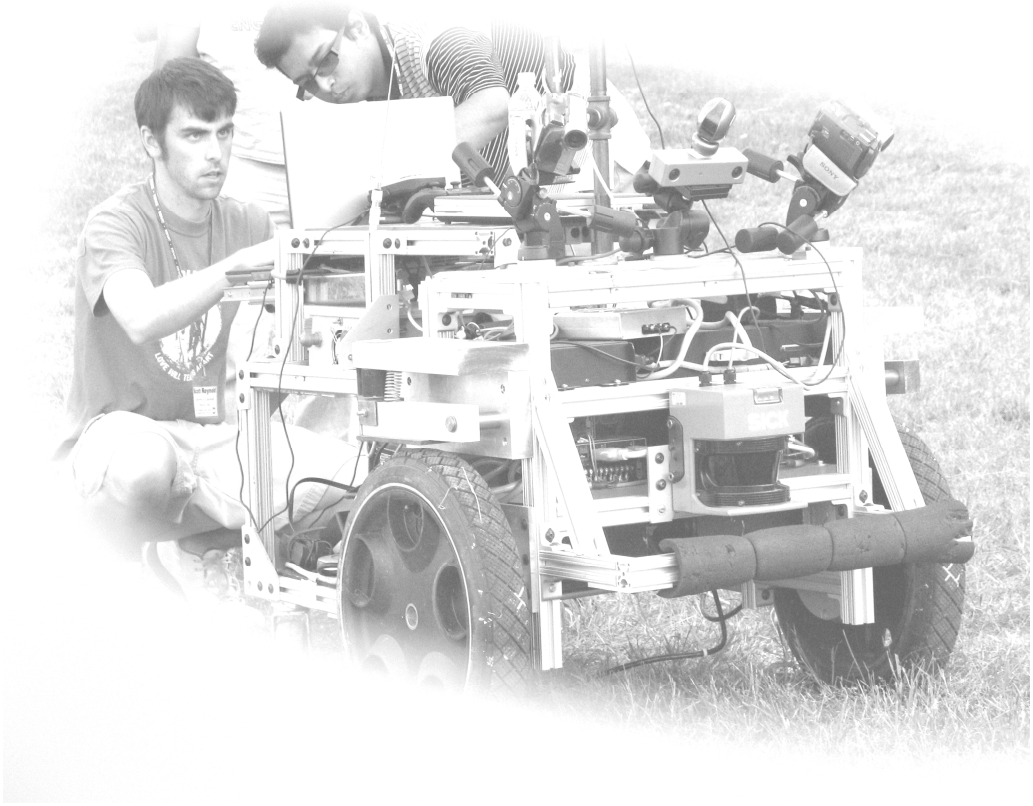
The 16th Annual Intelligent Ground Vehicle Competition
May 30 - June 2, 2008

Bearcat Cub
University of Cincinnati



CERTIFICATION

I certify that the engineering design in the vehicle Bearcat Cub (original and changes) by the current student team identified in this Design Report has been significant and equivalent to what might be awarded credit in a senior design course.

_____

Dr. Ernest L. Hall, Advisor

UNIVERSITY OF Cincinnati

Introduction

This year marks the 16th year that the University of Cincinnati Robotics Team has participated in the IGVC. This year's robot, which is based on the 2007 Cub robot, has been developed by a collaborative effort from a multidisciplinary team. It has undergone major changes in it's software along with changes in the support structure from the previous year's entry into the competition. New hardware was added to make it's functioning more predictable and reliable. This report describes the various aspects of Bearcat Cub's design, design trade-off considerations and improvements over the past IGVC entries by the UC Robotics team.

Design Innovations

The Bearcat Cub this year went through major revisions not only in the software design but also in its frame work which provides more space and accessibility. It now has a robust system that is implemented in C# which can be expanded easily to accommodate new sensors and new planning algorithms. The team added new software to create accurate global maps and for placing obstacles at their true latitude and longitude position. Using this new map, the vector based planner has been improved and has been made more efficient. Our vision system is now more robust using two cameras to detect the lanes instead of one wide-angle camera. The software has been implemented in a whole new architecture.

