the mobile robot, respectively.

The basic actuation device of almost all mobile robots is the DC motor. To include the driving mechanism, the motor load is the wheel driving force times the wheel radius, equation (4) takes the following form;

$$\begin{bmatrix}\dot{V}_G\\ \ddot{\varphi}\end{bmatrix}=\frac{2}{D}\begin{bmatrix}\frac{1}{m} & \frac{1}{m}\\ \frac{b}{J_r} & -\frac{b}{J_r}\end{bmatrix}\left[k_1\begin{bmatrix}e_{aR}\\ e_{aL}\end{bmatrix}-k_2\begin{bmatrix}\omega_R\\ \omega_L\end{bmatrix}\right] \quad (5)$$

Where the constants k_1 and k_2 are functions of DC motor gear ratio, coil resistance, torque and back emf constants. Right and left wheel DC motor armature voltages are e_{aR} and e_{aL}, respectively. Equation (5) assumes that the mass and the moments of inertia of the castor and driving wheels are negligible. Fig. 3 shows the block diagram for simulating the mobile robot dynamics. The dynamic equation used is based on introducing the concept of total inertia seen at the load. Driving a mobile robot to any goal along any trajectory will eventually require varying ω_R and ω_L. This is done by varying the command signals ω_{cR} and ω_{cL}. The driving armature voltages are generated by the right and left wheel controllers. The driving motors used are unsymmetrical.

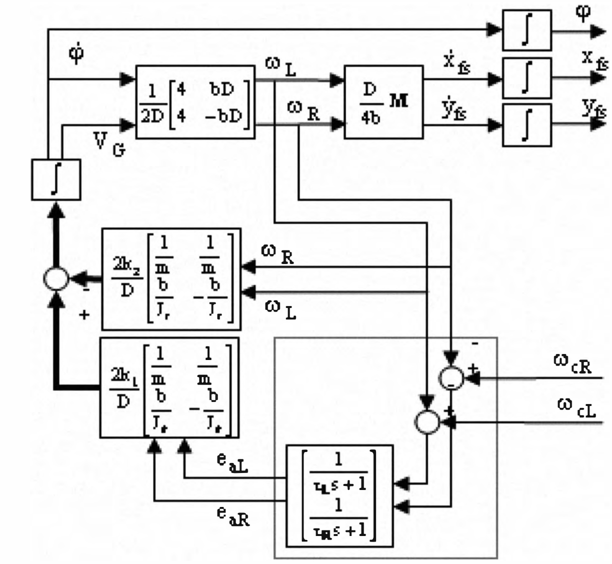


Fig. 3 Mobile robot and controller block diagram simulation

IV. MOBILE ROBOT CONTROLLER

The mobile robot controller has four basic functions: motion control; obstacle avoidance; self-location; and path planning (global and local). It is responsible for the mobile robot navigation after generating a trajectory between starting and goal points. Also it enables the robot to operate successfully in the presence of various obstacles present in any user built maps. Fig. 4 shows the controller structure scheme. Mobile robot dynamics block (Fig. 3) and measurements blocks have also been added. The noise block takes into account the uncertainty of the wheel diameters and the sensor errors. The next sections are devoted to the four functions of the controller.

