17th Annual Intelligent Ground Vehicle Competition

Georgia Institute of Technology – RoboJackets

Design Report: Candiii

June 4-8, 2009

Introduction

The RoboJackets is a group of Georgia Tech students, faculty, and alumni that aims to enhance the understanding of the field of robotics and its applications in depth of knowledge as well as to increase of the number of students exposed to it. In accordance with this goal, a team within RoboJackets has participated in the IGVC for the past several years. For the past two years, we have used the same mechanical base (called Candi), but each year we have redesigned most of the electronics and software. Since this is the third year using the same chassis, we have kept our past naming convention and added an extra i to the robot's name. We believe that Candiii is the best designed and implemented robot we have submitted so far.

Design Procedure

Last year, the RoboJackets IGVC team took a structured approach to management. All of the returning members lead a sub-team of their own choosing. There were sub-teams for all of the important systems on the robot. For example, last year's team leader was also the sub-team leader of the power and motor drive system. Sub-team leaders were meant to be responsible for everything relevant to their system and to work with new members who were interested in that system. The problem was that it put too much burden on sub-team leaders to find new members to help them with the work. Prospective members of RoboJackets often have little experience in their field and do not know exactly what they want to do. The veteran members of the club have a lot of experience, but they do not usually have the time to thoroughly teach each and every person who is interested in their work. New members often didn't know which sub-team to join or what they would do in that sub-team.

This year, instead of assigning team members large areas of responsibility, we assigned smaller tasks to each person. For example, one of the tasks was to create a software interface for the arduino microcontrollers on the robot. When the member assigned that task was finished after a few months, they picked a new task. This methodology allow the team members to focus on their work and put the job of resource management to the team leader. When new members came to our team, they could talk with the members about their work to see what they wanted to do, but it was the team leader's responsibility to put them in a place that fit their skills, and to manage the workload of all of the members.

The team met twice a week in our shop in the Tin Building. As with all the previous years we have done the competition, we had a one hour planning meeting during the week and a five hour work meeting during the weekend. The planning meeting is used to prioritize the tasks need to be done during the work meeting. We also used that time to report to each other of our work progress and to re-analyse the design whenever needed. The work meeting is intended to set aside time were members can finish their work or get help from others. Most members also worked on their own outside of each meeting.

This year the members were:

Paul Varnell – 3rd year Electrical Engineering – Team Leader and Electrical Lead

Tasks: motor interface board, IMU board, general electrical

Chris McClanahan – 4th year Computer Science – Software Lead

Tasks: vision, mapping, path planning

Stefan Posey – 4th year Aerospace Engineering – Mechanical Lead

Tasks: camera mount, sonar mount, general mechanical

Jean-Pierre de la Croix – 4th year Computer Science and Electrical Engineering

Tasks: mapping, path planning

Paul Foster – 3rd year Mechanical Engineering

Tasks: vision, sonar

Jacob Schloss – 2nd year Aerospace Engineering

Tasks: arduino interface, IMU

Curtis Mulady – 1st year Electrical Engineering

Tasks: sonar

Mechanical Design

The mechanical platform build two years ago has proven to be one of the most successful parts of our previous designs. The chassis has performed very well and has all of the features that are important for this competition including manoeuvrability, durability, and ease of control. Given this, we decided to reuse this platform with minimal changes.

The custom frame was fabricated from thin wall one inch square steel tubing, tapering in the rear to allow for easier turning. Each piece was cut and welded from six foot lengths of stock. Aluminum bars were used to mount the drive motors.

