

***Sonar Based Navigation: Follow The Leader Solution for  
Bearcat III***

A thesis submitted to the Division of  
Graduate Studies and Advanced Research  
of the University of Cincinnati

in partial fulfillment of the  
requirements for the degree of

MASTER OF SCIENCE

In the Department of Mechanical, Industrial and Nuclear Engineering  
of the College of Engineering

2001

by

**Aravind Muralidharan**  
B.E (Mechanical Engineering),

Govt. College of Technology, India, 1995.

**Thesis Advisor and Committee Chair: Dr. Ernest L. Hall**

# *Abstract*

Autonomous robots with mobile capability are finding lot of applications in manufacturing, medicine, space and defense. Design of such a robot is truly a daunting task. The issue is complex because the robot has to interact with its environment when performing the task. One of the possible application for a robot might entail moving the robot through a dynamically decided safe path. Such navigation could be seen as a guiding of a series of mobile robots to a desired destination along a just decided safe path. Numerous research works has been done in the area of path planning and obstacle avoidance algorithms for navigating a robot intelligently through a unknown, unexplored Environment. This research work was done towards fulfilling the requirement of designing a mobile robot to follow a moving leader. The Center for Robotics Research at the University of Cincinnati has built a mobile robot named Bearcat II for the International Ground Robotics Competition being conducted by the Association for Unmanned Ground Systems (AUVS) every year. The objective is to make the Bearcat II follow a lawn mower driven by one of the judges while maintaining a safe distance of about 3 meters. A Polaroid ultrasonic transducer mounted on a micro-motor with an encoder feedback was used to track the co-ordinates and motion of the leader and the steering system is suitably adjusted for re-orienting the robot and to maintain the fixed distance between the robot and the leader. The readings of the sonar at the known adjustable angles are translated to the co-ordinate and relative motion of the leader. The Galil DMC controller suitably drives the left and right motor to steer the robot in the proper direction and at proper speed. This design yields a portable independent system,

which could be suitably integrated or replaced with any different kind of sensor like a laser sensor, which could ascertain the position and motion of the leader.

## *Acknowledgments*

This work has been an interesting learning experience and would not have been possible without the support, help and guidance provided by the various team members who worked and are currently working on Bearcat III as well as Bearcat II. My sincere and deepest gratitude also go to my Thesis Advisor Dr. Ernie Hall for his valuable guidance, support and assistance in helping to conceive, conceptualize and implement this follow the leader algorithm. I also express my thanks for the members of my thesis committee for their suggestions in this research work. It would just be an understatement if I say that their comments helped me a lot in shaping my thesis. My special thanks go to Karthik Kumaraguru who was my constant source of encouragement and information for design of this system. My heartfelt gratitude goes to my friends, parents for being instrumental in helping me to complete this thesis.

# *Table of Contents*

1. Introduction.....	7
1.1 Motivation.....	7
1.2 Objective: .....	8
1.3 Problem Statement: .....	8
Sonar Based Object Detection .....	9
1.5 Organization of this Thesis: .....	10
2. Navigation - Introduction and Literature Review .....	11
2.1 Principles of Navigation .....	11
2.1.1 Overview .....	11
2.1.2 Frame or Reference for Navigation .....	12
2.1.3 Landmark-Based Navigation.....	12
2.2 Robot Navigation .....	13
2.2.1 Guided Vehicles .....	13
2.2.2 Navigation Based on Cartesian Reference Frames.....	13
2.2.3 Landmark Based Navigation.....	15
2.2.3.1 Clustering Techniques.....	15
2.2.3.2 Perceptual Aliasing .....	15
3 System Design and Development .....	17
3.1 The overall system.....	17
3.2 The Polaroid Ultrasonic Ranging System.....	17
3.3 Motor Control.....	20
4 Steering Control and Mechanism.....	21
4.1 Overview.....	21
4.2 Design .....	21
4.3 Motion Control System.....	22
4.4 System Model.....	22
4.4.1 The Encoder .....	25
4.4.2 The Controller .....	26
4.5 WSDK Software Kit.....	27
5.1 Overview.....	30
5.2 Sonar sensor model.....	30
5.3 Why Sonar - A Comparison Study.....	32
5.4 Sonar System Theory and Design .....	33
5.4.1 The Circular Piston Model: .....	35
5.6 Previous Work in Sonar Based Navigation.....	36
5.5 System Description.....	39
5.6 Practical Considerations .....	40
6. Algorithm and Logic for Following the Leader.....	42
6.1 Theory.....	42
6.2 Results.....	49
7. IGRC AUVS Competition.....	52
7.2 Events in the IGRC contest.....	52
7.2.1 Autonomous Challenge Competition.....	53
7.2.1.1 Overview .....	53

7.2.1.2. VEHICLE CONTROL:.....	53
7.2.1.3. OBSTACLE COURSE: .....	54
7.2.2 Follow the Leader Competition.....	56
7.2.2.1 Objective .....	56
7.2.2.2 First Stage.....	57
Phase II - Headway Maintenance and Free Following.....	57
8.2.2.3 Second Stage.....	57
7.2.3 Road Debris Competition.....	58
7.4 Preparation for 2001 contest .....	59
References .....	62
Appendix A.....	67
Relevant Programs .....	67

## List of Figures

Fig 1. System Diagram.....	18
Fig 2. Polaroid Sensor Connection Diagram.....	19
Fig 3. Motion System.....	23
Fig 4. Simulink Model. ....	27
Fig.6 Frequency Responce in a Open Loop .....	29
Fig 8. Sonar Model.....	31
Fig 9. Sonar Ranging System .....	34
Fig 10. Typical Sonar Beam Pattern .....	36
Fig 11. Radiation Pattern of Polaroid Sonar.....	37
Fig 12. Environmental Factors and their effect on quality of the Signal .....	41
Fig 13. Schematic Representation of Object Tracking .....	43
Fig 14. Ranging representation.....	44

## List of Tables

Table 1. Distance Measurement using SONAR .....	50
Table 2. Velocity at wheels for Object at different Positions.....	51
Table 3. Traffic Violations .....	55

# *1. Introduction*

## **1.1 Motivation**

Automated highway systems require technology that allows vehicles to form and maintain "platoons," in which vehicles autonomously follow each other with very short headway's, as short as two meters. The following vehicles must range to the leading vehicles and maintain the headway very accurately under varying speeds, acceleration, braking, and even emergency stops, while steering to track the highway lane markers. The military requirement is for convoys in which only the lead vehicle is manned, while following vehicles not only maintain headway but also steer so as to follow the leader wherever it might go in a trackless course. The Center for Robotics Research at University of Cincinnati has been a regular participant in the competition organized by the Association for Unmanned Ground Systems (AUVS) every year. University of Cincinnati's robot called Bearcat II has participated and won prizes in many of the events organized in this competition. In the 1999 competition a new event called Follow the Leader was introduced in a simulation of the above discussed real world example. The basic aim was to follow a lawn mower or cart driven by one of the judges. The objective was to follow the lead vehicle while maintaining the distance within a standard distance (3 meters). Presently Bearcat II uses an omni-directional motion system (Fish Eye Lens) to determine the angle of the target and the rotating sonar to measure the distance. This research work attempts to accomplish the task using the rotating sonar to detect and locate the object so that robot could be suitably steered to follow the object. The main attractions of sonar ranging system include their low cost, ease of implementation and

inherent safety. Although it comes with a disadvantages in the form of slower processing speeds, and wide beam pattern both of which might contribute to potentially large errors, the realized advantages makes it more suitable for this kind of application.

## **1.2 Objective:**

The primary purpose of this research work was to find a suitable reliable alternative for the existent solution for the follow the leader concept which would be feasible enough to achieve with the present resources and capability and which would be easy enough to implement and adapt to future needs. The use of this dynamic sonar system to detect and avoid obstacles has established the credibility of this system for Bearcat II. This research work attempts to take in to that feature of object detection to help in the process of following the leader.

## **1.3 Problem Statement:**

The Follow the leader consists of two stages namely Phase I – Headway and Phase II - Headway Maintenance and Free Following

### **Phase I - Headway**

The first half of the course will consist of a 15.4 meter length that makes an S-curve with a mean radius of 3 meters. The Challenge will start with the front of the following vehicle 3 meters behind the sign on the rear of the lead vehicle, and both vehicles stationary. The lead vehicle will accelerate slowly to 3 km per hour and hold that speed to the end of the run. When the rear of the lead vehicle reaches 15.4 meters, it will stop, and a measurement will be made of the distance between the two vehicles (from the center of

the sign on the lead vehicle to the front center of the following vehicle). Judges will necessarily walk and stand between the two vehicles to make this measurement. No manual adjustments to the vehicle will be permitted during this stop. The lead vehicle will then start up again and proceed into the second phase of the Challenge.

## **Phase II - Headway Maintenance and Free Following**

At the end of Phase I the course will continue on a track visible only to the lead vehicle, but which will consist of S-curves. Radii of curvature will be from 5 meters down to 3 meters. The lead vehicle will run at a constant 3 km/hr for a distance of 96 meters at the end of which it will stop with its rear end at the finish line, and a second measurement of distance between the two vehicles will be made. On occasion the lead vehicle on the unmarked course will pass within 1.5 meters of a construction barrel.

## **Sonar Based Object Detection**

The specific challenge of designing an intelligent controller is in deciding on requirements as to what information is needed to satisfy the requirements, how to measure it and how to use this information in a satisfactory manner that would meet the performance specification of the machine. The overall design objective was to design a suitable algorithm and solution methodology for navigating the robot through trackless terrain, with sharp turns and terrain elevation and at the same time enable the robot to follow the leader within a reasonably allowed distance. This implies design of a suitable subsystem, which could be easily integrated, with upper level control logic of the robot to enable it function as an integral system meeting all the performance requirements.

Sonar ranging with its inherent cost and easy to implement advantages is based on measuring time for a burst of continuous wave ultrasound to be returned to the sensors. The reflected echo strength depends on the size, shape, texture and orientation of the reflecting surface. Large surfaces reflect more and increase the chances of getting detected. Depending on its shape, a reflecting surface may cause dispersion or a focused reflected beam.

## **1.5 Organization of this Thesis:**

Chapter 1 is an introduction to this research work and explains the background, motivation and the technology behind this thesis. Chapter 2 explains the navigation theory and provides a literature survey on the previous work done on navigation. Here different subcomponents of the system are also explained and discussed. Chapter 4 explains the steering control design and mechanism. This chapter progresses from a basic introduction to the steering control to the actual design and implementation of this system in Bearcat II. Chapter 5 explains the sonar based object detection approach and discusses partially the advantages and disadvantages of this system when compared to the present fish eye lens based omni-directional motion system. Chapter 6 discusses in detail the algorithm behind the ultrasonic transducer based object tracking approach. It explains the science and methodology behind this approach. Chapter 7 gives an overview of the IGVC competition and Chapter 8 gives future conclusions and recommendations for future research. Finally, the programs behind the implementation of the logic are present as ANSI C code in the appendix.

## ***2. Navigation - Introduction and Literature Review***

### **2.1 Principles of Navigation**

#### **2.1.1 Overview**

The ability to navigate is the most essential and important capability of a mobile robot. The ability to stay operational by avoiding obstacles and dangerous situations such as collisions and staying within safe operating condition also comes as very essential and important, but navigation takes a higher priority because it is a must to perform tasks related to specific places in the robot's environment.

Navigation is a composite of three fundamental competencies

1. Self-Localization.
2. Path Planning.
3. Map building

Localization denotes the ability of the robot to establish its own position within a frame of reference. Path planning is just an extension of localization in the sense that it requires determination of the robot's current position and the position of the desired target, both calculated within the same framework of reference. Map building just means any notation describing the locations within the framework of reference.

### **2.1.2 Frame or Reference for Navigation**

The frame of reference is the most crucial component of the robot navigational system. The frame of reference is actually a fixed axis reference, which anchors the navigational components. The Cartesian co-ordinate system is a simple and easy to implement reference system. However, the problem with a mobile robot is that the frame itself moves with respect to its environment and it becomes impossible to measure precisely with dead reckoning the robot's movement because of problems like wheel slippage, elevation changes, etc. The robot can only measure through proprioception, i.e. via internal measurements and is therefore unable to detect the changes to the entire frame of reference.

Almost all of the work done on mobile robots, use internal geometric representation of the robot's environment to perform navigational tasks, for example the MOBOT III robot used by Knieriemann and Puttkamer<sup>1</sup> and Kampmann and Schmidt<sup>2</sup> use maps supplied by the designer to navigate the robot.

### **2.1.3 Landmark-Based Navigation**

Navigation with respect to the external landmarks is called landmark based navigation or piloting. Here the required course is not determined through path integration as in dead reckoning but through identifying landmarks or sequences of landmarks and either follow these landmarks in a particular sequence or order or recall the required compass of direction from a recognized landmark. If the robot is able to identify the landmarks unambiguously, navigation can be achieved with respect to the world, rather than with respect to internal frame of reference. One way to do this would be to use information

regarding the relationships between the landmarks. Such topological mapping occurs in animals and humans in mapping of sensors onto the cortex (Knudsen<sup>3</sup>).

