Steering of an Automated Vehicle in an Unstructured Environment

Sampath Kanakaraju, Sathish Shanmugasundaram, Ramesh Thyagarajan and Ernest L. Hall

Center for Robotics Research
Mail Location # 0072
University of Cincinnati
Cincinnati, Ohio 45221

ABSTRACT

The purpose of this paper is to describe a high-level path planning logic, which processes the data from a vision system and an ultrasonic obstacle avoidance system and steers an autonomous mobile robot between obstacles. The test bed was an autonomous robot built at University of Cincinnati, and this logic was tested and debugged on this machine. Attempts have already been made to incorporate fuzzy systems on a similar robot, and this paper extends them to take advantage of the robot's ZTR capability.

Using the integrated vision system, the vehicle senses its location and orientation. A rotating ultrasonic sensor is used to map the location and size of possible obstacles. With these inputs the fuzzy logic controls the speed and the steering decisions of the robot.

With the incorporation of this logic, it has been observed that that Bearcat II has been very successful in avoiding obstacles very well. This was achieved in the Ground Robotics Competition conducted by the AUVS in June 1999, where it traveled a distance of 154 meters in a 10ft wide path ridden with obstacles. This logic proved to be a significant contributing factor in this feat of Bearcat II

Keywords: Fuzzy, ZTR, Sonar

1. INTRODUCTION

Generally motion-planning algorithms for a mobile robot are developed based on the assumption that there are no changes in the surrounding environment. One of the goals in robotics is to endow robots with the ability to move and operate autonomously in an environment with unknown, perhaps moving obstacles. Robot navigation is described as the guiding of a mobile robot to a desired destination or along a desired path in an environment characterized by a terrain and a set of distinct objects, such as obstacles and landmarks. The robot must successfully navigate around obstacles, reach its goal and do so efficiently. Not only must a robot avoid colliding with an obstacle such as a rock, it must also avoid falling into a pit or ravine and avoid travel on terrain that would cause it to tip over.

The purpose of a path planner is to compute a path, i.e., a continuous sequence of configurations. The primary concern of path planning is to compute collision-free paths. Another key issue is the uncertainty problem. It is related to various sources of uncertainty affecting the actual robot (control and sensing errors, inaccurate models of the environment, etc.). Some approaches have been proposed to steer the robot in order to stay clear of some close obstacles, but they fail to plan maneuvers if they are necessary. In order to solve this problem, Laumond has proposed a three-stage algorithm. This method relies on seeking a collision free path for a holonomic robot and rebuilding it in order to take into account the nonholonomic (a wheel is a nonholonomic system; it can only move in a direction perpendicular to its axle) constraints. In this way a feasible path is created. During the last stage, the path is optimized. Using the same idea, the most widespread procedure is collision detection. In order to simplify it, usually the configuration space approach is used, which allows us to take into account the robot size [1].

Approaches to path planning for mobile robots can be broadly classified into two categories - those that use exact representations of the world and those that use discretized representations. The main advantage of discretization is that adjusting the cell size can control the computational complexity of path planning. In contrast, the computational complexity of exact methods is a function of the number of obstacles and/or the number of obstacle facets, which we cannot normally control [2]. In order to create an efficient method enabling to check if a planned path is admissible or not, the representation of obstacles is to be considered. The two main approaches that have been proposed for the representation of obstacles can be called as obstacle oriented and space oriented representations. The first approach approximates the shape of the obstacles in the form of polygons or other geometric figures while the second approach uses convex figures to represent areas that are occupied by the obstacles or not, for example, a probability grid representation. In the former approach, it is difficult to conceptualize the shape of the obstacle from the sensor data (especially ultrasonic sonars) and maintain the environment model, while in the probability grid model, it is difficult to extract necessary data to solve the task of robot localization. In this paper, we first begin with a brief description of the robot used. Then we describe the motion control of the robot and thus step into the logic, which forms the core of this paper. Finally, we give the scope of future work in this field and conclude with our acknowledgements.

2. THE ROBOT

The robot on which the following logic was successfully implemented is Bearcat II, the mobile robot built by the students at the Center for Robotics Research in the University of Cincinnati. The robot is basically built using off-the-shelf components (it has a frame built from 80-20 bars). It has two JVC cameras mounted on top to help in its objective of avoiding obstacles in its path, which is further buttressed by the presence of single rotating sonar. The main purpose is to compete in the International Ground Robotics Competition conducted by the AUVS each year.

