The 17th Annual Intelligent Ground Vehicle Competition
June 5 - June 8, 2009

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 17th consecutive year that the University of Cincinnati Robotics Team has participated in the IGVC. This year's robot, which is a compact version of our 2007 platform, has been developed by a collaborative effort from a multidisciplinary team. It has undergone major changes in its software along with changes in the support structure from the previous year's entry into the competition. New hardware has been added to make it more predictable and reliable. This report describes the various aspects of Bearcat Cub's design, design tradeoff considerations and improvements over the past IGVC entries by the UC Robotics team.

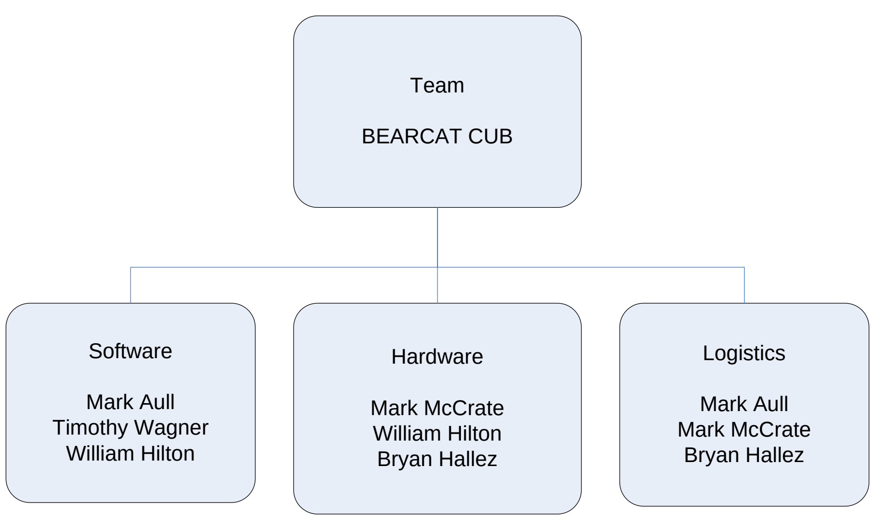
Design Innovations

The Bearcat Cub this year went through major revisions not only in the software design but also in its frame work which is drastically smaller than its predecessors. It features 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 improved and is more efficient. Our vision system is now more robust using two cameras to detect the lanes instead of a single 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 of eight graduate as well as three undergraduate students under the supervision of advisor Dr Ernest L. Hall.

Figure 1: Team Organization



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

Frame

The load bearing chassis of the bearcat cub is made of 80/20 aluminum extrusion which is light and can be used without compromising the strength of the frame structure. The advantage of using this modular type frame comes in the ease of reshaping, and the ability to quickly mount new components as they are brought into the design.

The aft shelving support uses aluminum window shade track which weighs less than $1/6^{th}$ the equivalent length of 80/20. Despite this reduction in weight, the shelving is still able to support at least 135lbs of distributed load. The profile of this material also enabled us to use the top shelf as a convertible desk.

Compacting The Bearcat Cub

Over its history, the Bearcat Cub has undergone incremental improvements in design from the first generation golf cart, the second and third generation cubes, to the fourth generation robot. However our newest model, the Bearcat Cub Micro is perhaps the most extensive revision yet due to its significantly smaller size.

The frame of the Bearcat Cub IV was stripped completely and cut to size around our battery dimensions. Planning for this involved using a SolidWorks® CAD model that proved the feasibility of overlapping our 2hp brushless servo motors and using every bit of available space. During construction, numerous additional improvements were made particularly through tight wiring and unconventional placement of power electronics and motion control boards.

While most of the frame is based on 80/20 aluminum extrusion, the aft shelving uses aluminum window shade track which weighs less than $1/6^{th}$ the equivalent length of 80/20. Despite this reduction in weight, the shelving is still able to support 135lbs. The profile of this material also enabled us to use the top shelf as a sliding retractable desk.

Bearcat Cub IV		Bearcat Cub Micro	
Pros	Cons	Pros	Cons
-Ample space -Easy to access -Instant energy replenishment	-Easily broken caster wheels -Weighs 300 lbs -Too large -Disorganized -Difficult to transport -Gasoline fueled -Vibrations, loud	-Half the volume -Weighs 180lbs -Easy to remove sensor tower -Well utilized space -Organized, more presentable -Sturdier design -15 hr mean run time before recharge -Quiet operation -Top slides to convert into a desk	-Hard to access -Limited run time -11 hr recharge

Figure 3: Bearcat Cub design comparison

Drivetrain

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 10" rear castor wheel provides the 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 PMA43R0011200, 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 gearbox 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 but for safety reasons the speed has been limited to 2 mph. The robot successfully climbed a 30 degree incline ramp.