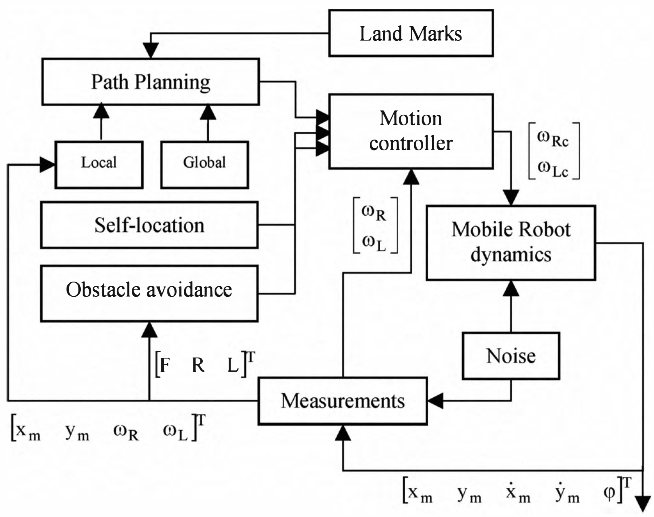


Fig. 4 Mobile robot controller structure scheme

A. Realization of the FLC for Obstacle Avoidance

This section is devoted to the implementation of the fuzzy logic reasoning to avoid obstacles where rules are put into operation to map inputs and outputs. Expert rules can be translated easily into IF-THEN statements used by fuzzy logic rules for any set of sensor inputs. Fig. 1 shows the proposed sensor layout for the mobile robot under study. Four fuzzy membership function inputs and two outputs are used, as shown in Fig. 5. The first input represents the reading of the left sensor (L); the second represents the reading of the front sensor (F), the third represents the reading of the right sensor (R) and the fourth input represents the direction of the robot according to the direction of the goal. The latter input is called the *heading* (Dir). Also, the first output represents the speed of the left wheel (V_L), and the second output represents the speed of the right wheel (V_R), that enables the robot to turn in both sides or move forward according to the difference between wheel velocities.

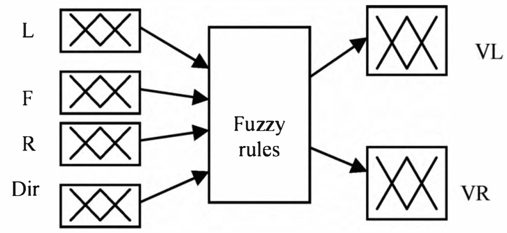


Fig. 5 Inputs and outputs of the fuzzy logic control

Once the inputs and outputs are identified and defined, the relationship between them must be established. The rule base was designed using human experience. The rules are translated into the fuzzy rules shown in Table 1.

TABLE 1
RULE-TABLE FOR OBSTACLE AVOIDANCE

		Right wheel			
	Dir	SL		SR	
F	R \ L	Rb	Rs	Rb	Rs
Fb	Lb	R	R	L	R
	Ls	F	F	L	F
Fs	Lb	R	R	L	R
	Ls	L	L	L	L

		Left wheel			
	Dir	SL		SR	
F	R \ L	Rb	Rs	Rb	Rs
Fb	Lb	L	L	R	F
	Ls	R	F	R	F
Fs	Lb	L	L	R	L
	Ls	R	R	R	R

Note: The symbols used refer to the following:
*For output variables: R: turn Right, F: go Forward and L: turn Left.

*For input sensors (R, L, F & Dir): Lb: Left big, Ls: Left small, Rb: Right big, Rs: Right small, Fb: Forward big, Fs: Forward small, SL: Small Left, SR: Small Right.

The MFs of inputs are shown in figs. (6a and b). Left and right wheel robot velocities output MFs are shown in Fig. 7. According to the rules in Table 1 if an obstacle is detected by R sensor then the following logic should be followed;
IF F is Fb AND L is Lb AND R is Rs AND Dir is DR THEN (VR is R AND VL is F) that means right wheel velocity should be bigger than left one hence the robot swerves left to avoid that obstacle. Fuzzy algorithm was implemented using MATLAB 7.0.

An execution panel was built using the Graphical User Interface (GUI) of MATLAB 7.0 [9]. The panel was used to insert the environment map. It was used to set the start and end positions of the mobile robot, specify its wheel's maximum speed and specify landmark points.

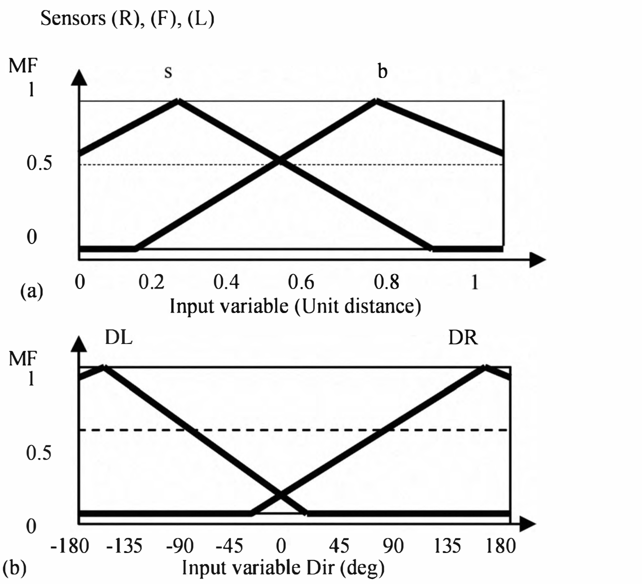


Fig. 6 Membership function of input variables (a) Sensors (R), (F), & (L); (b) Dir