Design Process and Team Organization

Our designs were developed using basic Agile techniques. We held SCRUM meetings twice a week to discuss plans, possible pitfalls and to update ourselves with current progress. This year the IGVC team consists graduate as well as undergraduate students under advisor Dr Ernest L.Hall.

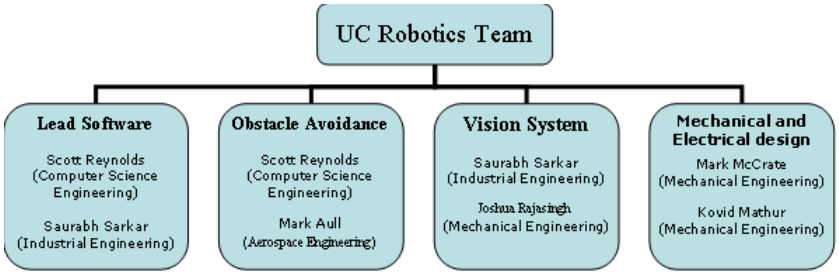


Figure 1: Team Organization

Task	Start	Finish	2007					2008				
			Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May
Analyze Existing Design	8/17/07	11/1/07										
Design Mechanical and Electrical System	10/5/07	12/2/07										
Implement Mechanical and Electrical System	12/2/07	12/10/07										
Design Autonomous Navigation system	12/15/07	2/10/08										
Implement Autonomous Navigation system	15/12/07	2/15/08										
Test Autonomous Navigation system	1/5/08	3/18/08										
Design Waypoint Navigation system	10/3/07	1/25/08										
Implement Waypoint Navigation system	1/3/08	3/12/08										
Test Waypoint Navigation system	3/15/08	3/30/08										
System integration and testing	3/2/08	4/30/08										
Final Design report	5/1/08	5/5/08										

Figure 2: Gantt Chart

This report is divided into sections, each explaining the different modules of the robot and can be categorized as following.

1. Hardware Design: This section describes the basic platform along with the hardware components which includes the framework, power system, the emergency stop and the motion control system.
2. Electrical and Electronics system: The section lists out in brief the computer system and the various sensors with schematics of its integration.
3. Software design: Describes in detail the algorithm used for mapping, lane detection, the vector field approach and path planning.

1. Hardware

Framework

The frame forms the base structure on which the Bearcat Cub is built. It is built of light weight 80/20 aluminum extrusion which is light and can be used without compromising the strength of the frame structure. The advantage of using this frame is the ease of reshaping and addition of new components as there are developments in the design. The design this year was modified for improved space utilization and easy accessibility.

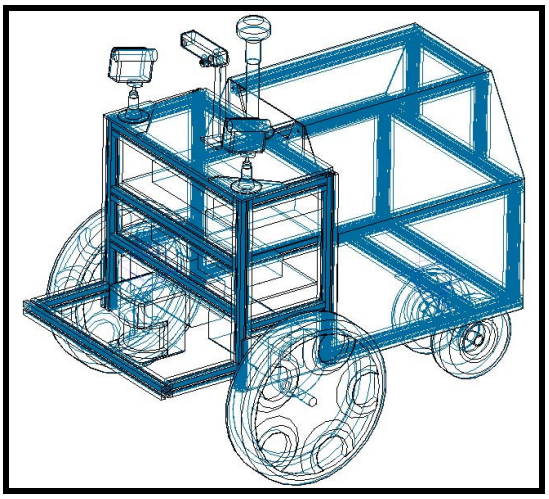


Figure 3: Basic Framework

The Cub has two types of wheels – two main drive wheels and two rear castor wheels. The 19 inch drive wheels are enhanced traction wheels designed by Michelin. They consist of a forged steel wheel hub with a glass-reinforced thermoplastic rim. The tires are made of a silica compound, which provides good traction even on wet surfaces.