The control equipment of the robot is shown in a schematic below [3]. As mentioned before, the major components of the system are vision guidance system, steering control system, obstacle avoidance system, Polaroid drive system, speed control system, power unit and supervisor control PC. The critical components of the robot are the obstacle avoidance and the steering systems.

Obstacle avoidance system consists of a single rotating transducer. Polaroid ultrasonic ranging system is used for the purpose of calibrating the transducer. An Intel 80C196 microprocessor and a circuit are used to process the distance calculations. The distance value is returned through a RS232 port to the control computer. A pulse of electronically generated sound is transmitted toward the target and the resulting echo is detected. The system converts the elapsed time into a distance value. The digital electronics generate the ultrasonic frequency. All the digital functions are generated by the Intel microprocessor. Operating parameters such as transmit frequency, pulse width, blanking time and the amplifier gain are controlled by the software supplied by Polaroid.

The drive system for the transducer consists of a DC motor and its control circuitry. With this arrangement the transducer is made to sweep and angle depending on the horizon (range between which we need detection). The loop is closed by an encoder feedback from an encoder. The drive hardware comprises of two interconnected modules, the Galil ICB930 and the 4-axis ICM 1100. The ICM 1100 communicates with the main motion control board the DMC 1030 through an RS232 interface. The required sweep is achieved by programming the Galil. By adjusting the Polaroid system parameters and synchronizing them with the motion of the motor, distance values at known angles with respect to the centroid of the robot are maintained.