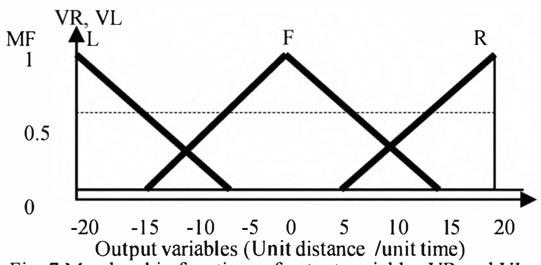


Fig. 7 Membership functions of output variables VR and VL

Briefly, the software that was developed [10] reads the map and the goal position (X_{goal}, Y_{goal}) first. Then it converts the environment map to a matrix of (40×40) elements. The matrix consists of 1's and 0's. The 1's represents the space and the 0's represents the wall's start and end points and the obstacles. The maximum (ω_{Rmax}, ω_{Lmax}) and the initial (ω_{R0}, ω_{L0}) angular velocities of right and left wheels are then read. Also, a number of landmark points are read. These landmarks are used to predefine a suitable path for the mobile robot to follow on its way to its goal. If an obstacle is detected, then FLC algorithm is applied to avoid it then it returns to its original path until the robot reaches its final destination [10].

B. Local-Path Planning

This method is also called on-line path planning. With this method there is no specific map stored in the computer and the user inputs the map only. So the mobile robot uses sensors to detect obstacles and find an appropriate path to the goal with the use of the FLC. After the map is drawn, the start and end points are specified then the program specifies a straight line between the initial position of the mobile robot and the target. Then the X and Y of all lines points (wall obstacle coordinates) are saved in arrays.

The mobile robot starts its way to the target by tracking a straight line between start and goal points. If an obstacle is detected (when a zero appears in the matrix (X,Y) , the FLC algorithm is applied to avoid the obstacle that is located in its way. Then a new straight line trajectory is generated to the target [10].

C. Global Path Planning

This method is called off-line path planning where the planning is based on a priori complete information about the environment stored in the controller processor. So the mobile robot plans and acquires (establishes) the path to the goal before it begins its journey.

Path planning starts by constructing an initial straight line between start and goal points. Using the stored map of the environment, if a wall is present in between start and goal points, a new fictitious goal point is generated based on logical reasoning. The procedure is repeated until the robot finds its way out of the room. The closing fictitious goal point now becomes the new start point and the process is repeated until the planned path attains its goal point. As the robot travels to its goal point, FLC obstacle avoidance is activated whenever obstacles are sensed [10].

D. Localization

Estimating the position of a robot based on sensor data is one of the fundamental problems of mobile robotics, which is called localization [11]. Two methods were used for locating the robot. The first one uses shaft encoders. The odometric measurement determines approximately the where about of the robot, since the system is an open loop and therefore is subjected to external disturbances. However it is useful in the sense that it provides information about the relocation of the robot in comparison with the previous cycle.

The second method uses the ultrasonic self location system, which gives a measure of the position. It uses an on-board ultrasonic transmitter and two receivers fixed appropriately in the ceiling of the environment to locate a point. It relies on the measurement of the Time-Of-Flight between the transmitter and receivers and the use of simple trigonometric relations. Such a system gives a measure of the actual position of the vehicle during its movement. The mobile is to check its position every (1 second) in order to determine its position with respect to a given map [12].

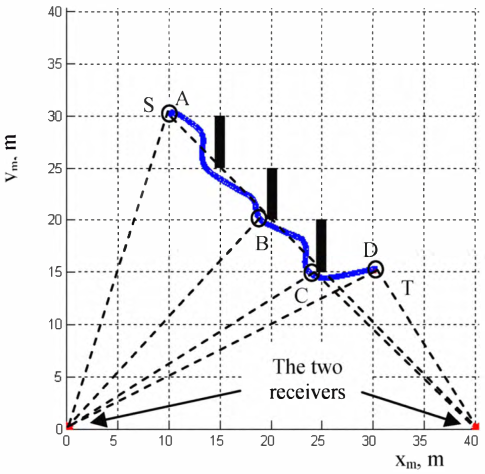
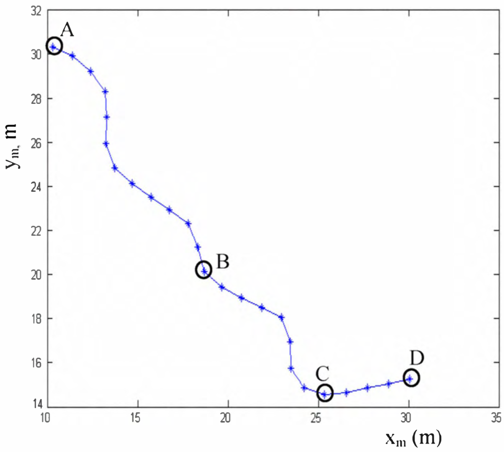


Fig. 8 The localization process for the mobile robot for a given map

In the simulation it is assumed that the calculations contain some error which cannot exceed certain values. A typical example of the localization procedure is shown in Fig. 8. The mobile robot through its journey from the beginning to the end point, defines its position on the map every 1 second. The trajectory of the mobile robot with defined location points is shown in Fig. 9. Points A, B, C and D are typical points.