Power System

The robot is powered by two 12v deep cycle marine lead acid batteries connected in parallel for total energy storage of 2064Wh. Power from the battery is sent to an 800W 120v inverter which powers all electronics including the motor amplifiers. When idle, the battery can run all electronics for up to 15 hours. When driving, the robot can run for 6 hours. Using batteries allows for silent, vibration and smoke free operation compared to a compact generator.

Emergency Stop

The robot stops using electronic dynamic braking that dissipates heat through a resistive load shunt. A manual E-stop button is located on the rear of the vehicle more than 2 feet above the ground which activates the brakes. A Futaba remote control radio E-stop can also apply the brakes from a distance of 65 feet.

As part of our emphasis on having a compact robot, the e-stop circuit has been totally redesigned. Figure 4 shows a comparison between the new and old e-stop.

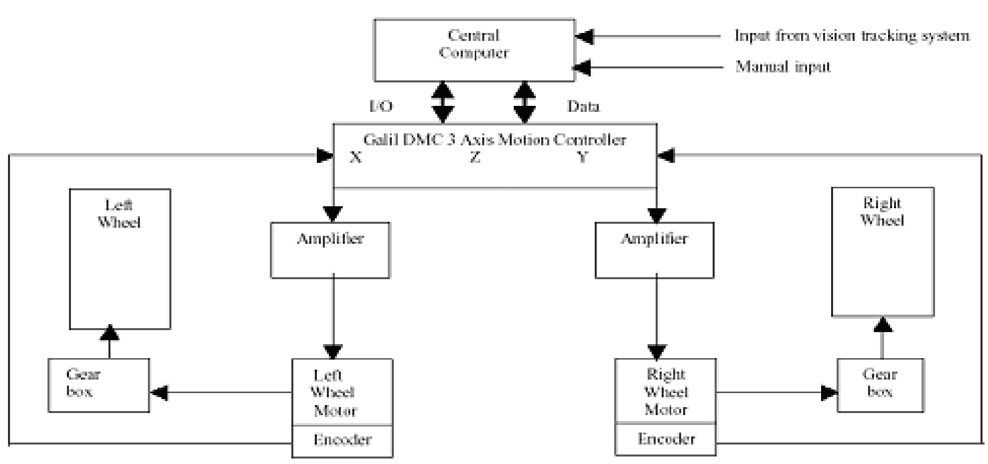
Old E-Stop		New E-Stop	
Pros	Cons	Pros	Cons
-Fully redundant circuit -One wire in, one wire out.	-Mechanical relay based, components may fail unexpectedly. -Weighs 20 lbs. -Very large: 10.25 x 12 x 5.5 inches	-CFCI based shut off, industrial grade. -Weighs less than 3 lbs. -86% reduction in volume.	-No redundancy. -Five wires exposed, mostly open. -Harder to reset.

Figure 4: E-stop comparison

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 5.

Figure 5: 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 6 below shows the general electronics layout.