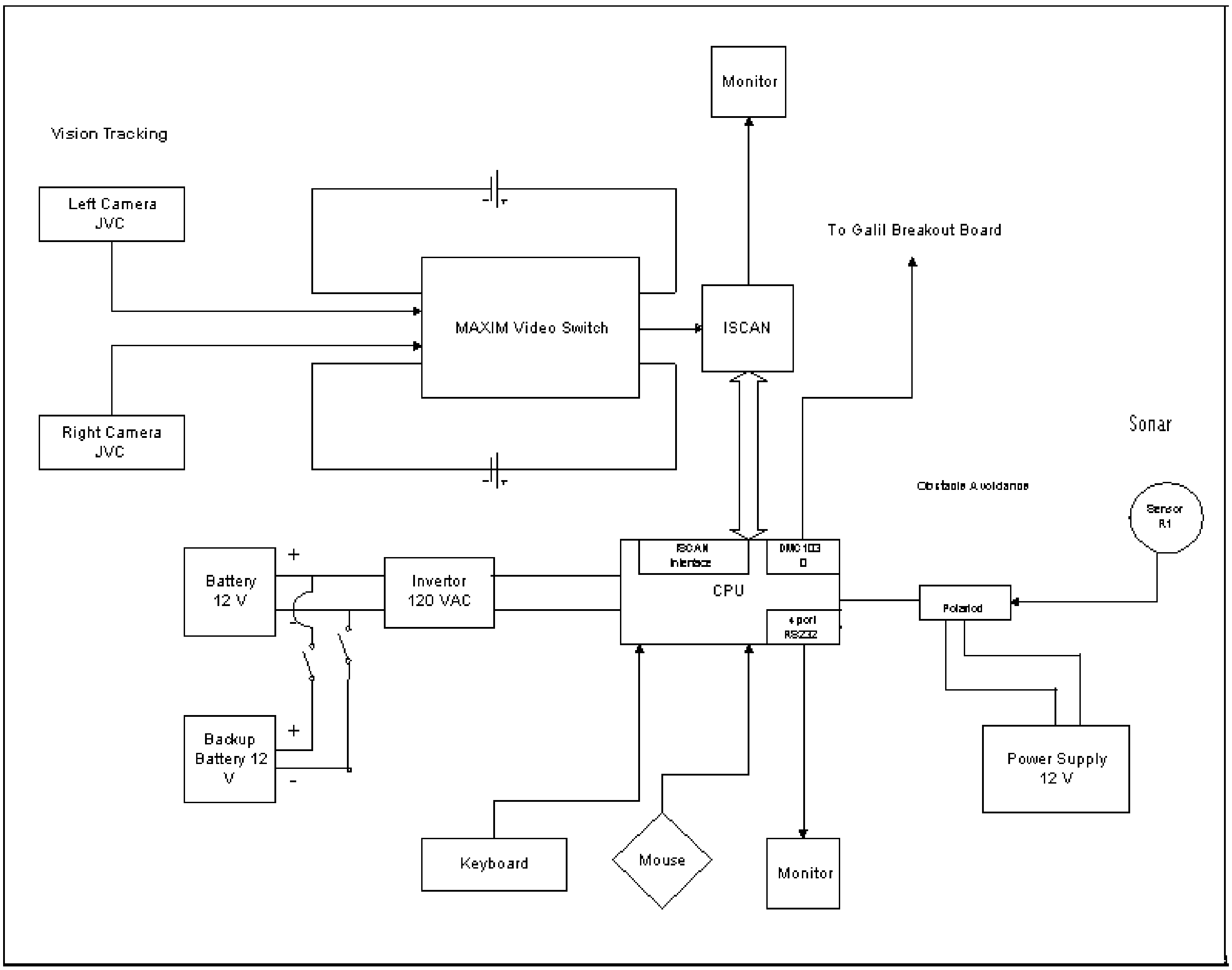


Figure 1: System block diagram

2.1 MOTION CONTROL

The motion control of the AGV designed, has the capability of Zero Turning Radius (ZTR) features, which is gaining popularity and expanding commercially in the U.S. mowing market. ZTR is the ability to make a turn about the center of the axis of the drive wheels. This unique design offers exceptional maneuverability such as sharp turns or turning in place.

Rotating one wheel forward and the other wheel backward accomplishes the ZTR function. However, in our design we can also vary the speeds of the left and right drive wheels while negotiating a curve. This enables the AGV to make a curved turning path parallel to the track lines. One important factor to note is that the wheels do not steer. They negotiate a turn by only changing their individual speeds.

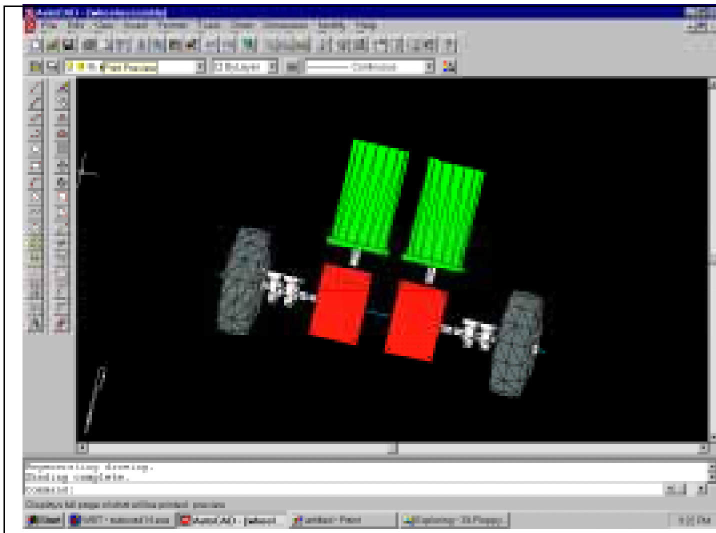
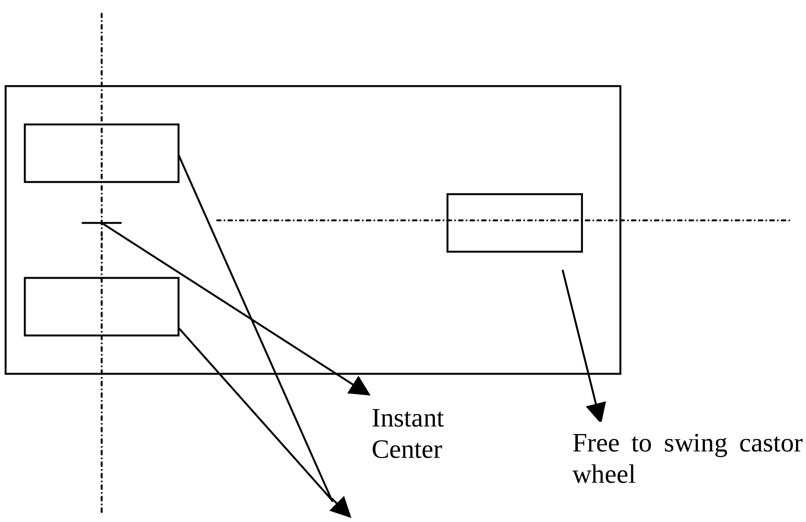