The 6" rear castor wheels are from Borne & Co provide stability needed for the Cub to perform zero turning radius turns. The robot is designed to run at a maximum speed of 5 miles/hour. A Pacific Scientific PMA43R-00112-00, 2H.P brushless servo motor has been installed on each drive wheel with a gear box of ratio 25:1. The gearbox and motors have been selected based on the design calculations taking frictional coefficient of 0.125 and 70% gearbox efficiency. This design incorporates the motor inside the hub of the wheel resulting in a compact and robust design.

During testing it was found that the robot can run more than 5 mph for safety reasons the speed has been limited to 2 mph. The robot successfully climbed a ramp of 30 degrees.

Power System

The robot's power system comprises of a 2 KW Honda EU-2000i, super quiet generator which can provide 4 ½ hours of continuous power. Also installed is an 800 W inverter with a 12V deep cycle alkaline battery. The generator set allows

more hours of operation compared to 1 ½ hrs of battery power and also reduces down-time since refueling the generator set is much quicker than recharging a battery. This is especially useful during long hours of experimentation on the robot.

Emergency Stop

The robot has both mechanical and electrical brakes integrated in the system. A manual E-stop button is located on the rear of the vehicle more than 2 feet above the ground which can activate the brakes. A Futaba remote control E-stop can also apply the brakes from a distance of 65 feet.

Motion Controller

The Galil DMC 2130 motion control board is used for the Cub and is controlled through commands sent via an Ethernet connection from the laptop. Copley amplifiers deliver power to the motors after amplifying the signals they receive from the motion controller. Steering is achieved by applying differential speeds at the right and left wheels. The Galil motion controller was chosen because it is Ethernet based, has PID and Bode plot tuning software, and is compact and enclosed in a durable package. The controller can accommodate up to 4 axis and can control stepper or servo motors on any combination of axes. The Bearcat Cub has the ability to turn about its drive axis effectively performing a Zero Turning Radius (ZTR) pirouette. The block diagram of the system is shown in Figure 4.

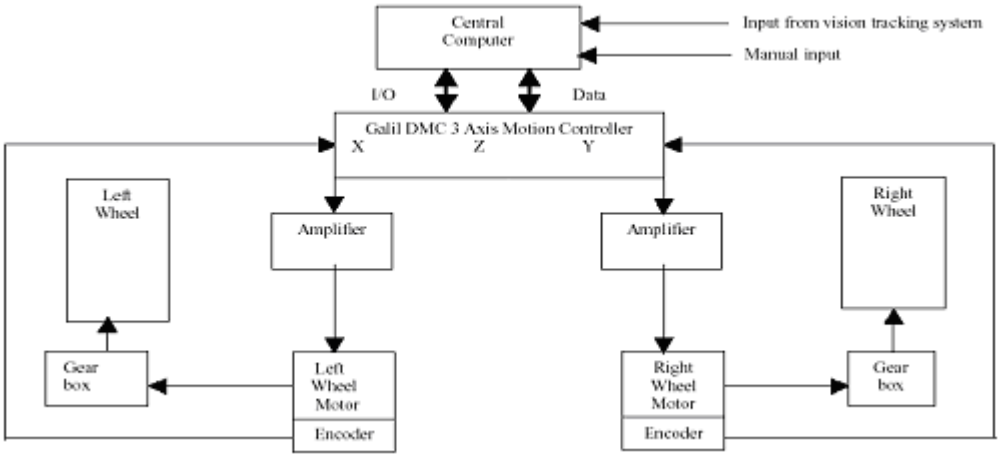


Figure 4: Motion control system

2. Electrical and Electronic Systems

The electrical systems of the Bearcat Cub consists of a motion controller, 2 amplifiers, 2 DC brushless motors, 2 digital cameras, a laser scanner, GPS unit, and an emergency stop. All power is provided by a general purpose gas AC generator which is then converted to DC power by individual power supplies for each of the system. This allows the Bearcat Cub to be outfitted with any set of sensors very easily since there is no need for the end user to customize any power supplies. The system acts like a hardware equivalent of software plug and play. Figure 5 below shows the general electronics layout.