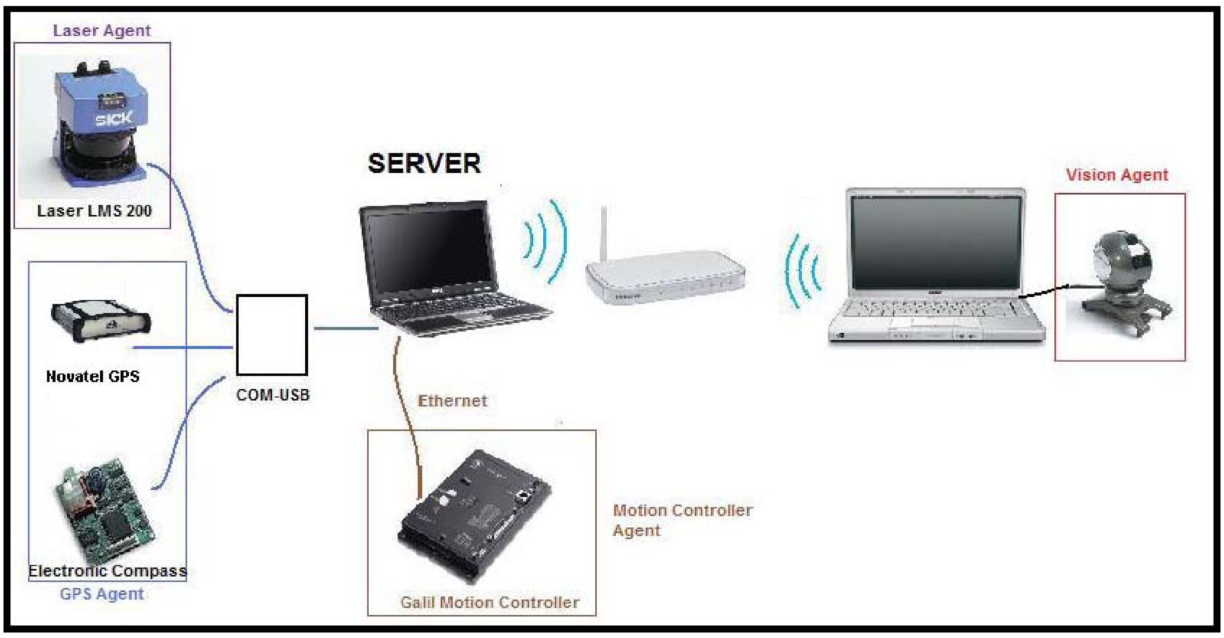


Figure 6: Bearcat Cub block diagram

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 communicating 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 using a USB communication. The accuracy achieved with this unit is 0.6m using SBAS channel.

Cameras

Two Sony DCR-TRV118 video cameras provide the images that are used by the line detection system. Wide angle lenses and built in image stabilization improve image quality.

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.

4. 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.

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

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

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 nonleaf 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.