Fig, 9 The trajectory of the mobile robot

V. NUMERICAL EXPERIMENTAL RESULTS

A series of numerical tests were carried out to test the ability of the mobile controller to deal with numerous circumstances. Obstacle avoidance aptitude procedure was checked by the two tests summarized in figs. (10) and (11). In the first test a path for the mobile robot was predefined with the presence of one obstacle. Fig. 10) shows the environment (thick black lines), start, target and two landmarks (L_1 and L_2) points. As can be seen from the figure, the robot travels in a straight line from points S to L_1. Then it continues its motion from points L_1 to L_2 in another straight line trajectory. However, a wall is detected (an obstacle) and accordingly the robot turns aside to avoid it and continues moving in a straight line. Finally, a straight line trajectory is followed until T point is successfully reached. Fig. 11 shows the case where two obstacles are present between points L_1 and L_2. The figure clearly shows that the mobile robot FLC successfully swerves the robot around the two obstacles. Also it can be seen that the robot always attempts to travel in straight line trajectories until the target is reached.

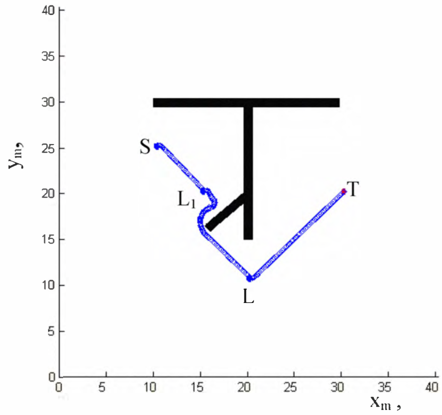


Fig. 10 The robot behavior with an obstacle in its trajectory

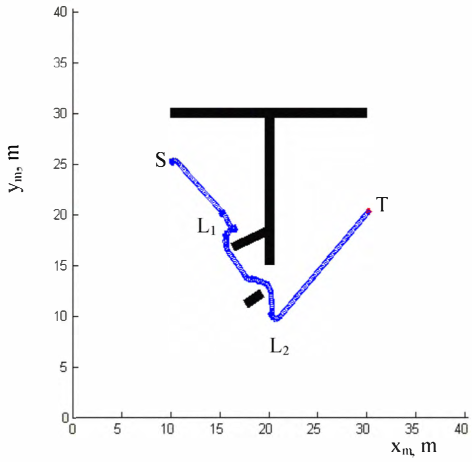


Fig. 11 The robot behavior with two obstacles in its trajectory

Local path planning mode of operation of the mobile controller is demonstrated by Figs. 12 and 13. The figures show two different missions of the mobile robot in the same environment. In Fig. 12, the robot travels through a passage outside the room from the start point S (10, 28) to the goal point T (36, 13). It can be clearly seen that the FLC has driven the robot successfully around the first room corner (20, 25). In the second mission shown in Fig. 13 the robot successfully move from point (5,20), which is outside the room, to the target (T), which is inside it.

The mission of the mobile robot shown in Fig. 14 is to go from a position in a passage to a target in room 2. The mobile robot starts its journey and reaches the goal in room 2 by traveling through room 1.