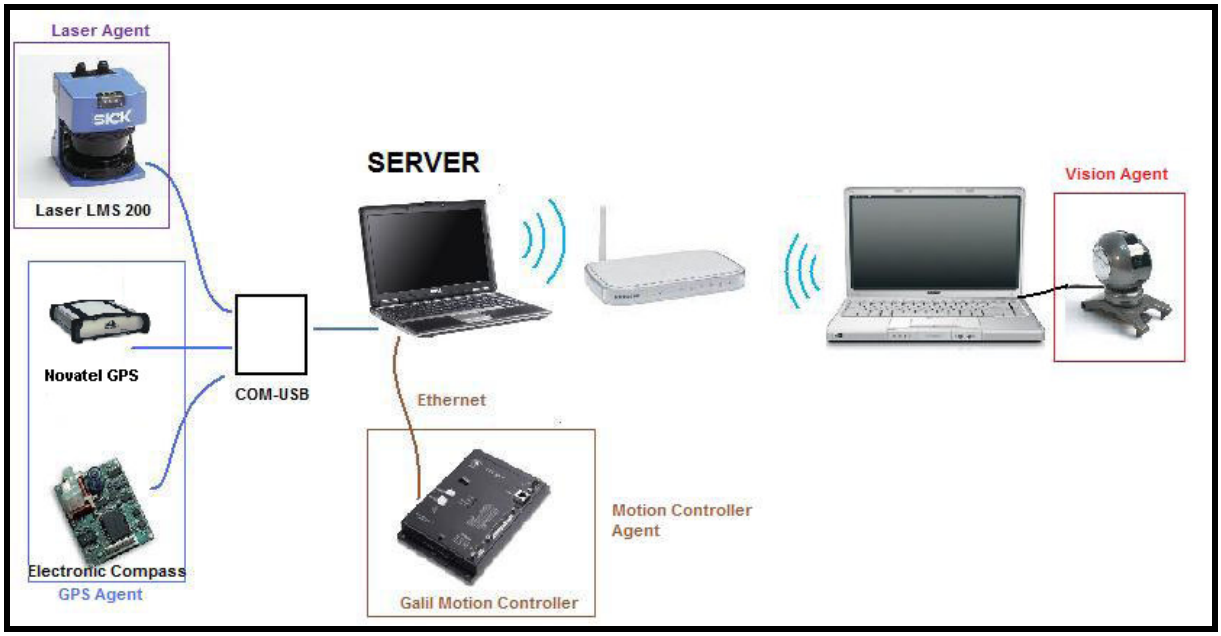


Figure 5: Bearcat Cub block diagram

Computer System

A Dell Latitude D830 laptop is the central processing unit of the Bearcat Cub. It has a 2.6 dual core Intel processor with 3.5GB RAM. It processes data from the laser scanner, GPS, motion control system and image processing system. All control programs have been made in C# taking advantage of the .NET Framework. A user friendly GUI was developed to track the Bearcat Cub's movement and position. A series of initialization files hold all calibration values and initial values for the system parameters.

Sensors

Laser Measurement System

The Sick LMS 200 scans a 2-dimensional plane of 180 degrees and returns obstacle distance measurements for up to 8.191 meters with a infrared laser beam (835 nm wavelength) based on it's time of flight. The resolution of scan is 0.5 degree. It is communicates with the computer using a RS 232 ports with a data transfer rate of 38,400 bauds.

Global Positioning System (GPS)

A Novatel's ProPak-V3 is a durable, high-performance receiver with advanced capabilities and uses USB communication. The accuracy achieved with this unit is 0.6m using SBAS channel.

Cameras

Two video cameras provide the images that are used by the line detection system. The cameras used by the Cub are Sony handy cams.

Compass

Honeywell HMR3200 digital compass is a 2 axis precision compass. The compass is oriented horizontally on the rigid body of the Cub. It provides 1 degree accuracy and operates at 19200 baud rate providing fast and accurate heading information to the robot for accurate path planning.