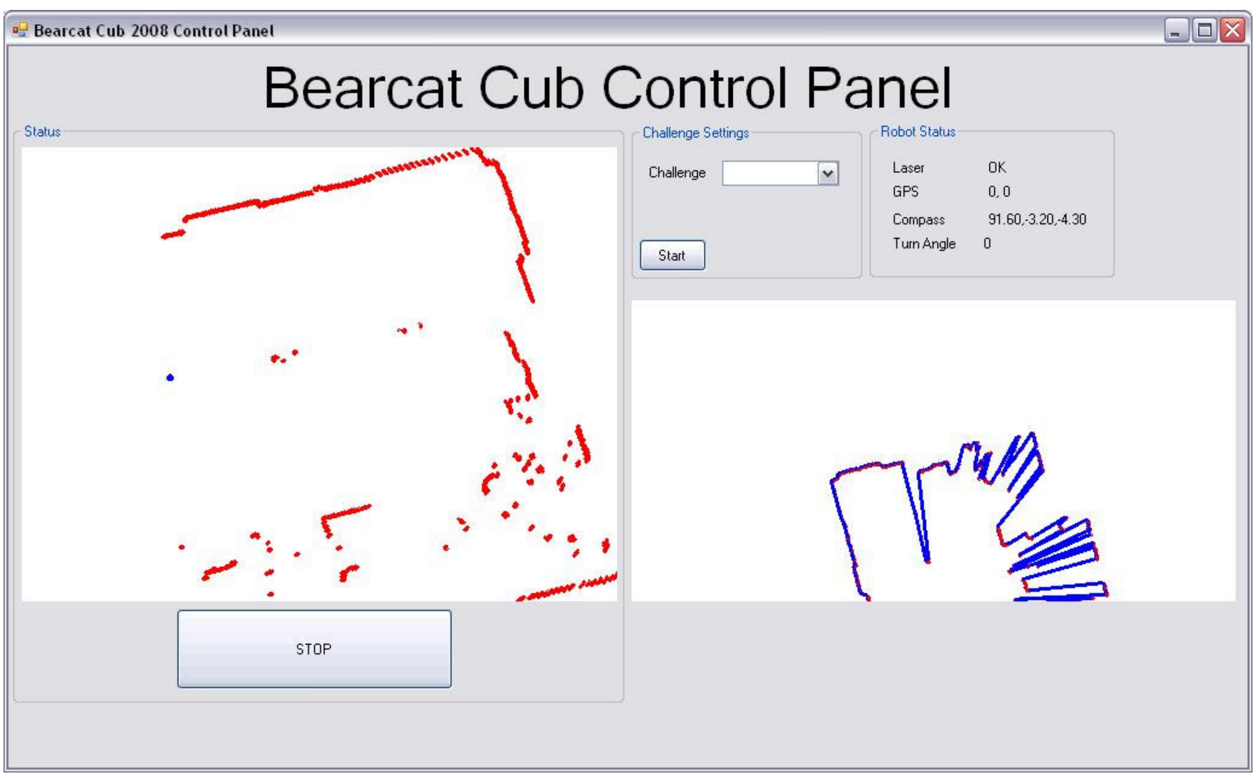


Figure 7: Control Panel mapping of the lab

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 figures below.

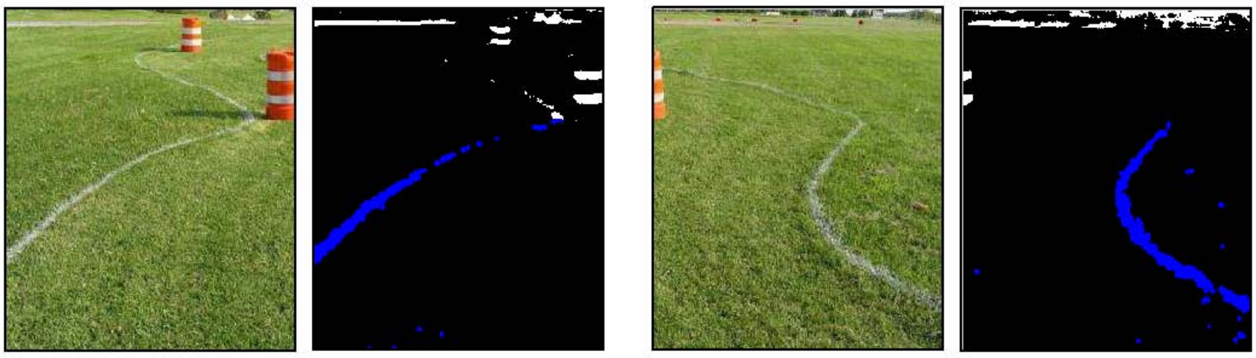


Figure 8: Line detection Left is the original image, Right is the image before transformed into binary image.

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

Let the line to be fitted be ax + b = 0 and (x1, y1), (x2, y2)…. (xn, yn) be the white points in the image. Solving the following equations for a and b gives the equation for the fitted line.