The base was designed as a front-driven differential drive system with a rear ball caster. This system has been preferred for its easy implementation and controllability. The two motors used are NPC Robotics 1.7 hp DC motors that were donated from the RoboJackets BattleBots team. These motors power the front two drive wheels, moving the entire vehicle. The traditional system of two swivel casters supporting the rear of the vehicle was avoided for several reasons:

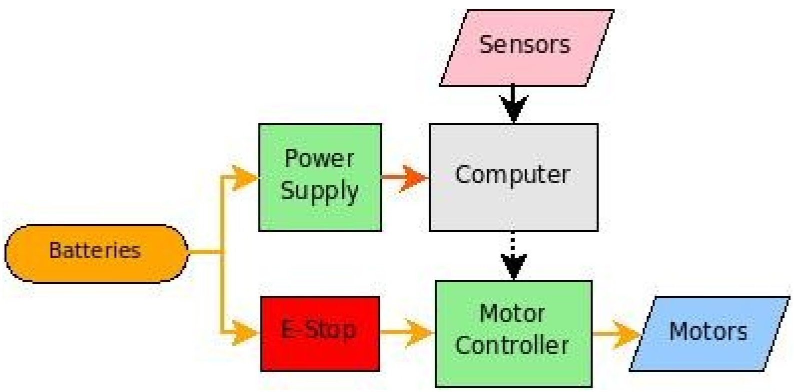
1. Swivel casters, from past experience, have introduced significant amounts of wobble into the rear end of the vehicle while traversing through soft grass, making control and vision more difficult.

2. Zero radius turning with traditional casters induces slight forward or backward translation to the entire chassis as the wheel swivels around its caster angle axis, introducing uncompensated motion to the controllers.

3. To maintain a ground clearance of approximately 6-7 inches, casters of only 3-4” diameter would be required, much smaller than those desired for traversing grassy terrain. The mechanical team decided to abolish the standard practice of utilizing swivel casters and implement one single 12” ball caster, providing one of Candii's largest and most effective innovations. This omni-directional ball caster solves the wobble issues, and can instantaneously compensate for changes in direction due to its zero degree caster angle. The 12” diameter ball will also leave a wider footprint in soft grass. The stainless steel ball is held in place by spring steel “net” of five small ball casters and four small roller bearings around the equator.

Since the main vision system camera was desired to be mounted as high as possible for the optimum field of view, chassis stability was placed as a top priority. To achieve stability, an independent suspension system was implemented on the front drive wheels. The inverted cantilever style suspension keeps the drive wheels behind the main pivot, effectively shortening the wheelbase, allowing for tighter turning. Suspension is provided through the implementation of two single acting pneumatic cylinders with coilover springs. Small valves were installed at each cylinder port and adjusted to introduce damping. Since the 5' camera post is mounted to the rear of the vehicle, the main rear ball caster base-plate was also suspended by eight small springs to allow for shock absorption.

Electrical Design



Power

The robot is powered by two 12 V lead acid batteries connected in series. The batteries are connected directly to the motor system through the E-stop. and are connected to the digital electronics through the power supply which powers all of the other electronics. The wires going to the motors and the power supply are connected close to the batteries to prevent noise from the motors from coupling into any of the sensitive electronics. The interface between the computer and motor controller is isolated using an opto-coupler to avoid ground loops and to further reduce noise from the motors. The power supply was custom designed to provided regulated 12 V, -12 V, 5 V, 11 V, and 19 V power to the electronics. All systems were carefully specified for their maximum current ratings and were properly fused.

Emergency Stop

The emergency stop (estop) allows the robot to be stoped immediately from either a large red button on the robot or a wireless button. The estop uses an industrial relay in series with the motors to instantly remove power from the robot. The relay is configured so that self-latching. Holding the green button on the back of the robot activates the relay's coil so power can pass to motors. If the stop button is pulled out and the wireless receiver is not active, the power passing through the relay will keep the coil activated. If the red button is pressed in or the wireless receiver is active, the relay will switch off and stop powering the motors. This simple design is very reliable and stable.

Arduino

An arduino is an open source microcontroller board that is based around Atmel AVR microcontrollers. There are many different versions of these boards, but all of them are programmable using an open source compiler and IDE and many of them have the same header configuration. These boards are ideal for new members because they are relatively easy to use and they have a lot of community support. They are also good for experienced members because they are programmed in C, most members' of RoboJackets language of choice, and are powerful enough to do useful things.