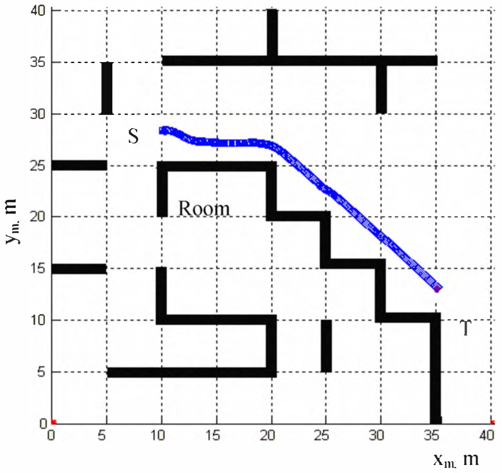


Fig. 12 The behavior of the mobile robot in a building

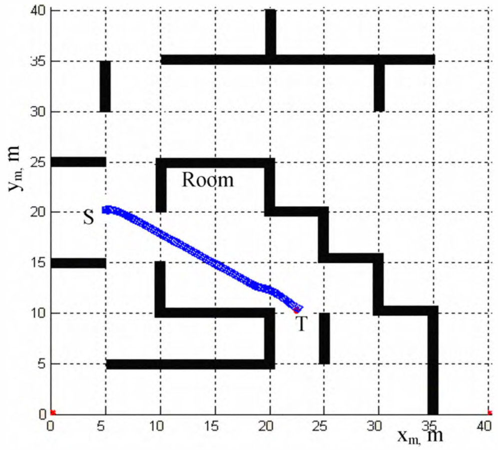


Fig. 13 The behavior of the mobile robot in the same building

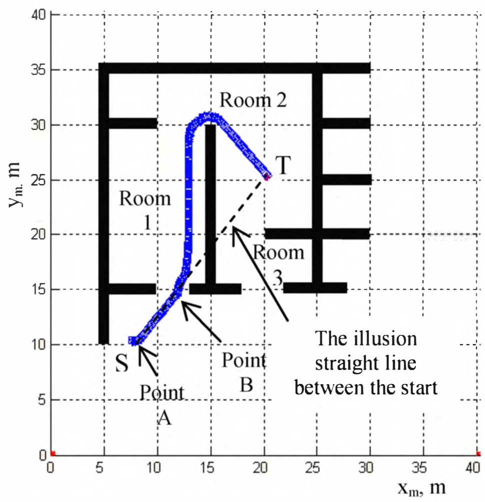


Fig. 14 The behavior of the mobile robot in an office like environment

This is because the robot always goes to the target in a straight line if there is no obstacle in between. However as it moves from point A to point B in a straight line it faces the wall between rooms 1 and 2. Hence it turns left and travels along the wall, since the wall-following method was adopted as was mentioned earlier. When the wall ends, the mobile robot goes to the goal again in a straight line. Fig. 15 shows the same mission however this time the mobile robot reaches the goal, by traveling through room 3 instead of 2. This is because the door of room1 is closed, so the robot turns to the right and enters room 3 then room 2.

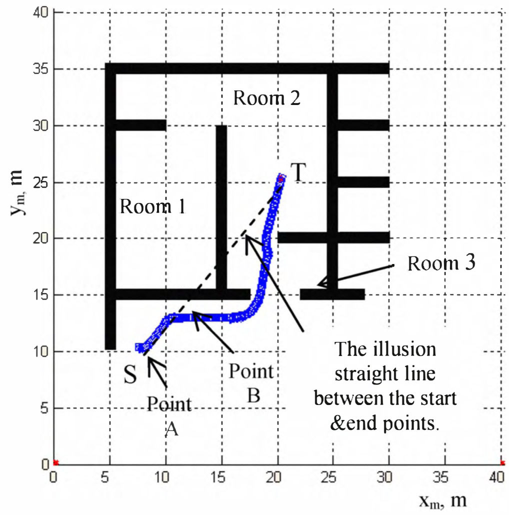


Fig.15 The same mission as in fig.14 but a different path is selected

The performance of the robot using global-path planning scheme was also tested. Fig. 16 presents a map of a building that consists of a passage leading to three rooms. The mobile robot must go from point (5, 25) in the middle of the passage to point T located in room1. Before motion starts, a proposed trajectory of motion is developed first, as shown in Fig. 16. Then after, the mobile robot will move following this trajectory and successfully reach the goal as shown in Fig. 17.