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

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

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

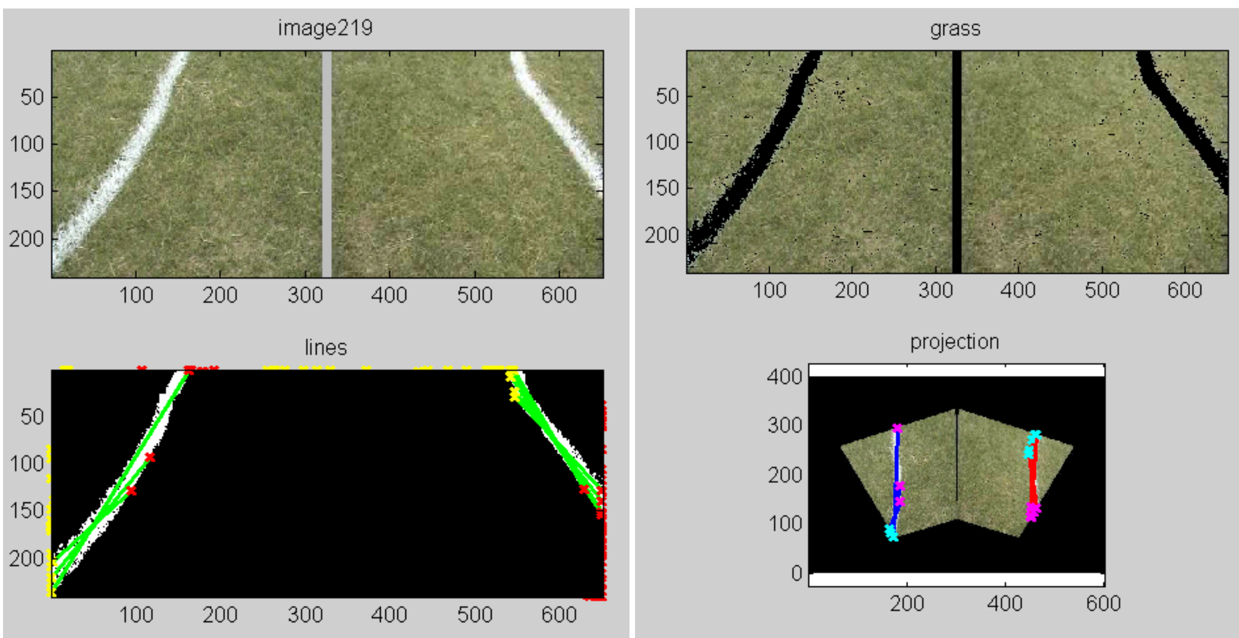


Figure 9: Resulting lines super-imposed on the original images

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.

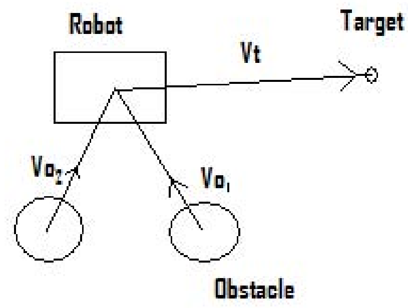


Figure 10: Robot with two obstacles and a target location

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).

$$\overrightarrow{V_p} = \overrightarrow{V_r} + \frac{1}{n}\sum_{i=1}^{n}\overrightarrow{V_{0i}} \qquad (5)$$

where n is the number of obstacles in range and Voi is the force exerted by them on the robot. VT 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.

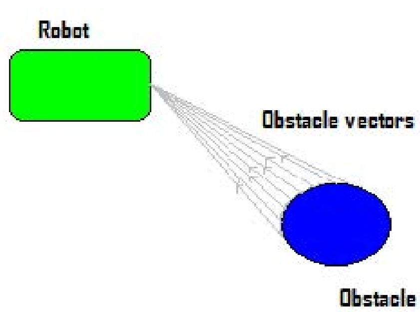


Figure 11: Multiple obstacle vectors covering the entire visible area

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)

$$\left|\overrightarrow{V_0}\right| = ke^{\frac{-(x-\mu)^2}{2\sigma^2}} \quad (6)$$

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.

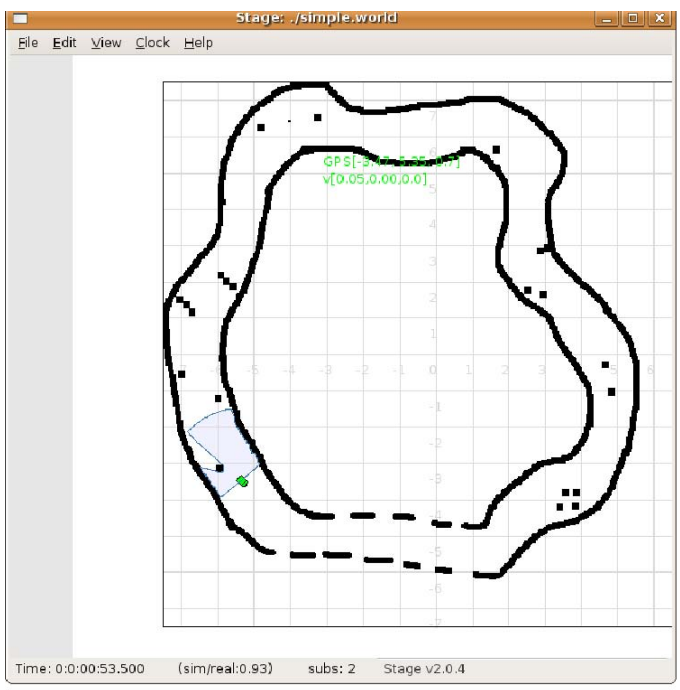


Figure 12: Player/Stage simulation

Navigation

Navigation is accomplished by using a Kalman filter to integrate data from a digital compass and one or more GPS devices. This data determines a heading, subject to obstacle avoidance. After the position estimate is within a critical radius of the target waypoint, the robot will spiral out to a variable radius, making the system more robust to GPS errors. The heading computed between the estimated and target positions is then modified by the obstacle avoidance algorithm.