3. Software

Mapping

The Bearcat Cub keeps track of a map of its surroundings as it moves through the environment. This map consists of all the detected obstacles latitude and longitude positions. Each sensor, running on separate threads, will inform the other parts of the program when an obstacle is detected and the distance the obstacle is from the robot. The map will then use the robots location and heading to calculate the

latitude and longitude of each detected point via the following Equations 1 and 2.

$$x = x_o + (r \times \cos(\theta + \phi) / R) \quad (1)$$

$$y = y_o + (r \times \sin(\theta + \phi) / (R \times \cos(x_o)) \quad (2)$$

Where x_o is the robot's latitude, y_o is the robot's longitude, θ is detected angle of the object from the robot, ϕ is the robot's heading, and *R* is the mean Earth radius in meters. The resulting *x* and *y* is the obstacle's latitude and longitude respectfully.

For each new obstacle detected a line is drawn from the obstacle to the sensor. The map is searched for previously detected obstacles that exist on the line and are between the sensor and the latest detected obstacle. These previously detected obstacles are discarded because the sensor should have detected them with the latest scan but did not and thus are considered as noise from a previous scan. The procedure constantly updates the map with accurate information and is resilient to error generated from noisy position, heading and sensor data.

In order to efficiently update our map in such a way, an R-Tree is used to store the detected points. An R-Tree is a tree structure that is specialized for spatial access. It is designed to be efficient at searching and discovering objects that are within a certain distance of another object. In an R-Tree all the points are stored at leaf nodes. All non-leaf nodes describe a rectangle that encompasses all the points that are below it. This structure enables O(N log N) complexity when searching for obstacles within a certain range of a given point.

Using the R-Tree[2] and the updating algorithm as described above, the map is updated first by querying the R-Tree for points that are within 10 meters of the robot's current position. This prevents the updating algorithm from having to go through all the points that have been detected. Then each of these points are put through the updating algorithm to determine those points should be discarded as noise. Below is a screenshot of the control panel and the map in action.