All of the sensors that could not directly connect to the computer are connected to an arduino. If a board need extra hardware, such as the IMU and the motor interface board, we custom designed arduino shields containing those parts. Shields are just boards that fit over the top of the arduino and plug into its headers.

Motion Control

Each motor is controlled using an Open Source Motor Controller (OSMC). The OSMC is a H-Bridge power amplifier released under the GNU Public Licence that is designed for use in abusive environments such as BattleBots. The OSMC varies the speed of each motor according to the duty cycle of the pulse width modulated (PWM) input. That PWM signal is generated by the isolated motor interface arduino board which sends and receives data from the computer. The motor interface board also has connections for the motor encoders and motor current sensors, which are used to provide control feedback. The board has a wired analog joystick connector that is used to drive the robot when a computer is not connected.

Software on the motor interface board allows the motors to be controlled directly by changing the PWM duty cycle or by setting the speed which is maintained using an controller that uses feedback from the motor encoders and motor current sensors. The software also limits the maximum wheel speed to stay under 5 mph.

Inertial Measurement Unit (IMU)

The IMU board is a arduino shield that holds several sensors used to detect the motion of the robot. It uses 3-axis accelerometers, gyroscopes, and magnetometers along with the motor sensors and GPS to track the robot's position. Software on the arduino combines the sensor data using a Kalman filter and sends this back to the computer. This board has been mostly designed, but not manufactured or tested. Hopefully it will be ready for the competition.

Software Design

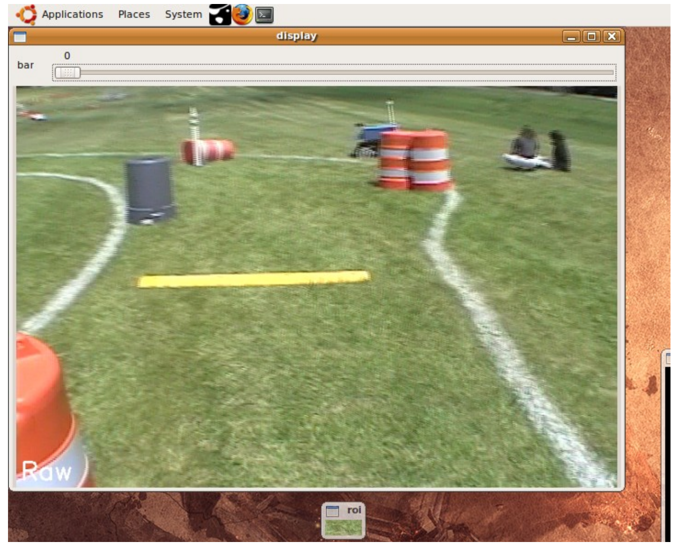
Arduino Interface

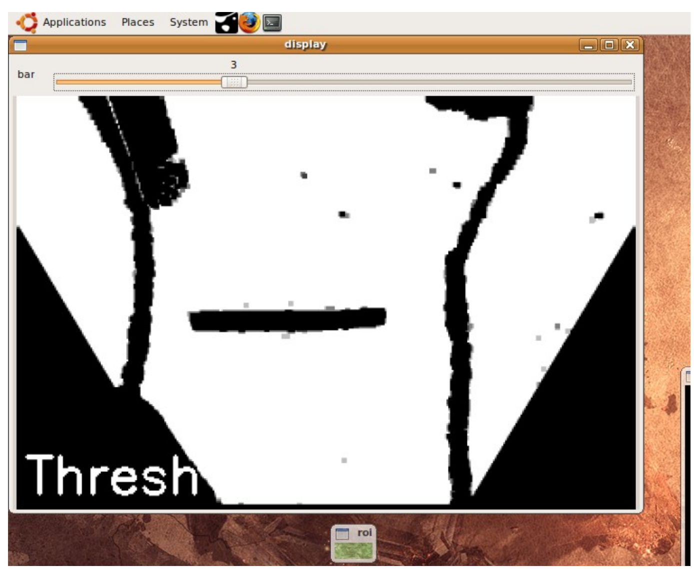
Since we are using more than one arduino on the robot, it became clear that we needed to have a standard interface for communication. We implemented a packet-based system that can handle data loss and retransmission and can be used for both push and pull architectures. The interface also provides usage guidelines for interacting with the code using the interface. It is also generic enough to support different communication links, such as Ethernet and CAN, if we choose to use those in the future.