## **2.2 Robot Navigation**

### **2.2.1 Guided Vehicles**

Guiding the robot is the easiest way of navigating it. This is done by, placing an inductive loop or magnet in the floor, painting lines on the floor or by placing beacons, markers or bar codes etc in the environment. They are primarily used to do transportation's tasks in industrial environment. Typically Automated Guided Vehicles (AGVs) are built for one specific purpose and are controlled by a specifically designed program or a human controller. Modifying them for alternative routes is difficult and may necessitate modification to the environment. AGVs are expensive to install or to modify after installation, because they require specially prepared environment.

### **2.2.2 Navigation Based on Cartesian Reference Frames**

Because of problems like wheel slipping it is difficult to design robot navigation based on dead reckoning. Many mobile robots use sensory information in addition. An example for robot navigation system based on a Cartesian frame of reference and dead reckoning is the use of certainty grids (Elfes<sup>4</sup>). Here the robot's environment is divided into cells of finite size, which are labeled initially as unknown. As the robot moves and explores its environment, estimating its current position by dead reckoning, more and more cells become "occupied" or "free" cells, dependent on range sensor data, until the environment is mapped in this fashion.

A similar system based on a Cartesian co-ordinate system is the Elden system (Yamauchi & Langley<sup>5</sup>), which uses “adaptive place networks to construct spatial representations. In this, units correspond to a region in Cartesian space, while links represent the topological relationships between these units. As this system is dependent on an accurate odometry, the robot has to recalibrate the odometry every 10 minutes.

The *OxNav* navigation system (Stevens et al.<sup>6</sup>) also used Cartesian map, which contains discernible sonar features, and is supplied before operation. The robot uses sonar sensors to track features, and then fuses the odometry data, feature data and the prior map information to ascertain its current position.

Maeyama, Ohya and Yuta<sup>7</sup> present a similar Cartesian map based robot system for outdoor use.

A similar system using a predefined network of fixed-location detectors is made by MTI Research Inc., Chelmsford, MA<sup>8</sup>. MTI’s computerized Opto-electronic Navigation and Control (CONAC<sup>™</sup>) is a navigational referencing system employing a vehicle mounted laser unit called Structured Opto-electronic Acquisition Beacon (STROAB). The scanning laser beam is spread vertically allowing the receivers, called Networked Opto-electronic Acquisition Datum’s (NOADs) to be mounted at arbitrary heights. Detection of incident illumination by a NOAD triggers a response over the network to a host computer, which calculates the implied angles  $\alpha_1$  and  $\alpha_2$ . An index sensor built into the STROAB generates a rotation reference pulse to facilitate heading measurement.

## 2.2.3 Landmark Based Navigation

In this method, navigation is done based on location-dependent sensory perception. Since sensory perception is subject to noise and variation, some kind of generalization, which retains the salient features of a landmark, but discards the noisy components is needed.

### 2.2.3.1 Clustering Techniques

In this method a clustering mechanism is used to cluster the data in a topological fashion. This means that even if two perceptions are not identical, but merely similar, they will nevertheless be clustered together. Kohonen's<sup>9</sup> self-organizing feature map is one such example. Another example for self-organizing clustering mechanism is Restricted Coulomb Energy networks (RCE) whose application can be found in the work of Kurz<sup>10</sup>.

### 2.2.3.2 Perceptual Aliasing

If an assumption that the robot will only visit fixed number of locations could be made, a unique perception for each location could be obtained. Franz, Schölkopf, Mallot and Bühlhoff<sup>11</sup> for example use a graph to represent their robot's environment. In their case no metric information at all is used, instead, nodes representing 360-degree camera snapshots are used. The edges between the snapshots represent the relationship between them.

Another method is to incorporate history of local sensor sequences into the recognition of locations. Tani and Fukumura<sup>12</sup> add previous sensor information to the input field of an artificial neural network controller in their goal directed mobile robot.

Yet another method is to use the combination of perception and dead reckoning to disambiguate similar looking locations. For example, Kupiers and Byun<sup>13</sup> move the robot itself towards locally distinctive places as a means of correcting the drift error due to odometry. This method is effective if the robot knows where it is at the start.

Atomic Energy of Canada Ltd. (AECL) and Ontario Hydro Technologies jointly developed a system that uses Landmark based navigation called, “Autonomous Robot for a Known Environment”(ARK) with the support of University of Toronto and York University<sup>14</sup>. The ARK module uses an interesting hybrid approach: the system stores and remembers landmarks by generating a three-dimensional gray level from a single training image taken from a CCD camera. The robot when operational searches for those landmark that are expected from a known momentary position.

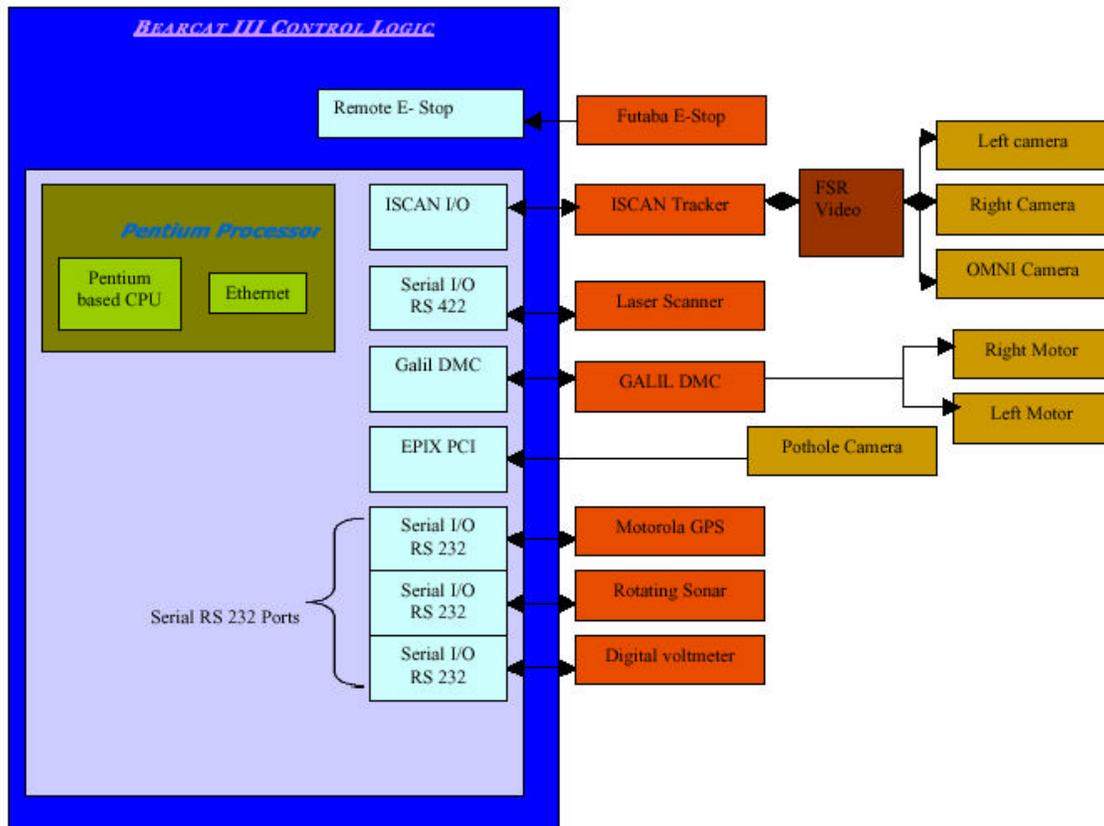
## ***3 System Design and Development***

### **3.1 The overall system**

The mobile robot is a sophisticated, intelligent, controllable, programmable system. The adaptability of the robot basically depends on the conceptual, analytical and architectural design of the sensing and controlling system used. The mobile robot provides an excellent test bed for investigations into generic vision guided robot control since it is similar to an automobile and is a multi-input, multi-output system<sup>15-18</sup>. The main components of the Bearcat II robot are (1) vision guidance system (2) steering control system (3) object tracking system (4) speed control (5) safety system (6) power and supervisory unit. The block diagram of the system is shown in the Fig 1.

### **3.2 The Polaroid Ultrasonic Ranging System**

The object tracking system consists of multiple ultrasonic transducers. The ultrasonic transducers were calibrated using a Polaroid ultrasonic ranging system. The processing for data calculations was done using an Intel 80C196 microprocessor and a circuit board with a liquid crystal display. The distance values are captured and sent through a RS232 port to the control computer. The systems require a separate power supply of 10-30 VDC, 0.5 amps. The two major components, which make the ultrasonic ranging system



**Fig 1. System Diagram**

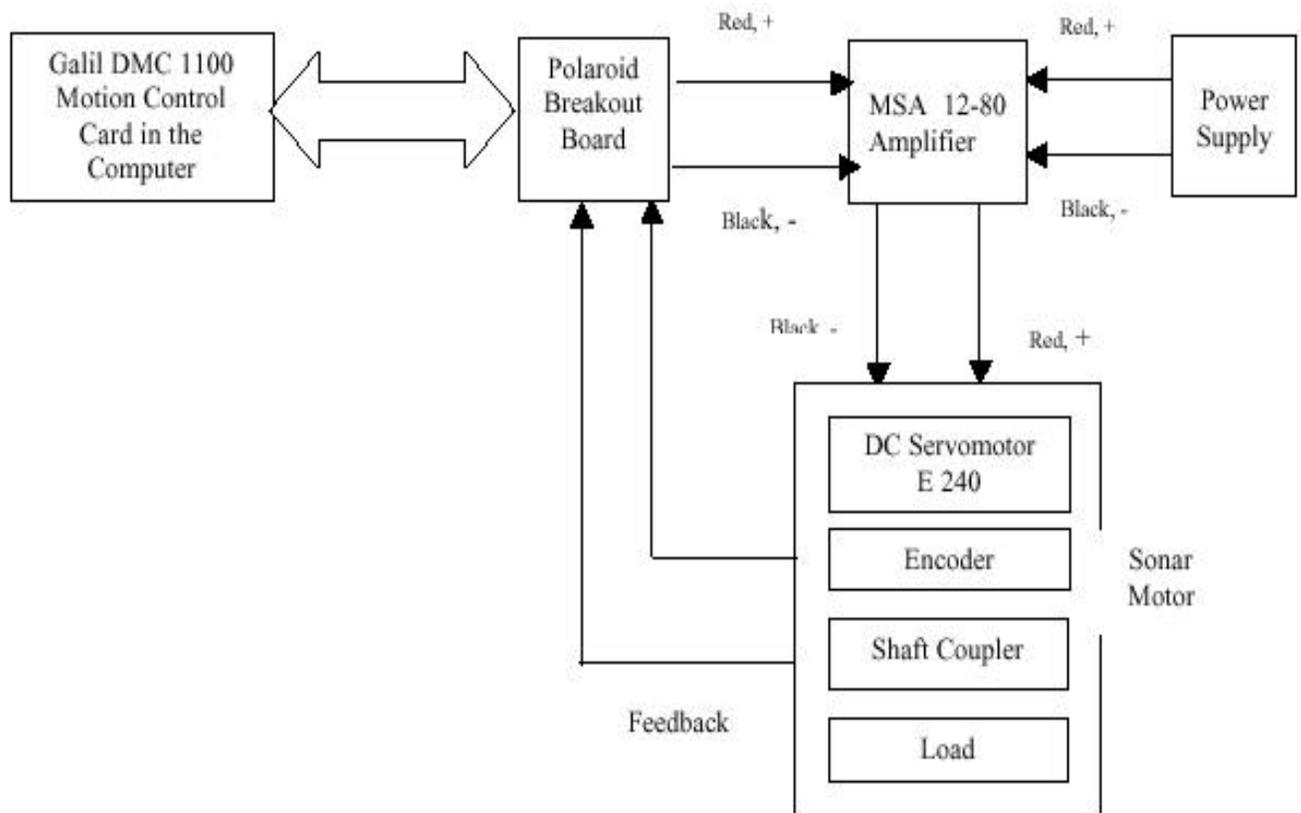
are the transducer and the drive electronics. A pulse of electronically generated sound is transmitted towards the target and the resulting echo is detected. The time elapsed between the start of the transit pulse and when the echo pulse is received is computed. Since the speed of the sound in the air is known, the system can convert the elapsed time into a distance measurement<sup>19</sup>. The speed of the sound in air is given by

$$C = \sqrt{\frac{B}{r}}$$

Where

B – Bulk Modulus of the air and  $\rho$  is density of the air. The speed of the sound around room temperature is 336 m/sec.

The two major components of the drive electronics are digital and the analog system. The ultrasonic frequency is generated using the digital electronics. A drive frequency of 16 pulses at 52 kHz is used for this application. The Intel microprocessor generates all the digital functions for this system. The analog functionality is provided by the Polaroid integrated circuit. All the operating parameters like transmitting frequency, pulse width, blanking time and the amplifier gain are managed and controlled by developer software provided by Polaroid.



**Fig 2. Polaroid Sensor Connection Diagram**

### **3.3 Motor Control**

The transducer is made to sweep angles depending on the horizon using a closed loop DC motor arrangement (this is approximately 64 degree for a range of 8 in and about 53.130 for a range of 10 feet and 10 feet 10 inches horizon). The loop is closed with a encoder feedback from a Reliance Electric brushless motor with encoder.