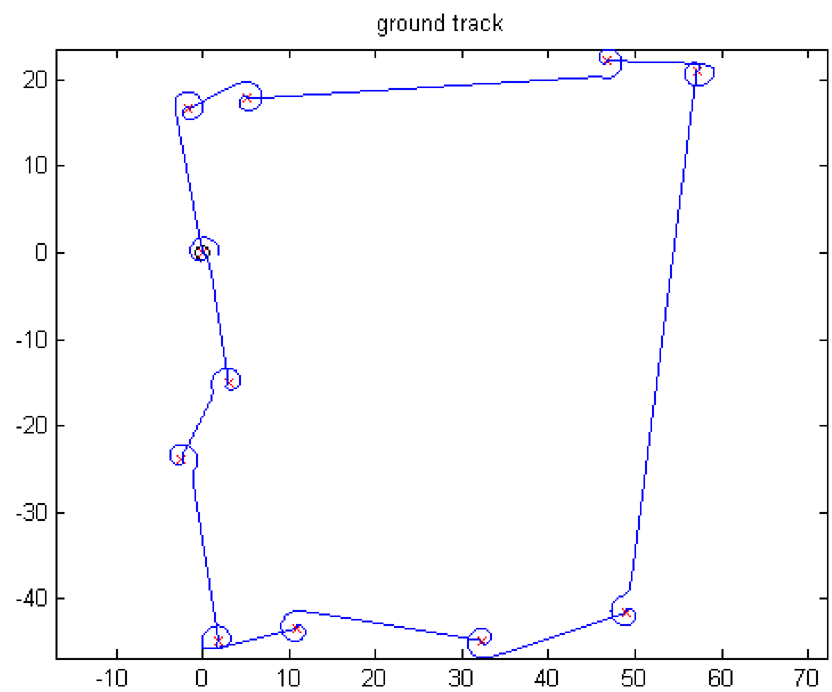


Figure 13: Navigation simulation showing spirals around waypoints

Obstacle Avoidance

The obstacle avoidance is a force vector field variant. It takes its primary inputs from the laser scanner, but is augmented with information from the cameras and stereovision system. This means that lines can be considered as obstacles to be avoided, and obstacles that are transparent to the laser (either too high, like tables or mesh-like, like fences or bushed).

JAUS

Due to the JAUS protocol being rewritten, we have scrapped the JAUS library we wrote for previous competitions, and will now try to use the openJAUS library. Once the mission referred to in the rules ("Upon receipt of the “Resume” command by the COP, the team’s entry will commence the execution of the mission prescribed by the competition organizers") IS "prescribed by the competition organizers", we will have some idea of whether we will be able to\attempt to complete it.

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, "RTrees: 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. 15721577.

[4] I. Ulrich, and J. Borenstein," VFH*: Local Obstacle Avoidance with LookAhead Verification." IEEE Int. Conf. on Robotics and Automation, April 2000, pp. 25052511.

[5] R. Siegward, I.R. Nourbakhsh "Introduction to Autonomous Mobile Robots," MIT Press, Cambridge, Massachusetts, London, England, 2004, pp. 267272.

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

1	Mark Aull	Aerospace Engineering
2	Mark McCrate	Mechanical Engineering
3	Srinivas Tennety	Mechanical Engineering
4	William Hilton	Mechanical Engineering
5	Bryan Hallez	Mechanical Engineering
6	Timothy Wagner	Mechanical Engineering
7	Joshua Rajasingh	Mechanical Engineering
8	Kovid Mathur	Mechanical Engineering
9	Anup Deshpande	Mechanical Engineering
10	Sohan Mahajan	Mechanical Engineering
11	Deepthi Sharan Thatiparti	Mechanical Engineering
12	Saurabh Sarkar	Industrial Engineering

Bill of Materials (Appendix B)

Part	Manufacturer	Model No	Quantity	Unit Price	Total
Frame	80/20 Inc.	Custom design	1	950	950
Batteries	Rocket	DC31DT	2	85	190
Motors	Pacific scientific	PMA43R00112-00	2	970	1,940
Amplifiers	Copley Controls Corp.	Xenus Servo Drives XSL230-36	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	PVDV51	2	290	540
Estop	Futaba	FRF0302U	1	321	321
Motion controller	Galil Inc.	DMC2130 Ethernet	1	2,800	2,800
Inverter	Whistler	800 W	1	52	52
GPS	Novatel	ProPakV3HP	1	3,252	3,252
Miscellaneous				300	300
			Total		$13,968