Vision and Mapping

Our robot last year used a combination of reactive and graph-based planning acting on a single camera frame to determine the direction to drive the robot. The major problem with that method was that it could only avoid obstacles it could currently see, and not any of the obstacles that it passed. As we passed obstacles and they left our field of view, we would often turn into them. There are two ways to solve this: increase the field of view of the robot to cover the sides or have the robot retain some memory of the obstacles it has seen. We used the first solution partially in buying a wider angle lens and equipping the robot with sonars, but the problem of turning into passed barrels still remained. We decided to try using the second solution by creating a map of the course and obstacles in real time on the robot. Our implementation of this idea uses the motion data from the camera and from other sensors to draw the local visual data into a global occupancy map. The map slowly erases the data as time passes to prevent large (loop closure) errors from accumulating and to conform to the rule that the course can not be mapped before each run.

The first step in the vision system is to convert each camera frame into information that can be easily used by the mapping software. The camera is mounted on the top of the robot looking down on the course at an angle. From the camera's perspective, closer objects are larger and far objects are smaller. The images are normalized using a perspective transform that changes the image so that it appears to be taken from directly overhead, where objects are of uniform size at all distances. The transformation assumes that everything on the course is on a single plane, which causes distortion of the barrels which is taken care of by the mapping software. This step is computed on the graphics card using OpenGL, an open source graphics library. At the same time the perspective transform is applied, other transforms are applied to correct for lens distortion.