Figure 2: Steering Mechanism of Bearcat II

The AGV is driven and steered by two independent 48 volt, 15 amp motors. These motors drive the left and right drive wheels, respectively, through two independent gearboxes, which increase the motor torque by forty times. The power to each individual motor is delivered from an AMC DC 48A amplifier [4], which amplifies the signal from the Galil DMC motion controller. To complete the control loops; a position encoder is mounted on each of the drive motors. There is a castor wheel in the back of the vehicle, which is free to swing when the vehicle has to negotiate a turn. The figure below gives an idea of the way in which the wheels are arranged. The position of the instant center varies according to the speed and the direction of motion of the two wheels. The figure below shows the position when the wheels are moving in the same direction with different speeds, i.e., $V_R > V_L > 0$.



Controlling the sum and difference of the two wheel speeds does the control of the vehicle.

$$V_L + V_R = 2V_M \tag{1.0}$$

$$V_L - V_R = W\left(\frac{d\theta}{dt}\right) \tag{1.1}$$

Substituting the value of V_R from equation (1.0) in equation (1.1), we have-

$$V_L - (2V_M - V_L) = W\left(\frac{d\theta}{dt}\right) \tag{1.1a}$$

$$2V_L - 2V_M = W\left(\frac{d\theta}{dt}\right) \tag{1.1b}$$

$$V_L - V_M = \frac{W}{2}\left(\frac{d\theta}{dt}\right) \tag{1.1c}$$

$$V_L = V_M + \frac{W}{2}\left(\frac{d\theta}{dt}\right) \tag{1.2}$$

Similarly substituting the value of V_L from equation (1.0) in equation (1.1), we have-

$$(2V_M - V_R) - V_R = W\left(\frac{d\theta}{dt}\right) \tag{1.2a}$$

$$2V_M - 2V_R = W\left(\frac{d\theta}{dt}\right) \tag{1.2b}$$

$$V_M - V_R = \frac{W}{2}\left(\frac{d\theta}{dt}\right) \tag{1.2c}$$

$$V_R = V_M - \frac{W}{2}\left(\frac{d\theta}{dt}\right) \tag{1.3}$$

Assuming a continuous correction control loop, we can confine V_M to a base speed, the optimum value of which was found by experimentation for minimal processing time delay and minimal error. The robot behaved smoothly at $V_M \cong 35000$ to 36000 counts. For Bearcat II, 2000 encoder counts make one revolution. So 36000 counts is equivalent to 18 revolutions. With W/2 being a constant at 13.45 inches, the factor to determine the crux of the logic mainly rests with the $d\theta/dt$ term in equations (1.2) and (1.3).

The ISCAN image-tracking device accomplishes image tracking. This device finds the centroid of the brightest or darkest region in a computer-controlled window, and returns the X, Y coordinates of its centroid and size information of the blob. If no object is found, a loss of track signal is generated. This information is updated every 16 ms. however, the program must wait 10 ms after moving the window to get new data. The vision algorithm feeds the co-ordinates of the centroid of the blob of two snap-shots alternatively as above. In other words, it gives x_1, y_1 and x_2, y_2 (see picture below) [5].