The drive hardware comprises two main interconnected modules, the Galil ICB930 and the four axes ICM1100. The communication between the main control board DMC 1030 and ICM 1100 is established through a 50 pin parallel interface. The Polaroid sensor control is connected through an RS232 interface. The transducer sweep is achieved by programming the Galil<sup>20</sup>. The sonar measurements is achieved through the Polaroid system. By tuning the sonar parameters and synchronizing them with the motion of the motor, the distance values at known angles are measured with respect to the centroid of the robot.

## ***4 Steering Control and Mechanism***

### **4.1 Overview**

The motion control of the Bearcat II robot has the ability of turning about its drive axis, which is called Zero Turning Radius (ZTR). ZTR is gaining popularity and finding expanding use in commercial US mowing market. This feature offers extensive maneuverability and can negotiate very sharp turns with much ease. The ZTR functional capability is achieved by turning wheels in opposite direction. When one is rotated forward and the other one is rotated backward the robot could turn around its own drive axis. By actually rotating the wheels at different speeds and in opposite directions BEARCAT II is able to negotiate curves smoothly.

### **4.2 Design**

The robot base is constructed from a 80/20 Aluminum Industrial Erector Set. Bearcat II is steered using two independent 36 Volt, 12 Amp motors. Each of these motors drives one of the wheels independently using a gearbox. The gearbox in the transmission system helps to amplify the motor torque by about 20 times. The power for the individual motor is supplied by a BDC 12 amplifier, which actually amplifies the signal from the Galil DMC motion controller. A position encoder mounted on each of the drive motors completes the control loop. A castor wheel supports the rear of the robot, which is free to swing when the robot has to negotiate a curve. The velocity feedback signal is derived by numerically differentiating the position encoder signal. The control of the motion is done by the usage of differential speed drive wheel. The drive wheel speeds could be varied according to the change in the direction of the path or track followed by the robot.

## **4.3 Motion Control System**

The crucial part of any control system is the selection of right parameters for the PID controller. The motion control system of the BEARCAT III robot should help it to maneuver curves and follow the leading vehicle or the leader. Designing the right PID controller for the Bearcat III robot is the most crucial step for its success.

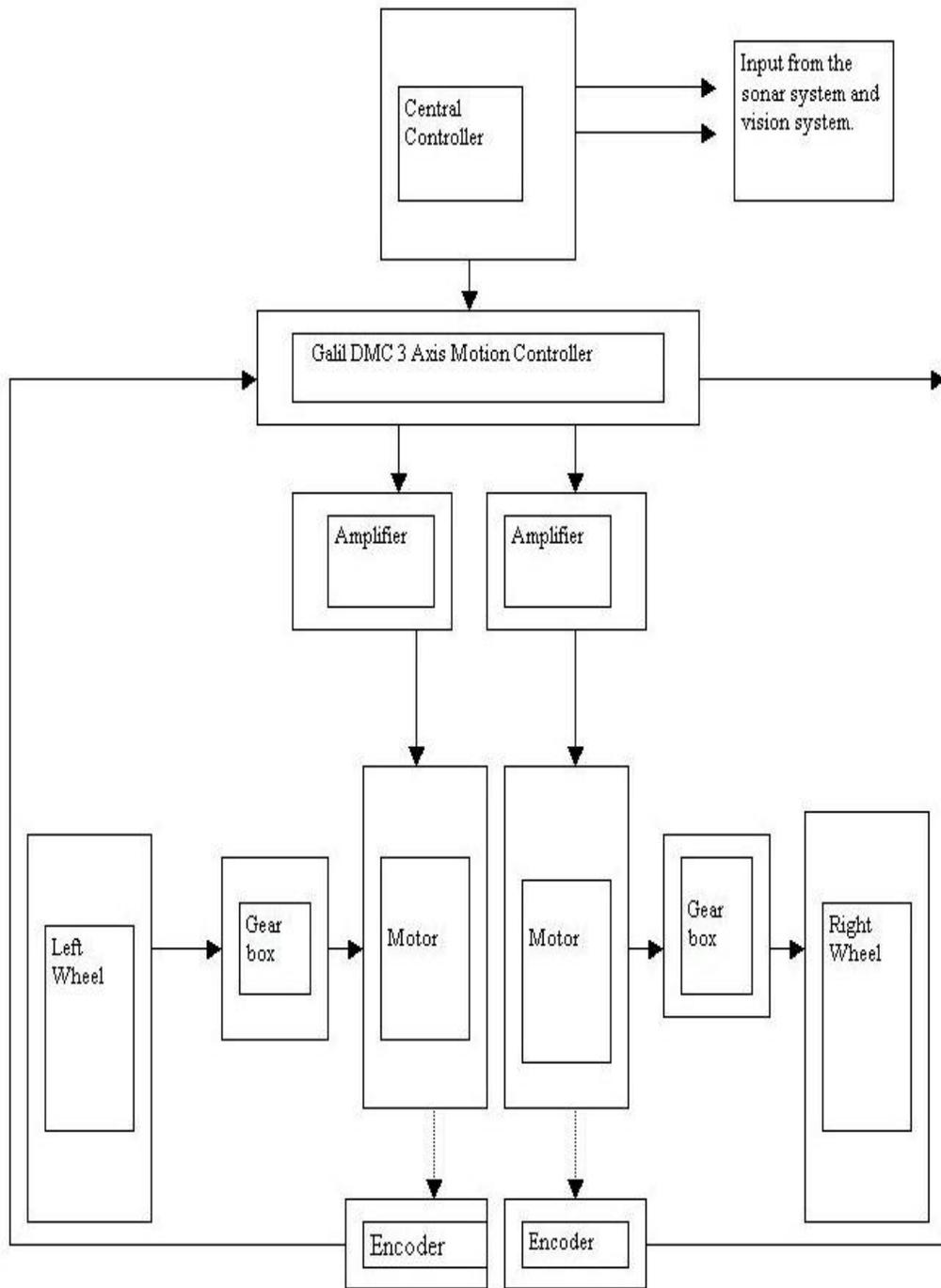
Two Electro-craft brush-type DC servomotors drive the two wheels independently. The encoders provide the feedback from the system. The two drive motors are operated in parallel using the Galil MSA 12-80 amplifiers. The Galil DMC 1030 motion control board is the main controller card of the system and it is controlled through a computer.

## **4.4 System Model**

The position-controlled system comprises a PID controller, a position servomotor with and Encoder and an amplifier. The servomotor is an Electro-craft brush type DC motor and the PID controller is a Galil DMC 1030 motion control board.

The amplifier model is configured in three modes, either in voltage loop, or in current loop or in a velocity loop. The transfer function, which relates the input voltage to the motor position, depends upon the configuration of the system, which is a four axis sytem.

board.



**Fig 3. Motion System**

a. Velocity Loop.

In this velocity loop, a tachometer feedback to the amplifier is incorporated in to the system. The transfer function now actually is a ratio of the Laplace Transform of the angular velocity to the input voltage V.

$$\text{i.e. } \frac{w}{V} = \frac{\frac{K_a K_t}{J s}}{1 + \frac{K_a K_t K_g}{J s}} = \frac{1}{K_g (s t_1 + 1)}$$

$$\text{Where } t_1 = \frac{J}{K_a K_t K_g}$$

$$\frac{P}{V} = \frac{1}{[K_g s(s t_1 + 1)]}$$

The motor parameters and the units are:

$K_t$  - Torque constant

R- Armature Resistance

J - Combined Inertia of the motor and the load

L – Armature Inductance

b. Voltage Loop

In voltage loop, the voltage source to the motor is the amplifier. The gain of the amplifier will be  $K_v$ . And the transfer function of the motor with respect to the voltage will be

$$\frac{P}{V} = \frac{K_v}{[K_t s(st_m + 1)(st_e + 1)]}$$

where

$$t_m = \frac{RJ}{K_t^2}(s)$$

and

$$t_e = \frac{L}{R}(s)$$

c. Current Loop:

The amplifier acts as the current source for the motor in this mode. The transfer function is:

$$\frac{P}{V} = \frac{K_a K_t}{J_s^2}$$

where

$K_a$  – Amplifier gain.

#### 4.4.1 The Encoder

The encoder is the integral part of the servomotor and has signals A and B, which are 90 degrees out of phase. The quadrature relationship between the signals makes the resolution of the encoder increased to 4N quadrature counts/rev. Here N actually denotes the number pulses generated per revolution.

The model of the encoder can be represented as

$$K_f = \frac{4N}{2\pi} \text{ counts/rad.}$$

#### 4.4.2 The Controller

The Galil DMC 1030 board has three components, the Digital to Analog Converter, the Digital Filter and Zero Order Hold (ZOH). The parameters of the controller are denoted by K, A and C. These parameters are selected based on the value of Proportional, Integral and Derivative gains ( $K_p$ ,  $K_i$ ,  $K_d$ ) of the PID controller.

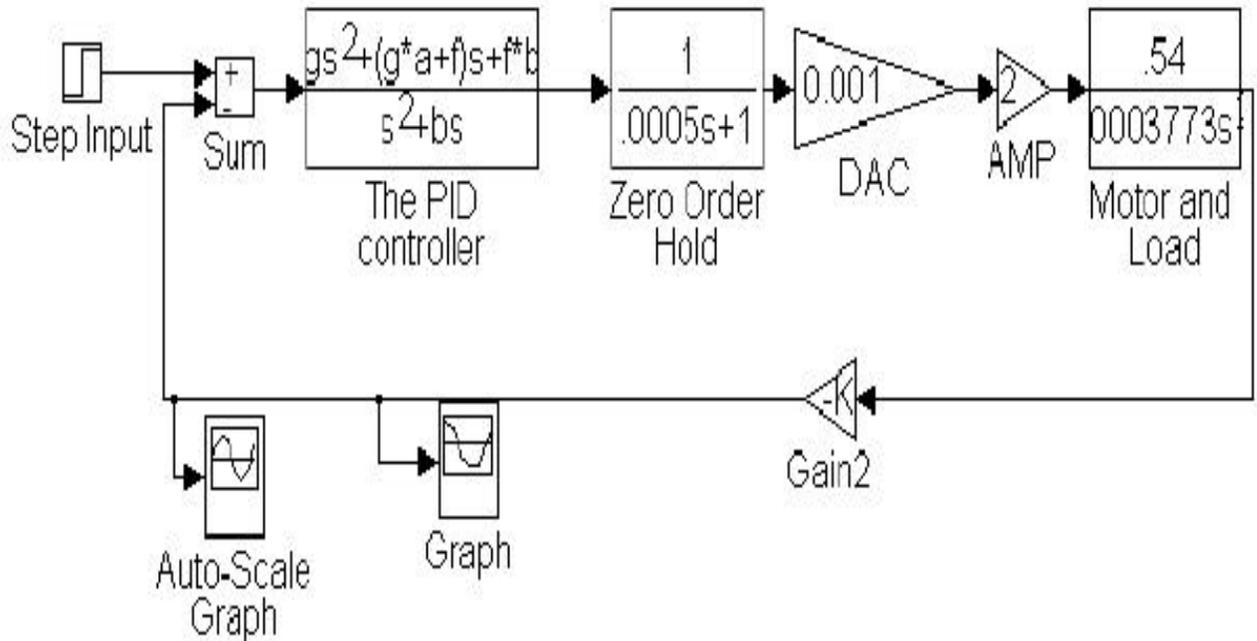
The design objective of the steering mechanism is to provide a stable control coupled with good phase and gain margin and a quicker response. A proper estimate and use of the  $K_p$ ,  $K_i$ ,  $K_d$  values are essential for a controlled behavior of the steering system. This is achieved by estimating these values under different conditions like

- (a) Steering wheel on the ground, Robot remains Stationary
- (b) Steering wheel on the ground, Robot is Moving and
- (c) Steering wheel is off the ground.

In addition to this the amplifier should be fine tuned to ensure that signals are amplified sufficiently to drive the wheels. Tuning the amplifier parameters, particularly loop gain and the selection of the PID parameters are very important and require iterative adjustments<sup>21</sup>.

A Simulink model for deriving the values analog gains for the PID controller was set with a Matlab file and input is considered as a step-input fed to the summation block.

The derived values for the PID controller are tested and fine-tuned using the software kit by Galil Motion Inc., WSDK 1000.