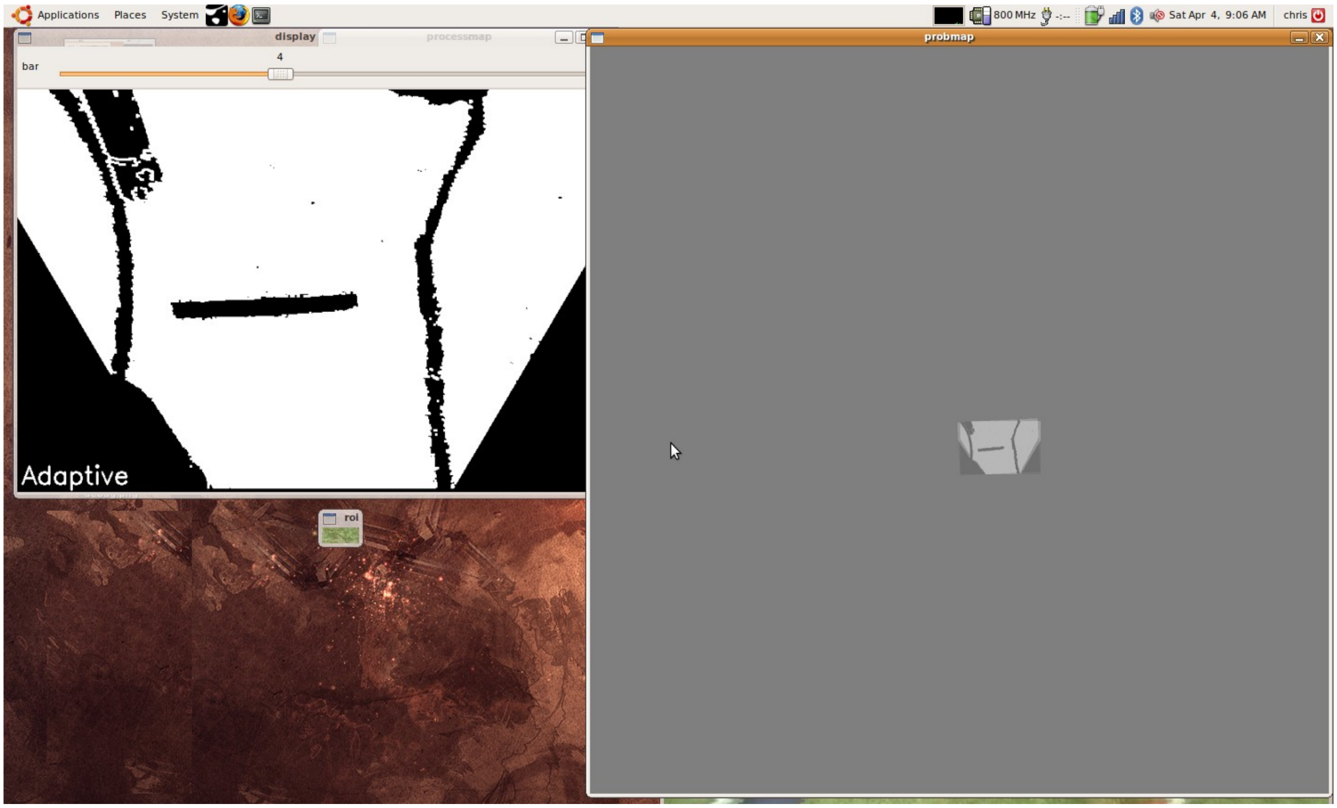




The transformed image is then thresholded based on color. Colors that are similar to grass are marked white and everything else is marked black. To account for various light conditions, the color for grass is found by averaging a small part of the image close to the robot which should contain grass. The grass color is also averaged over time to prevent fluctuations from things like dead grass. There is also a check to ensure that the grass color is somewhat similar to grass; if it is not, that frame is not included in the average.

The map is created using both the transformed image and the thresholded image. The transformed image is first converted to gray scale and used for feature matching. Feature matching finds easily identifiable points in one image and then tries to find the corresponding points in another image that has been translated and rotated. We used the feature tracking software in OpenCV, another open source computer vision library. Using the matched features, we then find the most likely homography matrix, translation and rotation, between two images. The matched features often included several incorrectly matched points, so we used RANSAC, a nonlinear filter that is good at handling outliers, to choose the best homography matrix.

Using the motion data from the camera (homography matrix) and from other sources such as GPS and motor encoders, camera frames are drawn into the world map. The world map is represented by a gray scale image where white (255) represents traversable area, black (0) represents non-traversable area, and grey (127) represents unknown areas. The thresholded image is added to the world map at the location of the robot. One is added to the world map at points were the thresholded image is white and one is subtracted for black points. Since the frame rate is fairly high, the map builds up quickly as the robot moves.



Every few frames, each pixel is moved by one towards grey so areas that have not been seen for a while fade back to unknown areas.

Path Planning

The path planning system uses a simple potential fields algorithm, where the robot is attracted to the goal but repelled from obstacles in the map. For the Autonomous Challenge, the goal position is chosen to be the farthest reachable point in front of the robot. For the Navigation challenge the goal is one of the GPS waypoints. The order of travel of the waypoints is picked manually beforehand and the robot goes to the next waypoint in order after it has picked the current waypoint.

Cost

Item	Retail Cost
Steel	$200
Aluminum	$200
Mechanical Accessories	$300
Motors	$0 (Donated)
Motor Drivers	$0 (Donated)
Custom Power Supply	$400
Garage Door Opener (estop)	$80
Laptop	$0 (Donated)
Camera	$580
Camera Lens	$140
GPS	$200
Arduino Boards	$100
IMU	$300
Motor Interface Board	$150
Sonars	$0 (Donated)
Total	**$2,650**