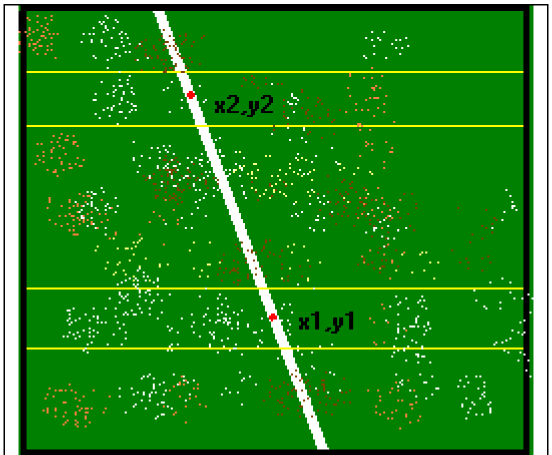


Figure 3: Co-ordinates of the centroid

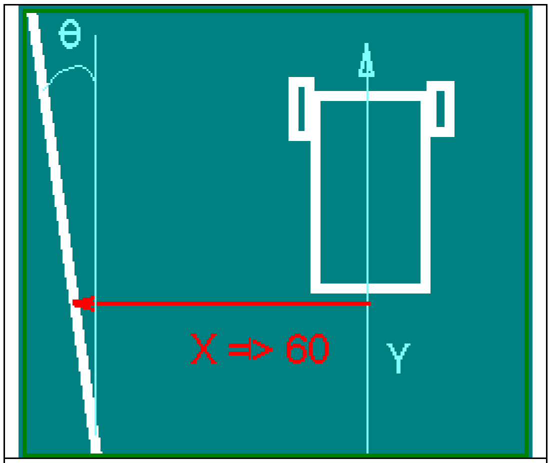
The obstacle avoidance system consists of one rotating ultrasonic transducer. A Polaroid [6] ultrasonic ranging system is used for the purpose of calibrating the ultrasonic transducer. In operation, the system uses a "Time of Flight" (TOF) approach to compute the distance by transmitting sound towards a target and detecting an echo. The elapsed time between the start of the transit pulse and the reception of the echo pulse is measured.

Knowing the speed of sound in air, the system can convert the elapsed time into a distance measurement and hence compute the distance. The range of the system depends on system parameters as well as outdoor operating conditions, but is approximately 40 feet.

Using a closed loop DC servomotor, the transducer is made to sweep an angle depending on the horizon (this is about 640 for a range of 8 feet and about 53.130 for a range of 10 feet horizon). The control loop is closed by an encoder, which measures position for feedback.

The transducer sweep is achieved by programming the Galil motion control system [8]. By adjusting the Polaroid system parameters and synchronizing them with the motion of the motor, distance values at known angles with respect to the centroid of the robot are measured.

Thus, the inputs to the system are: x1, y_1 and x_2, y_2 from the vision algorithm, sonar-position and sonar-distance from the ultrasonic transducer.



The angle 'θ' between the robot and track

The distance of the line from the centroid of the robot is-

$$d_1 = abs(x_2 + x_1)/2 \quad (1.4)$$

The angle of the line with respect to the Y-axis of the robot is given by:

$$\theta = \tan^{-1}\left(\frac{x_2 - x_1}{y_2 - y_1}\right) * \frac{180}{\pi} \quad (1.5)$$

The time instant Δt is given by:

$$\Delta t = t_2 - t_1 \quad (1.6)$$

where t_2 and t_1 are the times at two successive correction loops.

If the number of stops the sonar makes in one sweep is max stop S_m and the sonar position is given by S_p, then the location of the obstacle with respect to the robot is indicated by the factor-

$$\alpha = \left[\frac{S_m/2 - S_p}{abs(S_m/2 - S_p)}\right] \quad (1.7)$$

(Note: The number of stops made by the sonar will always be odd, as the sonar will make mirror stops on the left and right sides and will also take a reading at $\theta = 0$).

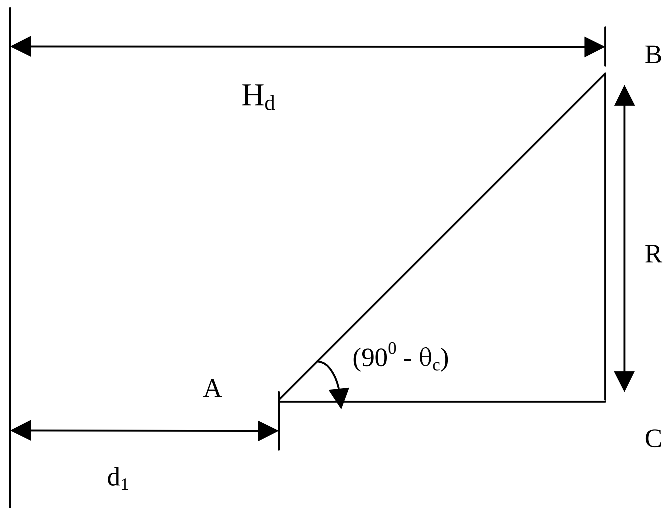
This will help the algorithm infer if the obstacle is in the left or center or the right.

α is taken as 2 if the obstacle is straight ahead.
α is taken as 1 if the obstacle is in the right.
α is taken as -1 if the obstacle is in the left. (1.8)

The sonar always returns some noise distance even if there are some obstacles. Also to see if the obstacle is critical or not, we introduce the following factor in the control sequence:

$$\left.\begin{array}{l}\beta \text{ is taken as 1 if the distance is less than 100 inches.}\\ \beta \text{ is taken as 0 if the distance is less than 100 inches.}\end{array}\right\} \quad (1.9)$$

A is the current position of the robot and for the left camera. If 'R' is the reaction distance tolerated, to go to the point B the robot should turn by θ_c.