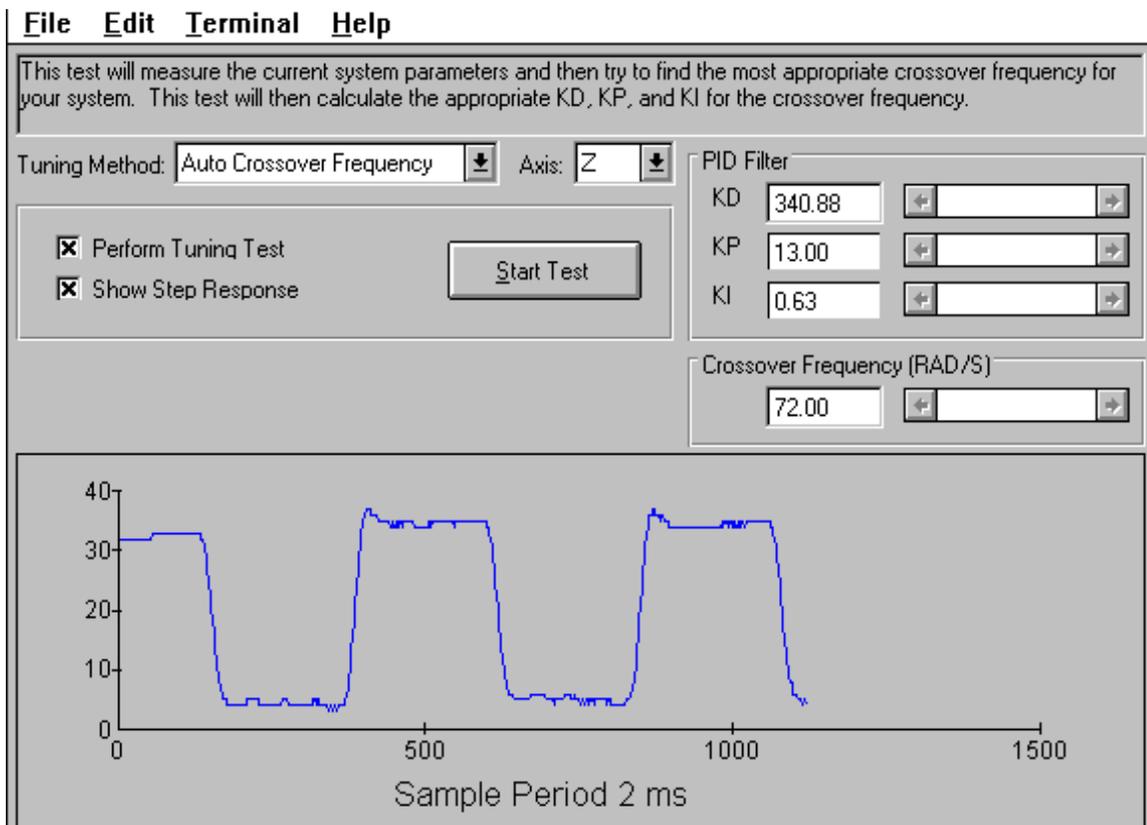


**Fig 4. Simulink Model.**

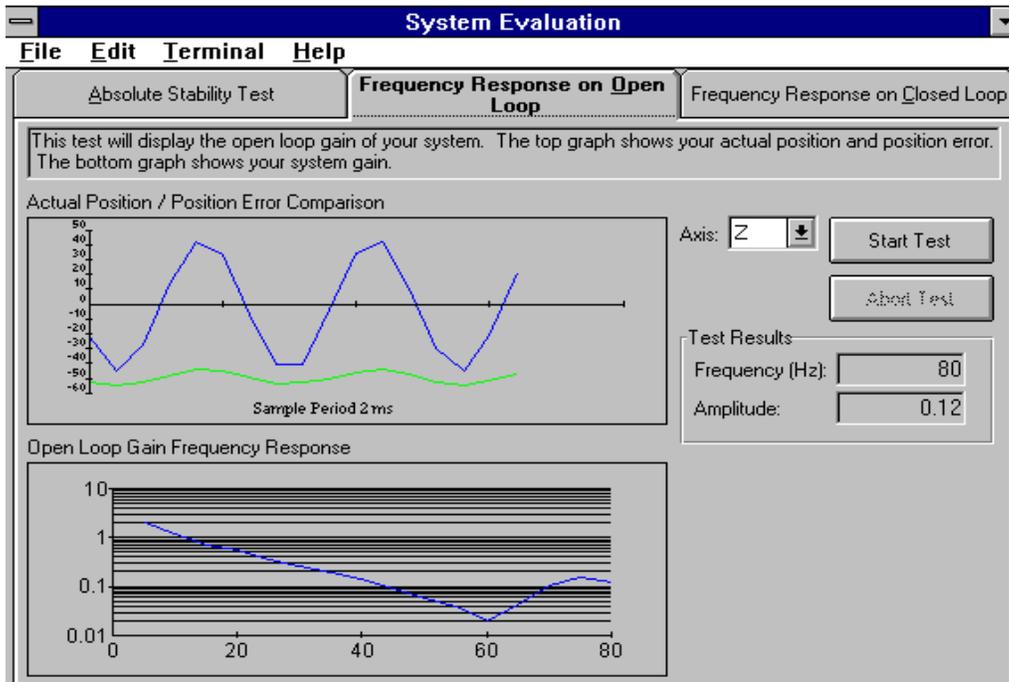
## 4.5 WSDK Software Kit

The Window Servo Design Kit (WSDK) is windows based software, which helps to set-up, configure, tune, test, and analyze Galil Motion Controllers. In order to communicate with the Galil Controller, first it has to be registered with the windows registry. Once registered, the servo design kit program can communicate with the controller. The parameters associated with the PID transfer function namely, KD, KP and KI, for the optimal performance of the Galil Motion Controller, can be determined manually or automatically using WSDK. The fig. 5 below shows the automatic cross frequency

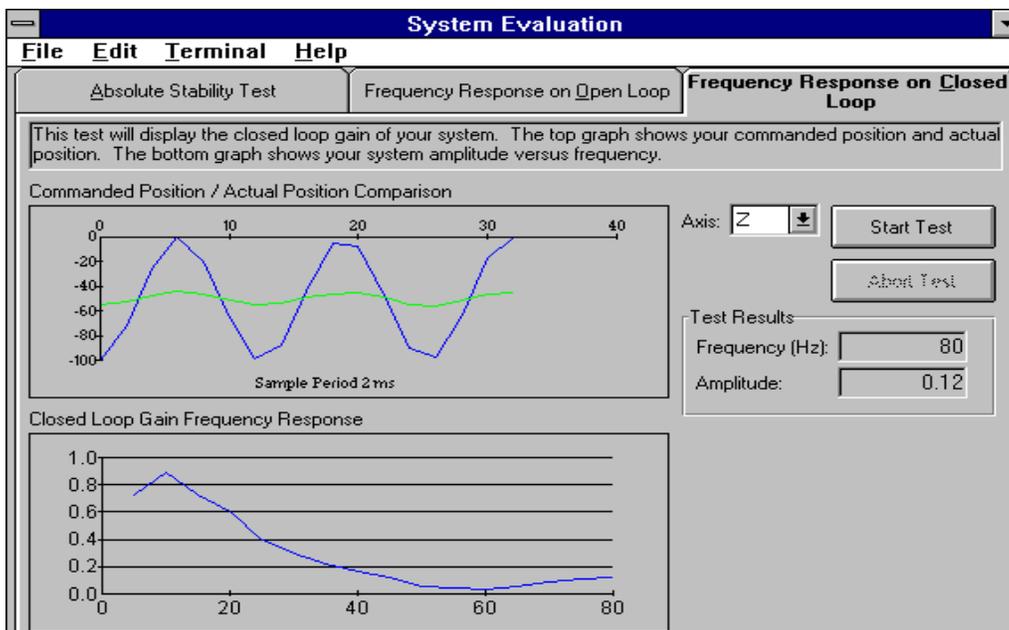
method of measuring the KD, KP and KI values. The cross frequency which is measured in radians/second, is defined as the maximum frequency to which the system can respond without loss of gain. Once the tuning is complete the response characteristics of the system should be checked as a system evaluation. There are three kinds of system evaluation tests that could be done using the WSDK software kit. The fig. 6 & 7 shows the frequency response test on the open loop and the closed loop. These tests check the open-loop/closed-loop frequency response of the system by plotting the actual position against the positional error.



**Fig 5. Autocross-over frequency calibration**



**Fig.6 Frequency Response in a Open Loop**



**Fig 7. Frequency Response in a closed loop system**

# ***5. Sonar Based Navigation in Mobile Robots***

## **5.1 Overview**

The motivation to use sonar for mobile robot navigation comes from the ultrasonic sensing capabilities of bats, which use echolocation to determine their position and prey. The time-of-flight (TOF) based navigation and ranging has received great deal of effort, but the successes are not very promising when it comes to ascertain absolute positions. More sophisticated techniques like amplitude based interpretation, frequency modulation are finding place in applications where demand for accurate estimation is more. Nevertheless in an application like following a leading vehicle with the distance kept within an acceptable range, typical time-of-flight sonar can in fact fulfill the perception role of autonomous navigation.

## **5.2 Sonar sensor model**

Sonar actually stands for Sound Navigation and Ranging and was initially developed for underwater applications<sup>22</sup>. However the term SONAR has been used here to describe airborne ultrasonic range sensing. The transducer is of monotonic type where both the transmitter and the receiver are same. Range measurements are actually obtained through Time of Flight calculation. We conveniently neglect echo amplitude and the frequency modulation information for this specific purpose. The sonar system that is being used here is the Polaroid ultrasonic ranging system.

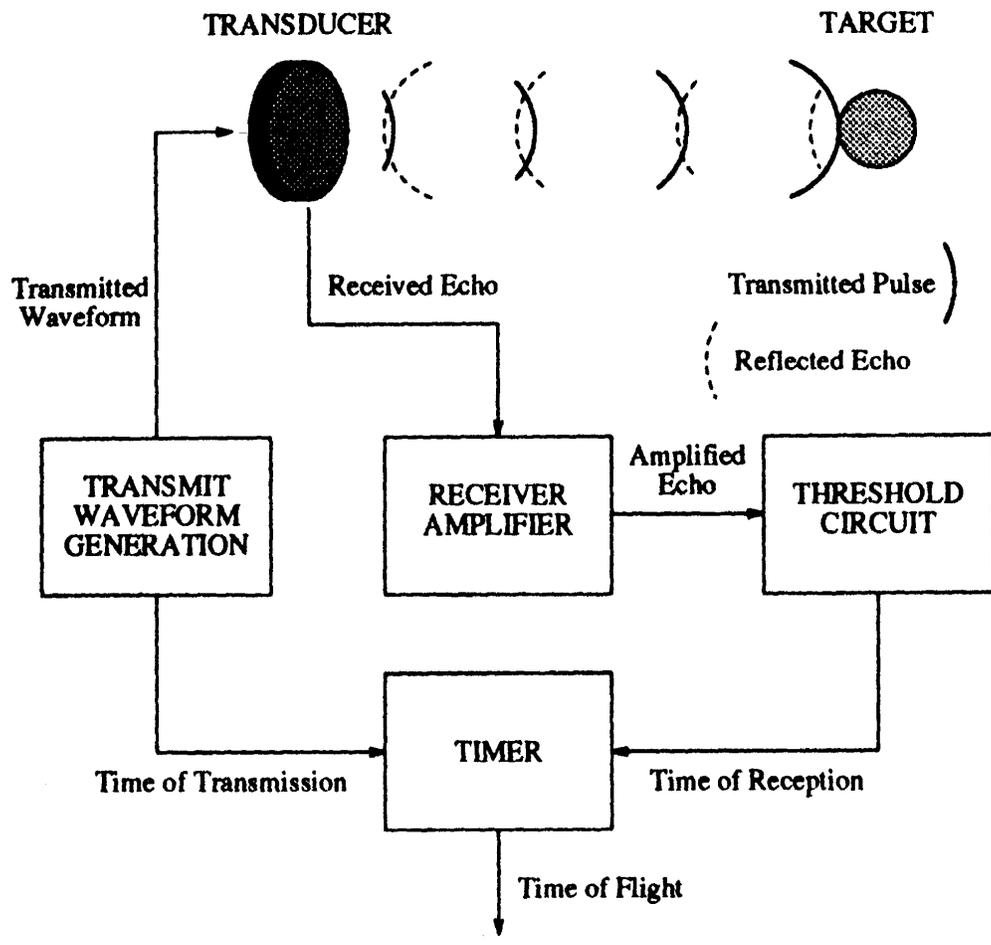


Fig 8. Sonar Model

## 5.3 Why Sonar - A Comparison Study

The reasons for sonar being selected for mobile robot navigation's are multi-fold.

Although the optical range finders like the laser based systems used by Hinkel et al<sup>23</sup>, to inexpensive amplitude-based infrared devices, as used by Flynn<sup>24</sup> and Connell<sup>25</sup> could be seen as potential technical solutions, the latter gets eliminated because of its inability to provide accurate range information because of its surface sensitive reflectance properties. Laser based systems with dense scan in less than 1 second although looks outclassed, sonar's potential is in the fact that typical outdoor environments in a contest like this have only surprisingly few acoustic targets. Hence the task of achieving and maintaining correspondence between observations and map targets, if any, will be considerably easier in many environments with acoustic sensing.

Also sonar is a better position estimator than its reputation goes. The experimental evidence provided by Leonard and Durrant-Whyte<sup>26</sup> substantiates that.