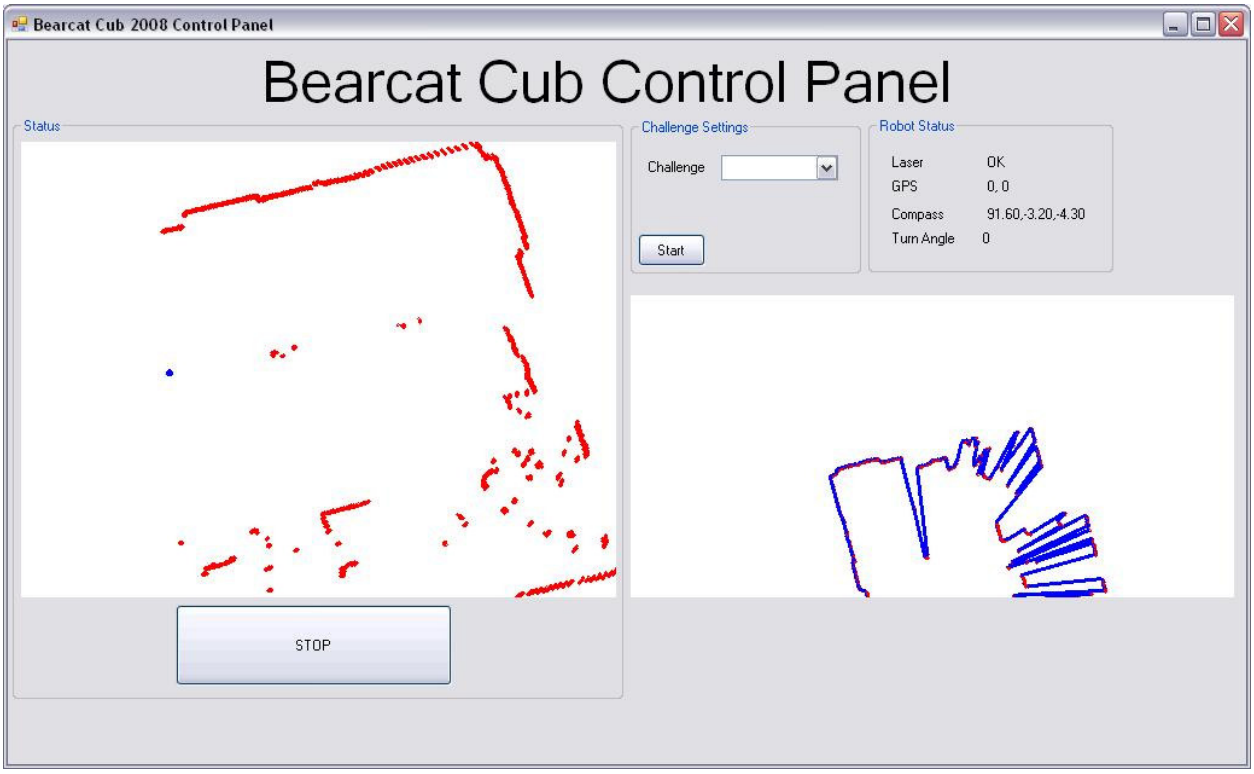


Figure 6: Control Panel mapping of the lab

Lane Detection Algorithm

Our lane detection algorithm captures two images from the cameras located on either side of the robot. The colors of each image are filter out so as to enhance the white lane markers' contrast and remove everything else from the image. The image is then converted to a binary image and simple noise removal is done. The results are seen in the figure below

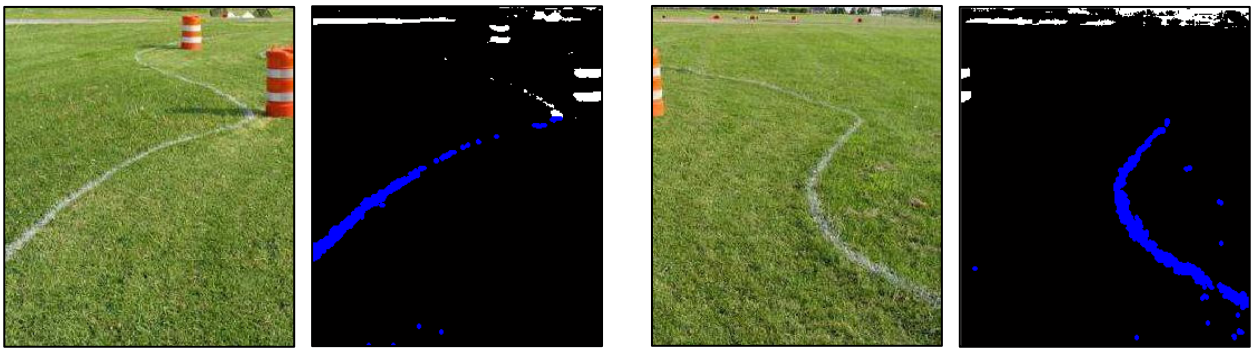


Figure 7: Line detection Left is the original image, Right is the image before transformed into binary image.

In each image, all the white points in the image are taken and line-fitting operation is done using least square method. A line is fit so that the sum of squares of all the deviation $\left(\sum \delta^2\right)$ is minimized.

Let the line to be fitted be $ax + b = 0$ and (x_1, y_1), (x_2, y_2)…. (x_n, y_n) be the white points in the image. Solving the following equations for *a* and *b* gives the equation for the fitted line.

$$\delta_i = y_i - (ax_i + b) \qquad (3)$$

$$\sum \delta^2 = \sum \{ \text{yi } (\text{axi} + \text{b}) \}^2 \qquad (4)$$

The standard deviation of all the white points is calculated and points that are 3 standard deviations away or more from the fitted line are eliminated. A new line is fitted with the remaining points and its slope is calculated.