Here H_d is the 'Hugging distance', i.e., it is the preferred distance the robot should maintain from the white line and d_1 is the present distance of the robot from the line.

Then we have-

$$\theta_c = \cot^{-1}\left[\frac{R}{H_d - d_1}\right] * \frac{180}{\pi} \quad (1.10)$$

If θ is the angle of the robot with respect to the lines, introducing the correction for this angle, we have-

$$\theta_c = \cot^{-1}\left[\frac{R}{H_d - d_1}\right] * \frac{180}{\pi} + \left[\theta + \left(\frac{abs(H_d - d_1)}{(H_d - d_1)}\right) * 90\right] \quad (1.11)$$

as $\theta + \left(\frac{abs(H_d - d_1)}{(H_d - d_1)}\right) * 90$ is the correction taking into account the distance error.

Including the correction for the obstacle data, from equations (1.7), (1.8) and (1.9), θ_c becomes-

$$\theta_c = \cot^{-1}\left[\frac{R}{H_d - d_1}\right] * \frac{180}{\pi} + \left[\theta + \left(\frac{abs(H_d - d_1}{(H_d - d_1)}\right) * 90\right] + \alpha * \beta * 15 \tag{1.12}$$

This θ_c is the instantaneous correction angle for the left camera.

Hence let us introduce a correction factor for the right camera as follows:

$$\theta_c = (-1)^c \left[\cot^{-1}\left[\frac{R}{H_d - d_1}\right] * \frac{180}{\pi} + \left[\theta + \left(\frac{abs(H_d - d_1)}{(H_d - d_1}\right) * 90\right]\right] + \alpha * \beta * 15 \tag{1.13}$$

where, c = 1, if left camera is used.
and c = 2, if right camera is used.

Hence the final expressions for V_R and V_L are-

$$V_R = V_M - \frac{W}{2}\left[\frac{\theta_c}{(t_2 - t_1)}\right] \tag{1.12}$$

-

$$V_L = V_M + \frac{W}{2}\left[\frac{\theta_c}{(t_2 - t_1)}\right] \tag{1.13}$$

3. CONCLUSION AND RECOMMENDATIONS

A stable platform for the purpose of testing has already been designed and constructed. With the values of the velocities for the left and right wheels, the speed and direction of the robot motion can be easily controlled. This logic when incorporated along with the vision and the rotating sonar systems mounted on the Bearcat II has yielded excellent results. Further improvements in the logic have been identified and work is going on in order to improve the logic better. Since this logic considers only the width of the robot, its base speed and the inclination of the robot with respect to the track, it is stable enough to accommodate for changes in these values.

4. ACKNOWLEDGEMENTS

The work has been a continual learning experience and the authors wish to express their thanks to the previous members of the team. The authors like to express their special thanks to Karthik, who is the backbone behind the team's untiring efforts and Sameer, for his constant encouragement. The authors also wish to express their thanks to Bo Westheider (Electronics) and Dave Breheim (Machinist), and all our sponsors for this effort.

5. REFERENCES

[1] B. Kreczmer, **"Path Planning System for Car-Like Robots"**, Proceedings of the IEEE International Conference on Robotics and Automation, (1998), Katholieke Universiteit Leuven, Leuven, Belgium. V 1, pp 40-45.

[2] A. Yahja, et.al., **"Framed-Quadtree Path Planning for Mobile Robots Operating in Sparse Environments"**, Proceedings of the IEEE International Conference on Robotics and Automation, (1998), Katholieke Universiteit Leuven, Leuven, Belgium. V 1, pp 650-655.

[3] K. C. Kolli, K. Kola, E. L. Hall, **"Design Development of Steering and Speed Control for an Intelligent Mobile Robot"**, Proceedings of SPIE, Intelligent Robotics and Computer Vision, V 3522-36, Boston, Nov. 1998, pp 295-302.

[4] Reliance Electric, Electro-Craft BDC-12 Instruction Manual, Eden Prairie, Minnesota 1993.

[5] T. Samu, **"Vision System for Three Dimensional Line Following of an Unmanned Autonomous Mobile Robot"**, MS Thesis, University of Cincinnati, 1996.

[6] Polaroid Corp., Ultrasonic Ranging System, Cambridge Massachusetts (1992).

[7] Galil Inc, DMC-1000 Technical Reference Guide, Ver. 1.1, Sunnyvale, California (1993).