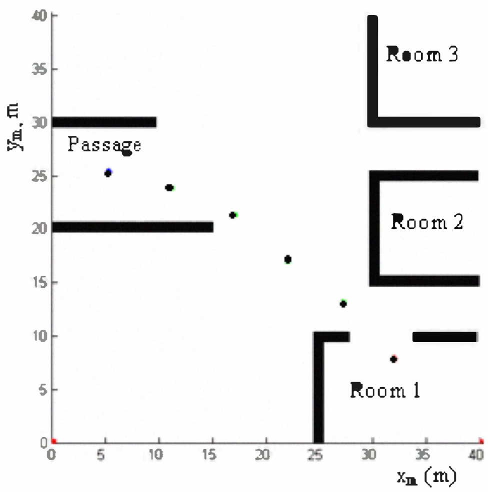


Fig.16 The initial path developed by the global-path planning scheme

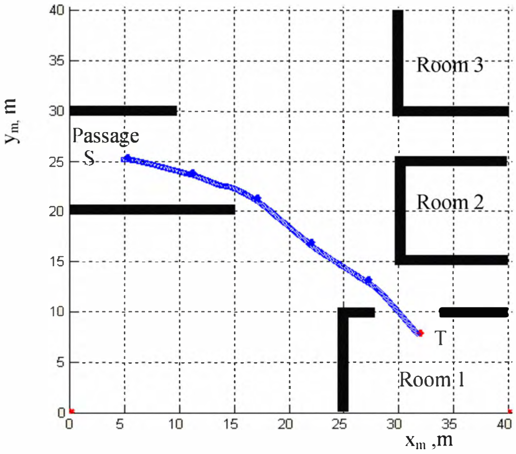


Fig. 17 Trajectory of the mobile robot in the global path planning mode

VI. CONCLUSIONS

In this paper we have presented a hierarchical controller for mobile robot for application in workshops. A model-based controller was designed to drive the mobile robot from start to goal points. Fuzzy inference methods have been applied in building the fuzzy controller for obstacle avoidance. The proposed trajectory planning and control methodology was successfully tested numerically.

The robot always attempts to move in a straight line between the start point and the goal point. When it faces an obstacle, it turns around and then it resumes its original trajectory. Local-path planning makes use of the sensor information. It is not necessary to define a reference trajectory prior to start of motion and store world maps in the memory of the mobile robot. However, global-path planning requires the workspace map be stored in the memory of the mobile robot. The robot begins its journey by planning its trajectory to the goal before motion begins. The effect of varying sensor range showed that it does not only affect the distance between the mobile robot and obstacles and walls, but may also affect the shape of the trajectory to the goal.

Finally, the numerical experiments demonstrated that the indoor robot navigated successfully in tight corridors, avoided obstacles and dealt with a variety of world maps with various irregular wall shapes that were presented to it.

REFERENCES

[1] J. Ashayeri, L. F. Gelders, and P. M. Van Looy, "Micro-Computer Simulation In Design of Automated Guided Vehicle Systems", *IEEE Trans.*, pp.37-48, 1985.

[2] D. Kortenkamp, R. P. Bonasso, and R. Muphy,"Artificial Intelligence and Mobile Robots", *American Association for Artificial Intelligence*, 1998.

[3] L. Heath, *Fundamentals of Robotics Theory and Applications*, Reston Publishing Company, 1985.

[4] B. L. Brumitt, R. C.Coulter, and R. Murphy, " Artificial Intelligence and Mobile Robots", *MIT press*, pp.3-20, 1998.

[5] J. Laumond, "Robot Motion Planning and Control", *LAAS-CNRS*, Toulouse, August 1997.

[6] C. Canudas de Wit, B. Siciliano, G. Bastin (Eds.), *Theory of Robot Control,* Springer, Berlin, Heilderberg, New York, 1996.

[7] D. Xiao, and B. K. Ghosh, "Sensor-Based Hybrid Position/Force Control of a robot Manipulator in an Uncalibrated Environment", Transactions on control system technology, *IEEE*, Vol.8, No.4, July 2000.

[8] E. Papadopoulos and J. Poulakakis,"Trajectory Planning and Control for Mobile Manipulator Systems", *Proceedings 8th IEEE Mediterranean Conference on Control & Automation (MED'00)*, July 17-19, 2000, Patras, Greece.

[9] MATLAB®, "The Language of Technical Computing-Getting Started with MATLAB®/ Version 7.0", WWW.mathworks.com.

[10] F. S. Khoshaba, "A Rule-Based System for Trajectory Planning of an Indoor Mobile Robot," M.Sc. Thesis, Control & Systems Eng. Dept., University of Technology, 2005.

[11] F. Dellaert, D. Fox, W. Burgard and S. Thrun, "Monte Carlo Localization for Mobile Robots (M CL)", *Proceedings IEEE conf. Robotics & Automation*, pp.1322-1328, 1966.

[12] C. Ferrari, E. Pagello, M. Voltolina, J. Ota, and T. Arai, "Multirobot Motion Coordination Using a Deliberative Approach", *Proceedings of the Second Euromicro Workshop on Advanced Mobile Robots (EUROBOT'97)*, October22-24, pp.96-103, 1997.