A weight is determined using the number of white points in each image. This weight is used to create a weighted mean slope from the slopes obtained from both images. The position of the robot with reference to both lines is calculated by finding the midpoint of the intersection of both the left and right lines and the y-axis. This gives us the proper information to send to the mapping algorithm so that the lines can be modeled as obstacles. The resulting lines are shown in the figure below.

Figure 8: Resulting lines super-imposed on the original images

Path Planning

Our approach builds on general vector field theory. In this theory obstacles apply force on the robot that pushes the robot away from the obstacles. The sum of all the forces will dictate the direction the robot chooses. The force applied to the robot from a particular obstacle is proportional to the distance the robot is from the obstacle[5].

Vector field general theory

In the vector field concept (VFC)[1 3 4] the robot is considered to be in a force field where all the obstacles push the robot away and the target pulls the robot to it.

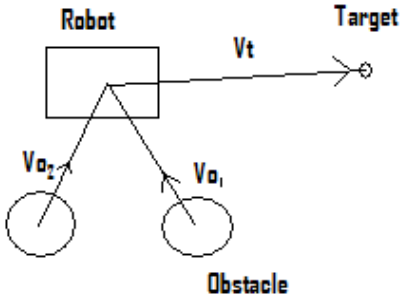


Figure 9: Robot with two obstacles and a target location

The resultant force acting on the robot is the sum of the repulsive force from the obstacles and the attractive force from the waypoint target as shown in Equation (5)

$$\vec{V}_p = \vec{V}_T + \frac{1}{n}\sum_{i=1}^{n}\vec{V}_{oi} \qquad (5)$$

where n is the number of obstacles in range and V_{oi} is the force exerted by them on the robot. V_T is the pulling force exerted by the target on the robot.

Note that the magnitude of the force exerted by the obstacle decreases with distance from the robot. The magnitude of the waypoint or target vector remains constant irrespective of the magnitude of force exerted by obstacles.

Modified Vector field Concept

The VFC uses just one vector to represent the obstacle. It is possible that obstacle might have a part sticking out of the main body. This may become a potential

hazard for the robot. If multiple vectors were considered originating from the visible surface of the obstacle the robot would know about the protruding part.

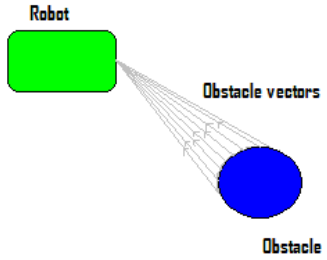


Figure 10: Multiple obstacle vectors covering the entire visible area

This enables the robot to pass very close to the obstacle and through narrow passage ways.

The magnitude of the obstacle vectors is determined by Gaussian distribution shown in Equation (6)

$$\left|\vec{V}_o\right| = ke^{\frac{-(x-\mu)^2}{2\sigma^2}} \qquad (6)$$

The resultant of all obstacle vectors forms the final obstacle vector.

Simulation using Player/Stage:

The Player is a robot device interface and Stage is a multiple robot simulator. The Stage supports various sensor models such as laser scanners, PTZ cameras etc., which are simulated using control programs with great accuracy. An image file of the environment is created and a user defined control program acts as client to the player server. The communication between the server and the client is through a TCP socket .This approach was used to test various algorithms and control programs developed for the robot which later were implemented directly or with very few changes in the physical robot. The control programs were developed in Java using the JavaClient2 libraries.