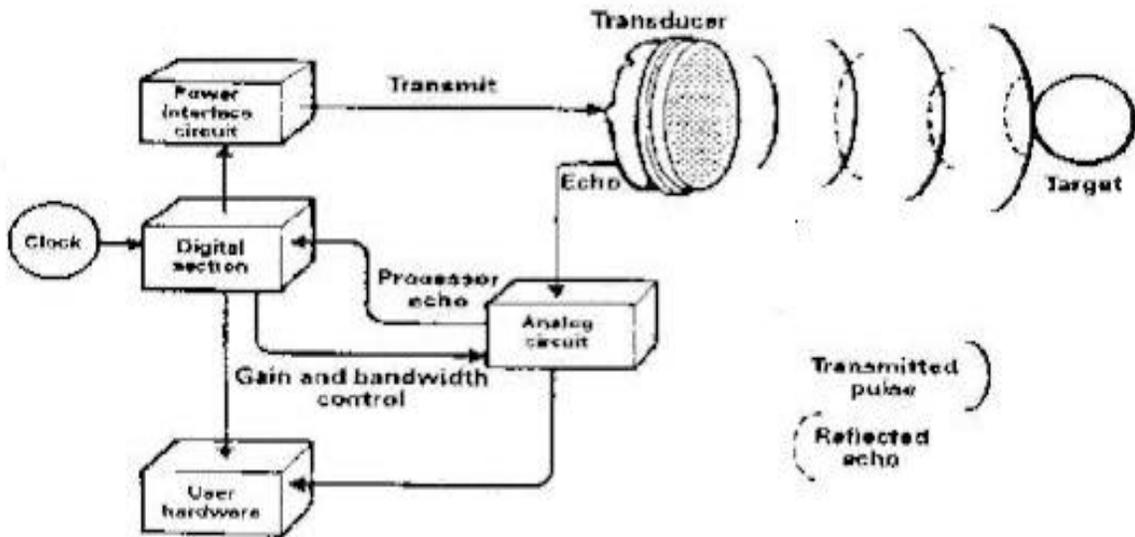
Sonar is better than vision in the sense the wealth of information available with a vision system is enormous and the problem actually becomes extracting useful, required information from the huge data available. Since the present omni-directional vision based approach actually depends on the presence of the Dark and Light spots on the trailing face of the leader, this implementation restricts itself to an application where such detection is feasible. However with sonar even though it might not depend on the surface brightness of the leader, the presence of any obstacle or error response could be a misleading signal for the system. There have been practical difficulties in continuously detecting the leader in the omni-directional vision method, even though it gives reliable

results when experimented in lab, the disturbances that actually happen when moving the robot in a trackless field path because of presence of other possible reflecting interfering surfaces actually make the operation less reliable. Even though sonar might end up giving wrong signals because of unintended reflections, the possibility of integrating the design with other systems in eliminating such noise makes it a better choice.

## **5.4 Sonar System Theory and Design**

The two major components of the ultrasonic ranging system are the transducer and the drive electronics. In operation, a pulse of ultrasonic sound is transmitted towards a target and the resulting echo is detected. The elapsed time between the start of the transmit pulse and the reception of the echo pulse is measured. Since the speed of the sound in air is known, the system can convert the elapsed time into a distance measurement.

The transducer acts as a loudspeaker as well as a microphone. The diameter of the speaker determines the acoustical lobe pattern and acceptance angle during transmission and reception of the sound waves. In case of electrostatic transducers the foil forms the moving element which transforms the electrical energy in to sound waves and returning echo back in to the electrical energy. See Fig. 9 for details.



**Fig 9. Sonar Ranging System**

However the transmission property of the air is much favorable than water as the absorption rate of sound in air at 80 kHz is 2.9 dB/m compared with 0.02 dB/m for water. The attenuation of the sound increases with the frequency and is actually a function of the temperature and humidity. Also the speed of sound in air is temperature-dependent, and characterized by the relationship

$$c = 331.4 \sqrt{\frac{T}{273}} \text{ m/sec}$$

Where T is the ambient temperature in Kelvin.

#### 5.4.1 The Circular Piston Model:

Electrostatic transducers of the Polaroid type are generally modeled as a plane circular piston in an infinite baffle<sup>27</sup>, yielding a radiation characteristic function of

$$P(\mathbf{q}) = \frac{2J(ka \sin(\mathbf{q}))}{(ka \sin(\mathbf{q}))}$$

Where  $k = 2\pi/\lambda$  is the wave number,

a is the transducer radius and

$\theta$  is the azimuth angle measured with respect to the transducer axis.

For a Polaroid transducer the values that are considered are  $a=19$  mm and  $\lambda= 6.95$  mm.

However to really answer the question over what range of angles is a target visible, we need to know the characteristics of the transmitted pulse, target surface, receiver amplifier and thresholding circuit. Actually radiation patterns of Polaroid transducers are not symmetric and actually vary from transducer to transducer. This effect is more pronounced for the side lobes.

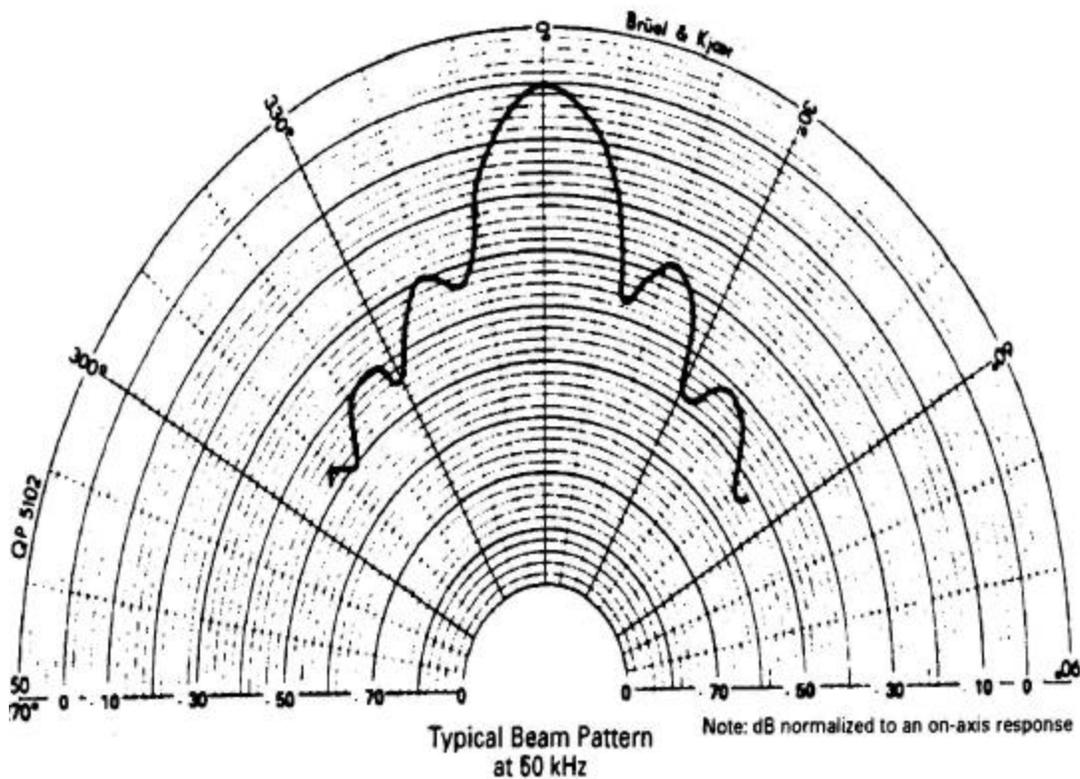
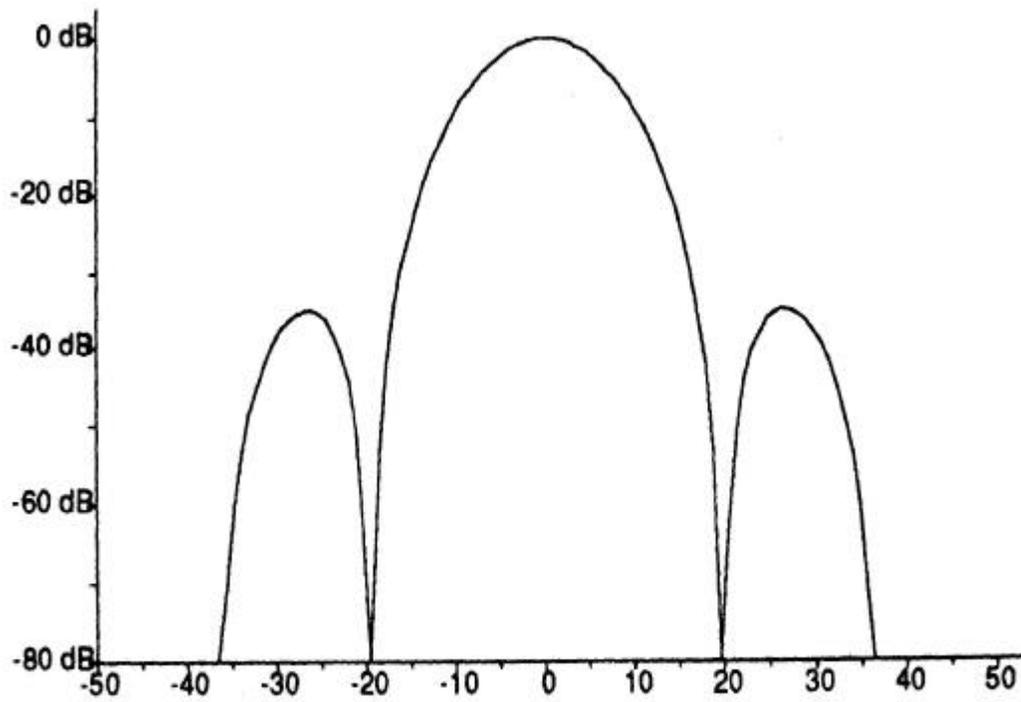


Fig 10. Typical Sonar Beam Pattern

## 5.6 Previous Work in Sonar Based Navigation

Ultrasonic ranging sensors have been used to obtain ranging and imaging information for many years. For example naval vessels used sonar sensors to detect submarines as early as in 1940s. Polaroid and Ultrasonic sensors have been used on many mobile robots to determine distance from obstacles.



Main-lobe and first side-lobes of the plane circular piston radiation pattern.

**Fig 11. Radiation Pattern of Polaroid Sonar**

Drexel University developed an experimental device, which actually eliminates positional inaccuracy in mobile robots by actually finding the distance from walls and known obstacles using rotating sonar<sup>28</sup>. Recalibration of the system was done 'on the fly' by measuring distance from known obstacles.

The French HILARE robot used 14 narrow angle ultrasonic sensors placed at various locations with the transmission angle of the detectors at 60 degrees and the receiving angle at 15 degrees to eliminate the blind spots around the perimeter of the robot<sup>29</sup>.

Rencken<sup>30, 31</sup> at Siemens Corporate Research and Development center in Munich, Germany used 24 ultrasonic sensors to collect features about the obstacles seen. He then constructs hypotheses about these features. Once a feature is confirmed it is constructed into a map. This map of the environment actually solves the bootstrap problem resulting from the uncertainty in position when a mobile robot conquers and unknown environment.

Wen-chuan Chiang, Dhyana Chandra Ramamurthy and T. Nathan Mundhenk<sup>32</sup> actually used a Polaroid ultrasonic transducer to generate a range map for the obstructed path of an autonomous guided vehicle.

Kleeman<sup>33</sup> used ultrasonic Beacons for optimally estimating the position and heading of the mobile robot by localization using dead reckoning.

Beckerman and Obrow<sup>34</sup> have developed a rule-based approach, which deals with the treatment of systematic errors while doing a world modeling. The scan environment is divided in to cells and each cell is labeled either empty, occupied, unknown or conflict. Conflicts may occur when same object is observed at different times from different locations.

Markus Buchberger, Klaus-Werner Jörg, Edwald von Puttmaker<sup>35</sup> did a sonar based world modeling and motion control for obstacle avoidance mechanism of MOBOT-IV robot. The mechanism utilized heterogeneous information obtained from a laser scanner and sonar sensor system in order to achieve a reliable and complete world model for real time collision avoidance.

Nagashima and Yutra<sup>36</sup> used multiple ultrasonic range readers to ascertain the position of an object with greater precision by combining the readings from them.

Similarly Peremans, Audanaert and Van Campenhout<sup>37</sup> also used sonar based localization technique called Sonar based Tri-aural Perception.

## **5.5 System Description**

For the piezo transducer the aluminum cone is the moving element, and the attached piezo material provides the drive force.

The drive electronic is composed of two important components

- (a) Digital and
- (b) Analog

Digital electronics generate ultrasonic drive frequency, which in default mode will be 16 pulses at 52 kHz. In addition, other functions such as blanking time, analog gain control, repetition rate and detect circuitry are all given by the digital electronics. The digital electronics has a Intel 80C196 microprocessor.

The analog circuitry is a variable gain, variable Q system. Because that there is a reduction in return signal strength over long distances a dual role is played by the analog amplifier. This analog amplifier not only processes the echo but also performs the function of maintaining a tailored sensitivity over the entire operation of the system. High ranges may require more amplification than the low ranges. All analog functions are performed by custom Polaroid integrated circuit 614906.

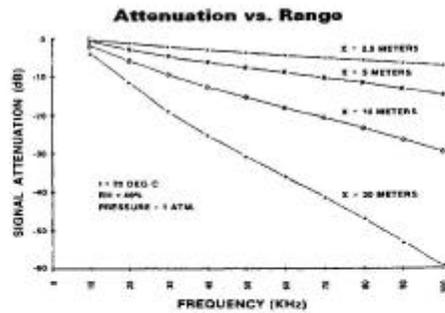
Since the performance of the Polaroid sonar system is dependent on the environmental factors like temperature and humidity, the effect of these factors should be studied.

Fig. 9 shows the attenuation rate with respect to the frequency and humidity.

## **5.6 Practical Considerations**

The sonar model used here is approximated to a ray-trace scanner model. This approximation retains its validity considering the application and the calculation requirements. Also since the distance as measured and given by sonar TOF will not necessarily be the true range of the object the average of different scans could be approximated to it. Also the real world is a complex mixture of specular and diffuse targets, however since our scope is restricted to identifying the leader and follow it, the problems because of weaker reflections could certainly be overlooked. Since Kuc<sup>38</sup> predicts that in a specular environment comprised of planes, corners and edges, sonar measurement should be comprised of sequence of headings at which range value measured is constant, the constant measured readings of sonar could be considered to be

the reflections from regions of constant depth which should reflect the true range of the object.



**Attenuation vs. Temperature/relative humidity**

deg F	15%	20%	35%	30%	35%	40%	45%	50%	55%	60%	65%	70%
63	.2162	.2570	.2984	.3387	.3764	.4104	.4398	.4642	.4833	.4975	.5070	.5124
68	.2456	.2973	.3490	.3950	.4362	.4703	.4968	.5157	.5278	.5335	.5343	.5311
73	.2823	.3462	.4006	.4567	.4972	.5284	.5447	.5537	.5551	.5485	.5418	.5302
78	.3272	.4033	.4687	.5189	.5526	.5710	.5767	.5730	.5628	.5479	.5309	.5127
83	.3807	.4671	.5330	.5751	.5952	.5981	.5889	.5723	.5518	.5291	.5063	.4842

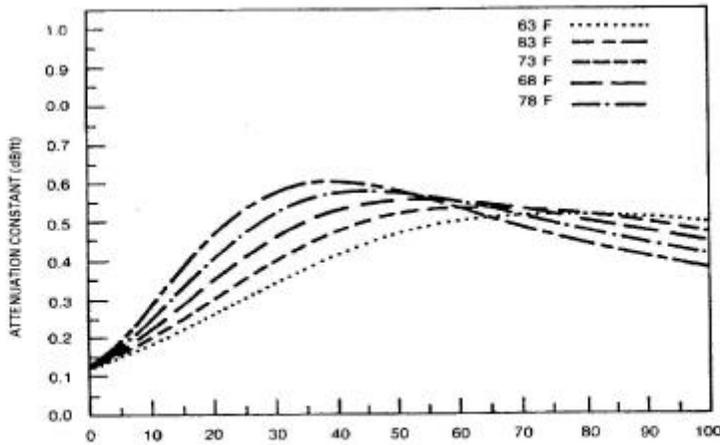


Fig 12. Environmental Factors and their effect on quality of the Signal

## ***6. Algorithm and Logic for Following the Leader***

### **6.1 Theory**

We have seen that the time of flight information provides the distance measurement for the robot and the motion to the robot is actually provided by rotating the two drive wheels using corresponding motors. The orientation of the object with respect to the axis, drive axis of the robot is given by the sonar's orientation. Actually since for a given range of detection's, since only the average of the value would be considered, this should provide the orientation of the object with respect to the robot. The distance between the object and the robot could be set and maintained within a desired distance level so that any change in that distance measurement would either accelerate or decelerate the robot accordingly.

Let

$Z_1$  be the initial distance between robot and the leader.

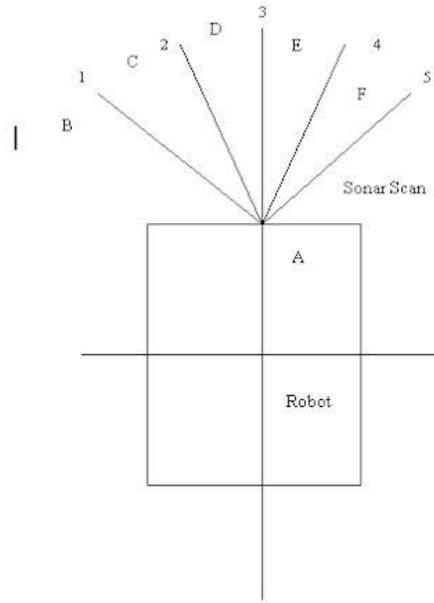
$Z_2$  be the measured distance between the robot and leader now.

$D$  be the diameter of the wheel.

$W$  be the width of the robot.

$\phi$  be the angle between the axis of the robot and the direction of the motion of the leader.

Figure 9 shows the way the sonar detects and tracks the object. The sonar is assumed to have a pentagonal sweeping area. This is shown by ABCDE in the figure, where A is the vertex where sonar is present.



**Fig 13. Schematic Representation of Object Tracking**

The distance AB is assumed to be equal to AE and CD is assumed to be equal to DE.

The distance  $CD = \text{Width of the Robot} + 2 * 6 \text{ in.}$  (clearance on both sides). This assumption has been made to ensure that the sonar tracks a distance greater than its width, which helps in detecting the object when it is very close to the robot. The distance AF is set at 125 inches. This ensures that the robot tracks the object up to the distance of 125 inches. Experimentally it has been found that the sonar reading is more reliable up to the distance of ~150 inches.

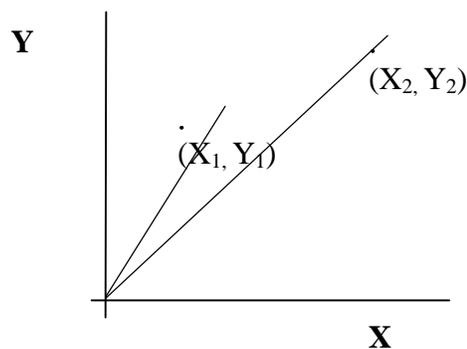
In this approach the following assumptions have been made,

- (a) Counter Clockwise direction is Positive
- (b) Clockwise direction is negative

Although the sonar can be made to stop and scan at any number of points, the fuzzy approach in this algorithm considers 5 positions for Bearcat II. The sonar is made to stop and scan at these fixed positions.

$X_1, Y_1$  be the coordinate of the object first detected.

$X_2, Y_2$  be the coordinate of the object as detected now.



**Fig 14. Ranging representation**

The values of X and Y are determined from sonar's orientation and distance of the object as measured from the sonar. When the object is detected at more than one position then the average of the two sonar positions would give the orientation of the sonar.

Let the sonar detection positions be at angles  $\alpha, \beta, 0, -\beta, -\alpha$  at the positions 1, 2, 3, 4 and 5 respectively, assuming a symmetric scan as discussed above. The angle 0 signifies that it's along the axis of the robot.

For a detection at the position 1 a value is  $-2$  is assigned, for the position 2 a value of  $-1$  is assigned, for the position 0 a value of  $0$  is assigned and for positions 4 and 5 a value of  $+1$  and  $+2$  are assigned respectively.

The logic behind deciding the orientation of the driving direction of the object with respect to the robot is as follows:

Scan using the sonar for all the five positions.

Find the sum of all the values.

If sum =  $-3$  then

$$\text{Sonar detection } \phi = (\alpha + \beta) / 2$$

If sum =  $-2$  then

If detection at position 4 is true

$$\text{Sonar detection angle } \phi = \alpha / 2$$

Else

$$\text{Sonar detection angle } \phi = \alpha$$

If sum =  $-1$  then

$$\text{Sonar detection angle } \phi = \beta / 2$$

If sum =  $0$  then detection angle  $\phi = 0$

If sum =  $1$  then

$$\text{Sonar detection angle } \phi = -\beta / 2$$

If sum =  $2$  then

If detection at position 2 is true

$$\text{Sonar detection angle } \phi = -\alpha/2$$

Else

$$\text{Sonar detection angle } \phi = -\alpha$$

If sum = 3 then

$$\text{Sonar detection angle } \phi = -(\alpha+\beta)/2$$

In all the above cases for the distance of the object the average of the detected values ( $R_1, R_2, R_3, R_4$ ) is selected as  $Z_2$ .

$V_L$  = the speed of the left drive wheel.

$V_R$  = the speed of the right drive wheel.

$V_M$  = the Base speed of the motion of the robot.

$V_{mx}$  = the last recorded base speed of the robot

$T$  = the time interval between the two consecutive sonar reading.

$$V_L + V_R = 2 V_M \quad (1)$$

$$V_L - V_R = W \{d\phi/dt\} \quad (2)$$

Substituting the value of  $V_R$  from the equation (1) into equation (2)

We get

$$V_L - (2 V_M - V_L) = W \{d\phi/dt\}$$

$$2 V_L - 2 V_M = W \{d\phi/dt\}$$

$$V_L - V_M = W/2 \{d\phi/dt\}$$

$$V_L = V_M + W/2 \{d\phi/dt\} \quad (3)$$

Similarly substituting the value of  $V_M$  from equation (1) into equation (2),

We get

$$V_R - (2 V_M - V_R) = W \{d\phi/dt\}$$

$$2 V_R - 2 V_M = W \{d\phi/dt\}$$

$$V_R = V_M + W/2 \{d\phi/dt\} \quad (4)$$

This leads to  $V_L$  and  $V_R$  equations as

$$V_L = V_M + W/2 \{d\phi/dt\}$$

$$V_R = V_M - W/2 \{d\phi/dt\}$$

The Increase or decrease in the distance between the robot and the leader

$$Z_{\text{change}} = Z_2 - Z_1$$

The relative change in velocity is

$$V_{\text{relative}} = Z_2 - Z_1 / T$$

The modified Value for  $V_M$  is

$$V_{M_0} = V_{mx} + (Z_2 - Z_1 / T) (1/\pi D)(86400)$$

Since for the Bearcat II, 2048 counts of encoder makes one revolution and a 36000 counts sets a speed of 25 rpm at the wheel.

The left and right wheel velocities are therefore

$$V_L = V_{M_0} - W/2[(\phi/T)]$$

$$V_R = V_{M_0} + W/2[(\phi/T)]$$

Initially the robot could be made to move at a specified speed, base speed  $V_{mx}$  of 36000 counts/sec and the value of the  $V_{Mo}$  could be dynamically computed to adjust the position and speed of the robot.

## 6.2 Results

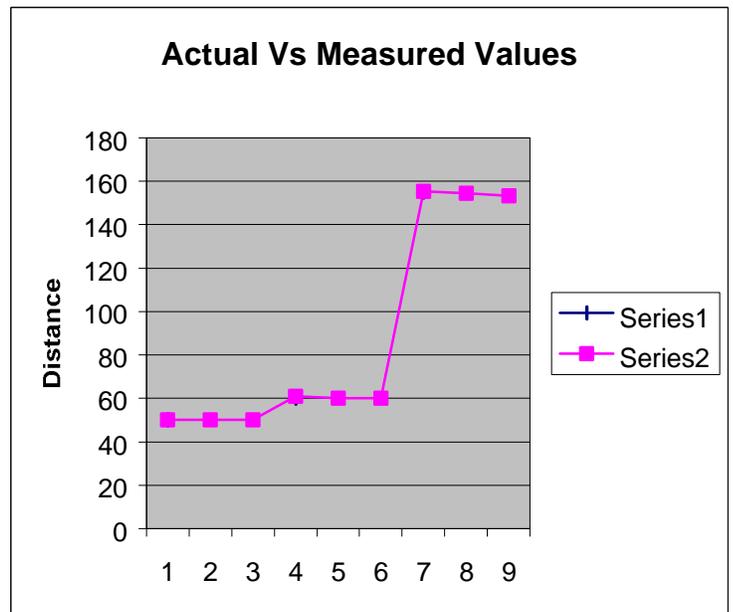
The results of the code and algorithm were tested under laboratory conditions as follows:

- (1) The ultrasonic transducer was mounted on a motor using a bakelite base
- (2) The motor was powered through a DC48A amplifier connected to a Galil Breakout Board ICM 1100 from a Pentium II computer.
- (3) The source code (See chapter 8) was compiled using Borland Turbo C++ compiler on a DOS operating System. The code was also made compatible to Wattcom Compiler, which is a 32 byte compiler system.
- (4) The Robot was jacked up, and was powered using external power supply.

Initially the program was executed in asynchronous mode without having a feedback from the Galil Drive to position the sonar motion and take readings. However after 24-25 cycles there seemed to be a overflow in the GALIL buffer and the sonar motor stopped rotating. This was corrected by making the executable wait till the sonar motor got positioned in the required position, and then the reading was taken from the sonar position. This feedback from the GALIL DMC was achieved by making the GALIL to wait till the motion request was completed and then a value was echoed from the control, which was captured from the read buffer.

Then the sonar was checked for reliability in data by making the sonar position fixed ( instead of rotating it) and the object was placed at different lengths and the readings were

Actual Distance from Robot ( in inches)	Calculated Distance ( in inches )
50	50.20
	50.00
	50.09
60	60.99
	60.10
	60.08
155	155.50
	154.40
	153.30



**Table 1. Distance Measurement using SONAR**

taken. After checking for the reliability of the sonar measurement, the sonar was made to scan at 5 positions about the axis of the robot and the obstacle was placed at different lengths and different positions and the detected distance, angles ( positions of the sonar) at which the object was detected, speedx ( speed of the left wheel ) and speedy (speed of the right wheel) were outputted to the screen and the values were verified by checking the rotation counts of the wheel and the roughly estimating the distance of the obstacle from the robot.