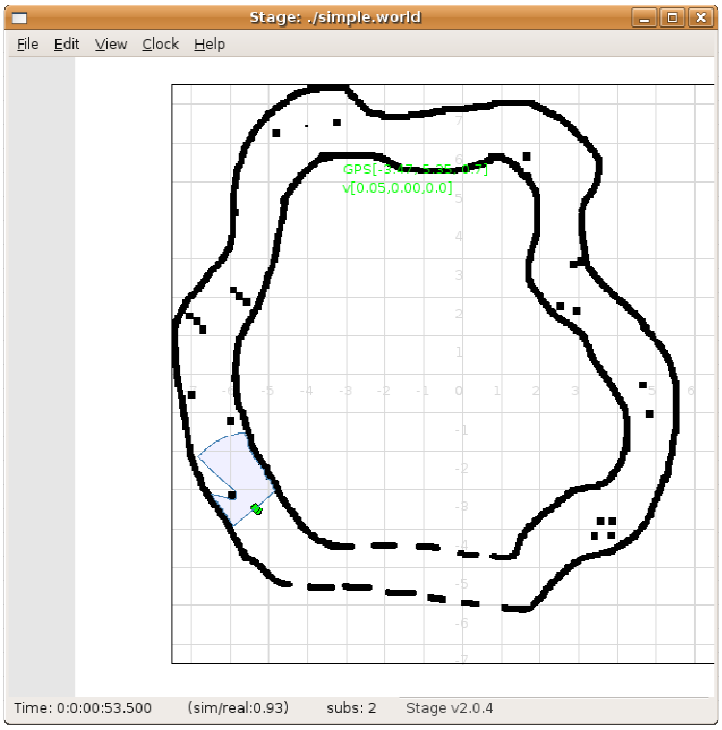


Figure 10: Player/Stage simulation

Conclusions

The Bearcat Cub continues to evolve each year into a more robust research vehicle. This year every module of the robot was tested constantly for durability and predictable behavior before their integration into one system. Various control algorithms were developed during the development stages which evolved with repeated testing in simulated and real world environment. The new mapping algorithm and the new vision system are expected to significantly improve the Cub's performance this year.

References

[1] J.C. Wolf, P. Robinson and J.M. Davies "Vector Field Path Planning and Control of an Autonomous Robot in a Dynamic Environment," FIRA Robot World Congress. 2004.

[2] A. Guttman, "R-Trees: A Dynamic Index Structure for Spatial Searching", Proc. 1984 ACM SIGMOD International Conference on Management of Data

[3] I. Ulrich, and J. Borenstein, "VFH+: Reliable Obstacle Avoidance for Fast Mobile Robots," IEEE Int. Conf. on Robotics and Automation, May 1998, pp. 1572-1577.

[4] I. Ulrich, and J. Borenstein," VFH*: Local Obstacle Avoidance with Look-Ahead Verification." IEEE Int. Conf. on Robotics and Automation, April 2000, pp. 2505-2511.

[5] R. Siegward, I.R. Nourbakhsh "Introduction to Autonomous Mobile Robots," MIT Press, Cambridge, Massachusetts, London, England, 2004, pp. 267-272.

Team Bearcat Cub for Intelligent Ground Vehicle Competition 2008 (Appendix A)

1	Brian Hallez	Electrical Engineering
2	Joshua Rajasingh	Mechanical Engineering
3	Kovid Mathur	Mechanical Engineering
4	Mark Aull	Aerospace Engineering
5	Mark McCrate	Mechanical Engineering
6	Sachin Raviram	Mechanical Engineering
7	Saurabh Sarkar	Industrial Engineering
8	Scott Reynolds	Computer Science
9	Srinivas Tennety	Mechanical Engineering
10	Timothy Wagner	Mechanical Engineering

Bill of Materials (Appendix B)

Part	Manufacturer	Model No	Quantity	Unit Price	Total
Frame	80/20 Inc.	Custom design	1	950	950
Generator	Honda	EU 2000i	1	778	778
Motors	Pacific scientific	PMA43R-00112-00	2	970	1,940
Amplifiers	Copley Controls Corp.	Xenus Servo Drives XSL-230-36	2	540	1,080
Drive Wheels	Segway	Enhanced Traction	2	188	376
Gearboxes	Segway	HT design, 25:1 gear ratio	2	488	976
Laptop	Dell	D830	1	1,181	1,181
Cameras	Sony	PV-DV51	2	290	540
E-stop	Futaba	FRF-0302U	1	321	321
Motion controller	Galil Inc.	DMC-2130 Ethernet	1	2,800	2,800
Inverter	Whistler	800 W	1	52	52
GPS	Novatel	ProPak-V3-HP	1	3,252	3,252
Miscellaneous				300	300
			Total		$14,546