Angle In Degrees	Distance In Inches	Speedx In Galil Counts	Speedy In Galil Counts
0.00	54.00	14000.00	14000.00
	85.00	18000.00	18000.00
-12.00	24.00	15614.00	12386.00
	76.00	19614.00	16386.00
18.00	28.00	11579.00	16421.00
	78.00	15579.00	20421.00

**Table 2. Velocity at wheels for Object at different Positions**

## ***7. IGRC AUVS Competition***

### **7.1 Overview**

Every year IGRC conducts a Automated Unmanned Vehicle System contest. The goal of the contest is to build mobile robots that could navigate itself through some planned and unplanned environment. This competition was started in the year 1991 and has been held every year ever since. There are many events that are held in this contest. The main event challenges mobile robots to navigate following a white line by keeping itself inside a track and avoid obstacles in its path. The basic aim of this main contest is to complete the course. In cases where more than one team completes the course the time factor is accounted for deciding the winner. It is multidisciplinary, theory-based, hands-on, team implemented, outcome assessed, and based on product realization. It encompasses the very latest technologies impacting industrial development and taps subjects of high interest to students. Team organization and leadership are practiced, and there are even roles for team members from business and engineering management, language and graphic arts, and public relations. Students solicit and interact with industrial sponsors who provide component hardware and advice, and in that way get an inside view of industrial design and opportunities for employment.

### **7.2 Events in the IGRC contest**

The three main events in the IGRC contest are Autonomous Challenge Competition, Follow the Leader Competition and the Road Debris Competition. In addition to this

there is also a Design Competition where the design, technology and safety features of the robot are evaluated.

## **7.2.1 Autonomous Challenge Competition**

### 7.2.1.1 Overview

In this event a fully autonomous, unmanned robot is expected to negotiate around an outdoor obstacle course within a prescribed time by restricting itself to a 5 mph speed limit and by avoiding obstacles on the track. The competitors would be judged based on the whether they complete the course. In the event that a vehicle does not finish the course, it will be ranked based on longest adjusted distance traveled. Adjusted time and distance are the net scores given by judges after taking penalties, incurred from obstacle collisions, pothole hits, and boundary crossings, into consideration.

### 7.2.1.2. VEHICLE CONTROL:

The requirement on the vehicles is that it should be unmanned and autonomous. They must compete based on their capability to perceive, understand the course environment and avoid obstacles. Vehicles should not be controlled remotely, during the competition. All computational power, sensing and control equipment must be carried on board the vehicle internally.

Every vehicle must have both a manual and a wireless (RF) remote emergency stop (E-Stop) capability. The effective distance for the E-Stop should be atleast 50 feet. The manual E-Stop must be easy to identify and activate safely, even while the vehicle is moving. Activating the E-Stop must bring the vehicle to a quick and complete stop. The

stopping distance is expected not to exceed 12ft on inclines up to 15%. Vehicles that do not meet the safety requirements would be classified as unsafe and will not be allowed to compete.

#### 7.2.1.3. OBSTACLE COURSE:

The course will be laid out on grass, pavement, simulated pavement, or any combination, over an area of approximately 60 to 80 yards long, by 40 to 60 yards wide. Continuous or dashed white and/or yellow lane markers designate the course boundaries. These lines are approximately three inches wide and will be painted to the ground. Track width will be approximately 10 feet wide with a turning radius not less than 8 feet. Alternating side-to-side dashes will be 15-20 ft. long, with 10-15 ft. separation.

The natural or artificial inclines will not exceed 15% gradient. Sand pits (sand depth 2-3 inches) will be placed randomly along with obstacles in the course. The difficulty of the course will increase as the course progresses. The sand pit may be simulated with a light beige canvas tarp covering the entire width of the track for 10ft. Obstacles on the course will consist of 5-gallon white pails as well as full-size orange and white construction drums that are used on roadways and highways. The places of the obstacles would be randomized and would be replaced before every run. Simulated potholes any location on the course, two-foot diameter white circles.

There are traffic violation tickets associated with this course. They basically define the effective distance covered by reducing the original distance covered because of the violations. The violations for this course and the corresponding tickets are

	Traffic Violations	5 point ticket	E-stop (see below)	Measurement
1	<b>Leave the scene</b> \course all four mech. footings	Yes	Yes	Yes
2.	<b>Crash:</b> obstacle displacement	Yes	Yes	Yes
3.	<b>Careless Driving</b>	Yes	No	No
4.	<b>Sideswipe:</b> obstacle touch (No displacement)	Yes	No	No
5.	<b>Student choice E-stop</b>	Yes	Yes	Yes
6.	<b>Judges choice E-stop</b>	No	Yes	Yes
7.	<b>Simulated pot holes</b>	Yes	No	No

**Table 3. Traffic Violations**

## **7.2.2 Follow the Leader Competition**

### 7.2.2.1 Objective

Automated highway systems require technology that allows vehicles to form and maintain "platoons," in which vehicles autonomously follow each other with very short headway's, as short as two meters. The following vehicles must range to the leading vehicles and maintain the headway very accurately under varying speeds, acceleration, braking, and even emergency stops, while steering to track the highway lane markers. The direct application of this is in military convoys in which only the lead vehicle is manned, while following vehicles not only maintain headway but also steer so as to follow the leader wherever it might go in a trackless course. This competition is designed to simulate both of these real-world requirements.

The lead vehicle will be a riding lawnmower on the back of which is mounted a flat sign 60 cm high by 80 cm wide. The design of the sign is shown in fig 7. The wooden plate will have enamel paint. It will be mounted with its center 75 cm above the ground.

The course will be run on a grassy field approximately 26 by 34 meters with a total travel distance of 112 meters.

There are two stages in this event. Each participant can take up to three tries in this event and the best-recorded distance will be accounted. Only after completing the first course the contestant will become eligible for the second course. The first stage will be run at constant speed, and the second at variable speeds.

#### 7.2.2.2 First Stage

##### Phase I - Headway

The first course is an S-curve course of length 15.4 meter and with a mean radius of about 3 meters. The Challenge will start with the front of the following vehicle 3 meters behind the sign on the rear of the lead vehicle, with both vehicles stationary. The lead vehicle will then accelerate slowly to 3 km per hour and will make a constant speed drive at 3 km to the end of the run. When the lead vehicle has covered around 15.4 meters, a measurement will be made of the distance between the two vehicles (from the center of the sign on the lead vehicle to the front center of the following vehicle). The lead vehicle will then start up again and proceed into the second phase of the Challenge.

##### **Phase II - Headway Maintenance and Free Following**

At the end of Phase I the course will continue on a track visible only to the lead vehicle, but which will consist of S-curves. The curves will have a radius of curvature from 3 meters to 5 meters. The lead vehicle will run at a constant 3 kph for a distance of 96 meters at the end of which it will stop with its rear end at the finish line, and a second measurement of distance between the two vehicles will be made. During the course the lead vehicle will pass within 1.5 from a barrel just to mislead the following vehicle.

#### 8.2.2.3 Second Stage

The second stage is very similar to the first with the only difference being that the lead

Vehicle here will travel at a variable speed from zero to 3 miles per hour. This means that the lead vehicle can come to a stop anytime during the course and proceed again. The following vehicle should be able to negotiate acceleration, deceleration and stops.

The judges will terminate a run, if the following vehicle touches any part of the lead vehicle or falls behind the lead vehicle by 6 meters, as determined by the judges. It will also be terminated if a judge walking behind the following vehicle at a distance equal to 3 vehicle widths, on the axis of the following vehicle, determines that the following vehicle has deviated far enough from following that his line of sight shows an open space between the two vehicles.

The winner for the First Course will be the one who complete the course and the score will be the total distance traveled. If no vehicles complete the course, the one traveling the greatest distance will be declared the winner. For those vehicles that complete the entire course, the two measured distances from the lead vehicle (at the end of Phase I and at the end of the course) will be averaged, and the smallest deviation of this average from 3 meters will determine the first stage winner.

### **7.2.3 Road Debris Competition**

In the 9<sup>th</sup> AUVS contest this competition is made as an Navigation Challenge competition. The challenge in this event is for a vehicle to autonomously travel from a starting point to several target destinations (waypoints or landmarks) and return to home base, given only a map showing the coordinates of those targets. Contestants are expected to use GPS, but may augment that technology with compasses and wheel odometers (use

of vision systems is not allowed). Coordinates of the targets will be given in latitude and longitude as well as in meters on an x-y grid.

### **7.3 Overview of the Results of 2000 Competition:**

Bearcat II robot performed extremely well in the 8th International Ground Robotics Contest, July 7-10,2000, Disney Coronado Springs Hotel, Orlando. It strongly participated in all the events and won prizes. The 8<sup>th</sup> AUVS competition was more challenging because a new event named Pothole Problem was introduced. This basically made the course in the Autonomous challenge competition more challenging by bringing in potholes in the track in certain regions and robot was expected to complete the course without getting into any of these. In the Autonomous Challenge Competition it passed through various terrain with much ease, avoided obstacles with very little difficulty, climbed ramps smoothly, negotiated curves with ease and covered about 180 feet before going off its track. It was a memorable performance by the Bearcat II robot and the whole teams were elated seeing their hard work realize. It secured the 3<sup>rd</sup> place leaving the 2<sup>nd</sup> place to the Hosei University, Amigo Japan only by 5.75 feet. Virginia Tech came first covering about 232.5 feet. In the road debris competition again Bearcat II performed beyond expectations and came 5<sup>th</sup> covering about 27.83 feet.

### **7.4 Preparation for 2001 contest**

For the 2001 contest lot of additional capabilities have been introduced and many existing technologies have been fine-tuned. Some of the new technologies that have been introduced are global positioning system (GPS) and laser sick scanner (LMS 200). Also an additional 36 V inverter had been added to increase the running time of the robot. The

team is working hard on implementing the new laser scanner solution for the obstacle avoidance algorithm. The pothole detection algorithm has been refined and together with this sonar based follow the leader, BEARCAT III and the team are once again ready to face the technical challenge and prove their mettle in the IGRC 2001 competition.

## ***8. Conclusion and Recommendation***

This work basically uses the Cartesian co-ordinate system with a plant model to detect the leader, which is the single target, and ascertains its position. However to record the path the robot has followed and to actually ascertain the absolute position of the robot at any point of time, a more dynamic model which would include a more sophisticated error modeling technique could be used. This work on single target tracking could also be extended to a multi target targeting by using a probability model for the detection of a target based on the data to data association hypothesis as suggested by K. Chang, S. Mori and C. Chong<sup>39</sup> in their multi-target tracking algorithm. The robot could also be made to trace a map, with information on the location of landmarks. This method of navigation of mobile robots using sonar sensors has been experimented and found to be successful. Instead of using a single rotating sonar, a series of mounted sonar's could be used. This would help to absolutely ascertain the position and the orientation of the object. Also by having two responses from a object it is also quite possible to eliminate the errors that are introduced because of difficulty in differentiating between the reflections from edges and corners and corners and planes. The work by Barshan and Kuc<sup>40</sup> could be made use of as a guiding work in that related area. Also a sonar fusion solution coupled with the use of a omni-directional camera would be able to avoid misleading signals and might actually provide a correcting mechanism for validating the robots position and its relative position with respect to the object or leader.

# *References*

- [1] Peter Kampmann and Gunther Schmidt, Indoor Navigation of Mobile Robots by use of Learned Maps, Information Processing in Autonomous Mobile Robots, Springer Verlag, Berlin, Heidelberg, New York, 1991.
- [2] T.Knieriemen and E.von Puttkamer, Real-Time Control in an Autonomous Mobile Robot, Information Processing in Autonomous Mobile Robots, Springer Verlag, Berlin, Heidelberg, New York, 1991, pp. 187-200.
- [3] E.I.Knudsen, Auditory and Visual Maps of Space in the Optic Tectum of the Owl, Journal of Neurosciences, 1990 Vol 2, pp. 1177-1194
- [4] Alberto Elfes, Sonar-Based Real-World Mapping and Navigation, IEEE, Journal of Robotics and Automation, 1987 RA Vol.3, Issue 3, pp. 249-265.
- [5] B.Yamuchi and P.Langley, Place Recognition in Dynamic Environments, Journal of Robotic Systems, Special Issue on Mobile Robots, Vol.14, No.2, pp. 107-120. 1996.
- [6] A.Stevens, M.Stevens and H.Durrant-Whyte, 'OxNav': Reliable Autonomous Navigation, Proc. IEEE International Conference on Robotics and Automation, Vol.3, pp. 2607-2612. NJ 1995.
- [7] S.Maeyama, A. Ohya, S.Yuta, Non-stop Outdoor Navigation of a Mobile Robot, Proc. International Conference on Intelligent Robots and Systems (IROS) 95, pp. 130-135. PA 1995.
- [8] MTI-MTI Research, Inc., Chelmsford, MA
- [9] Teuvo Kohonen, Self organization and Associative Memory, Springer Verlag, Berlin, Heidelberg, NY, 2<sup>nd</sup> Edition, 1988.

- [10] A.Kurz, Constructing Maps for Mobile Robot Navigation based on Ultrasonic Range Data, IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics, Vol. 26, No: 2, pp. 233-242, 1996.
- [11] Matthias Franz, Bernhard Schölkopf, Hanspeter Mallot, Heinrich Bülthoff, Learning View Graphs for Robot Navigation, Autonomous Robots, Vol. 5, pp. 111-125, 1998.
- [12] J.Tani and N. Fukumara, Learning Goal-Directed Sensory Navigation of a Mobile Robot, Neural Networks, Vol.7, No. 3, pp. 553-563, 1994.
- [13] B.J. Kuipers and Y.T. Byun, A Robust, Qualitative Method for Robot Spatial Learning, Proc.AAAI, pp. 774-779, Morgan Kaufman, San Mateo CA, 1988.
- [14] M. Jenkin, et.al, "Global Navigation for ARK", Proc. IEEE/RSJ International Conference on Intelligent Robotic Systems, Yokohama, Japan, 1993, pp. 2165-2171.
- [15] P.F.Muir and C.P.Neuman, "Kinematic Modeling of Wheeled Mobile Robots", Journal of Robotic Systems, 4(2), 1987, pp. 281-340.
- [16] E.L. Hall and B.C. Hall; "Robotics: A User-Friendly Introduction", Holt, Rinehart and Winston, New York, NY, 1985, pp. 23.
- [17] Z.L.Cao, S.J. Oh, and E.L.Hall, Dynamic omni-directional vision for mobile robots, Journal of Robotic Systems, 3(1), pp.5-17, 1986.
- [18] Z.L. Cao, Y.Y. Huang and E.L. Hall; Region filling operations with Random Obstacle Avoidance for Mobile robots, Journal of Robotic Systems, 5(2), 1988, pp. 87-102.
- [19] Polaroid Corporation, Ultrasonic Ranging System, Cambridge, Massachusetts (1992).
- [20] Galil Inc., "DMC 1000 Technical Reference Guide Ver. 1.1", Sunnyvale, CA

1993.

[21] Kolli, C.K et.al; “Design Development of Steering and Speed Control for an Intelligent Mobile Robot”, Proceedings of SPIE, Intelligent Robotics and Computer Vision, Boston, November 1997.

[22] R. Urick, Principles of Underwater Sound, McGraw-Hill, New York, 1983

[23] R. Hinkel, T. Knieriemen, E. von Puttkamer, Mobot-II an autonomous mobile robot for indoor applications, International Symposium and Exhibition on Robots, Sydney, Australia, 1988.

[24] A.M.Flynn, Combining Sonar and Infrared sensors for mobile robot navigation, International Journal of Robotics Research, 7(6), 1988.

[25] J.Connell, A Colony Architecture for an Artificial Creature, PhD Thesis, Massachusetts Institute of Technology, 1989.

[26] John J. Leonard, Hugh F. Durrant-Whyte, Directed Sonar Sensing for Mobile Robot Navigation, Kluwer Academic Publishers, Boston, 1992.

[27] P.M. Morse and K.U. Ingard, Theoretical Acoustics, McGraw-Hill, 1968.

[28] Einbinder, J Kirshtein, et al., Electronic Droid With Intelligence (EDWIN): An Autonomous Mobile Robot Prototype, Final Report, Senior Design Project ECE 3, Drexel University, Philadelphia, May 1983.

[29] Bauzil, Briot and Ribes, A Navigation Sub-system using Ultrasonic Sensors for Mobile Robot HILARE, Proceedings of the First Annual Conference on Robotic Vision and Sensory Control, Stratford-Upon-Avon, England, April 1981.

[30] W.D.Rencken, Concurrent Localization and Map Building for Mobile Robots using Ultrasonic Sensors, Proceedings of the 1993 IEEE/RSJ Conference on Intelligent Robots and Systems, pp. 2192-2197, 1993.

- [31] W.D.Rencken, Autonomous Sonar Navigation in Indoor, Unknown and Unstructured Environments, Proc. IEEE/RSJ Intelligent Robots and Systems, Munich, Germany, 1994, pp. 127-134.
- [32] Wen-chuan Chiang, Dhyana Chandra Ramamurthy, et al., Range Detection for AGV using a Rotating Sonar Sensor, SPIE Conference, Boston 1999.
- [33] L. Kleeman, Optimal Estimation of Position and Heading for Mobile Robots using Ultrasonic Beacons and Dead-Reckoning, Proc. of IEEE conference on Robotics and Automation, 1992, pp. 2582-2587.
- [34] Beckerman, M., Oblow, E.M., Treatment of Systematic Errors in the Processing of Wide-angle sonar sensor data for Robotic Navigation, IEEE Transactions on Robotics and Automation, Vol. 6, No: 2, April 1990.
- [35] M. Buchberger, K. Jorg, E.Puttkamer, Laser radar and sonar based world modeling and motion control for fast obstacle avoidance of the autonomous mobile robot MOBOT-IV, Proc. IEEE International Conference on Robotics and Automation, Atlanta, GA, 1993, pp 534-540.
- [36] Nagashima, Y., Yutra, S., Ultrasonic sensing for a mobile robot to recognize an environment – Measuring the normal direction of walls, Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, 1992, pp. 1025-1029.
- [37] H. Peremans, K. Audenaert, Van Campenhout, A High Resolution Sensor Based on Tri-aural perception, IEEE Transactions on Robotics and Automation, 9(1), 1993, pp. 36-48.
- [38] R. Kuc, A spatial sampling criterion for sonar obstacle detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-12 (7), July. 1990, pp. 686-690.

[39] K. Chang, S. Mori, C. Chang, Performance evaluation of a multiple hypothesis multitarget tracking algorithm, IEEE International Conference on Decision and Control, 1990. pp. 2258-2263.

[40] B.Barshan, R. Kuc, Differentiating sonar reflections from corners and planes by employing an intelligent sensor, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 6, June 1990, pp. 560-569.

# *Appendix A*

## *Relevant Programs*

### **# #Car.c**

```
#include <time.h>
#include <string.h>
#include <iostream.h>
#include <math.h>
#include <stdlib.h>
#include "globals.h"
/* Turbo C/DOS extension headers */
#include <constrea.h>

#include "g-cart.h"
#include "car.h"

extern time_t first;

void speedCARx(long int spx)
{
    //This command when invoked will set the speed of bearcat II.
    //It is called with a value between 0 - 250000.
    char inx[10],sendx[15];
    gcvt(spx,6,inx);
    char *commandx="JG";
    strcpy(sendx,commandx);
    strcat(sendx,inx);
    gotoxy(10,21);
    cout<< " Left-motor: ";
    gotoxy(24,21);
    cout << sendx;
    gotoxy(1,23);
    cout << "X-motor --> ";
    //if(!TEST)
        submitDMC(sendx);
        submitDMC("BGX");
}

void speedCARY(long int spy){
    //This command when invoked will set the speed of bearcat II.
```



```

        spdy = 9000;
        }
        else
        {

            spdx = 36000 +((134.5*val)/1);
            spdy = 36000 -((134.5*val)/1);
            }
    }
    else if (val >= 24 )
    {
        if(distance <125)
        {
            spdx = 4000;
            spdy = 10000;
        }
    }
    else
    {

        spdx = 36000 +((134.5*val)/1);
        spdy = 36000 -((134.5*val)/1);
        }

    }
    else if( distance > 75 )
    {
        if (distance < 125)
        {
            spdx=18000+((134.5*val)/1);
            spdy=18000-((134.5*val)/1);
            }
        else
        {

            spdx = 36000 +((134.5*val)/1);
            spdy = 36000 -((134.5*val)/1);

            }

        }
    }
    else if ( distance < 75)
    {
        if(distance < 20 ) stopCAR();
        else
        {
            spdx = 14000 + ((134.5*val)/1);
            spdy = 14000 - ((134.5*val)/1);
            }
    }

```

```

    }
    if (spdx > 36000)spdx =36000;
    if (spdx < -36000)spdx =-36000;
    if (spdy > 36000)spdy =36000;
    if (spdy < -36000)spdy =-36000;

//    cout << "\t\tspx = " << spdx << " ";
//    cout << "spdy = " << spdy << " ";
//cout << "increment = " << (int)((134.5*val)/dtime) << " ";
    speedCARx(spdx);
    speedCARy(spdy);
}

```

## ## RotatSonar.c

```

#include <iostream.h>
#include <stdlib.h>

#include <constrea.h>

#include "g-cart.h"
#include "rotatson.h"
#include "galil.h"
#include "car.h"

int positionz;
float avgdist;
int maxStop =5;

int rotatson()
{

float avgdist =0;
positionz = 0;
avgdist =0;
angleC =0;
//initsonarmotor();

for (int i =0; i < maxStop; i++)
normalsweep();

angleC = compute_angle();
avgdist = comp_avg_dist();

```

```

if(kbhit())
{
    if (getch() == 27)
    {
        submitDMC("PA,,0");
        exit(0);
    }
}

```

```

speedCAR(angleC, avgdist);

```

```

}

```

```

initsonarmotor()
{
    submitDMC("SHZ");
    submitDMC("AC,,350000");
    submitDMC("SP,,350000");
    submitDMC("DP,,0");
    submitDMC("DP,,0");
}

```

```

float normalsweep()
{
    static int sonarposindex=0;
    positionz = 70 * sonarposindex -140;
    float distance =0;
    positionsonar(positionz);
    //gotoxy (1,1);
    distance = snr();
}

```

```

if (sonarposindex == maxStop-1)
    sonarposindex = 0;
else
    sonarposindex++;

```

```

if (distance <= 150 && distance > 0)
{

```

```

    switch (sonarposindex)
    {
        case 0:
            detect[sonposindex] =2;

```

```

        obsdata[sonposindex] = distance;
        break;
    case 1:
        detect[sonposindex] = 1;
        obsdata[sonposindex] = distance;
        break;
    case 2:
        detect[sonposindex] = 0;
        obsdata[sonposindex] = distance;
        break;
    case 3:
        detect[sonposindex] = -1;
        obsdata[sonposindex] = distance;
        break;
    case 4:
        detect[sonarposindex] = -2;
        obsdata[sonposindex] = distance;
        break;
    }

}

else{
    detect[sonarposindex] = 0;
    obsdata[sonposindex] = 0;
}

}

```

## ## Sonar functions

```
#define port2 0x02f8 //COM2 PORT
```

```

void positionsonar(int posz)
{
    char inz[10], sendz[15];
    gcvt(posz, 6, inz);
    char *commandz = "PA,,";
    strcpy(sendz, commandz);
}

```

```

        strcat(sendz,inz);
        submitDMC(sendz);
submitDMC("SHZ");
        submitDMC("BGZ");
submitDMC("WT 500");
submitDMC("V=3;AMZ;V=");
getDMCvalue();
}

```

```

void submitDMC(char str1[15])
{
//This function when invoked will send the DMC a command, and if the command is
//legal and there are no errors it will say so.
    char error;
    sendDMC(str1);

//check for colon:
    error=fromDMC();
    if(error!=':')
        cout<<"GALIL: Error!! ""<<error<<"" << str1 <<"\n";
    else
        cout<<"GALIL: Command Received\t";
}

```

```

int getDMCValue()
{

    int j;
    while((j=int(inp(1000))) != '3')
    {

    }
    return 0;
}

```

```

void sendDMC(char str1[])
{
//This function when invoked will send a string to the DMC.
//Add a null character to the string
    char send[15];
    char *command=str1, *null = NULL;
    strcpy(send, command);
}

```

```

    strcat(send, null);

    if(galilIsReady())
    {
        //Send character by character to DMC
        int i;
        for(i=0; send[i]; ++i) {writeDMC(send[i]);};
        writeDMC(13);
    }
}

void writeStatus(char a)
{
    outp(DMC_Status,a);
}

int readStatus()
{
    return(inp(DMC_Status));
}

int galilHasInfo()
{
    return((inp(DMC_Status) & 32)== 0);
}

int galilIsReady()
{
    return((inp(DMC_Status)&16)==0);
}

char getDMC()
{
    //This function when invoked will confirm data is available and the will
    //retrieve it.

    if(galilHasInfo() )
        return(readDMC());
    else
        return(0);
}

```

```

}

void initDMC()
{
//This function when invoked will prepare the DMC by initializing it to 128 FIFO

    // Set FIFO buffer to 128 bytes
    writeStatus(5);
    writeStatus(0);
    // clear dmc buffer
    clearDMC();
}

void clearDMC()
{
//This function when called clears the DMC buffer.

    writeStatus(0x01);
    writeStatus(0x80);
    writeStatus(0x80);
}

float snr()
{
    float dis;
    unsigned char buffer[6];
    unsigned char value1,value2;
    unsigned int i,j;

    outportb(port2,'@' );

    while((value2=inportb(port2))!=12);

    for (i=0;i<6;i++)

    {
        while((inportb(0x02fd) & 0x01) != 0x01);
        buffer[i]=inportb(port2);
    }

    dis=atof(buffer);
    return dis;
}

```

```

float compute_angle()
{
int sum =0;
float angleC =0;
while (i < maxStop)
    sum += detect[i];
//A 48 degree, odd number of positions evenly spaced about the robot axis in straight
forward
//position is assumed.

switch (sum) {

case -3:
    //Between the -24 deg and -12 deg position
    angleC = 18;
    break;
case -2:
    // There can be two cases involved

    if (detect[4] == -2 && detect[3] == -1 && detect [1] == 1)
    {

        angleC = 12;
    }
    else
    {
        angleC = 24;
    }
    break;

case -1:
    angleC = 6;
    break;
case 0:
    angleC = 0;
    break;
case 1:
    angleC = -6;
    break;
case 2:
    if (detect[0] == 2 && detect[1] == 1 and detect[3] == -1)
    {
        angleC = -12;
    }
    else{

```

```

        angleC = -24;
    }
    break;
case 3:
    angleC = 18;
    break;
default:
    break;
}

return angleC;

}

void initialize_detect() {

for (int i=0; i <= maxStop; i++)
    detect[i] = 0;

}

float compute_avg_dist()
{

float sum =0;
for (int i =0; i< maxStop; i++)
    sum += absdata[i];
return sum/maxStop;
}

```