Rose-Hulman Autonomous Terrain Traverser

Michael Auchter, Jay Kinzie, Jon Klein, Tom Most, Andy Spencer
{auchtemm,kinziejh,kleinjt,mosttw,spenceal}@rose-hulman.edu

Robotics Team, CM 5000
Rose-Hulman Institute of Technology
5500 Wabash Avenue
Terre Haute, IN 47803-3999

May 15, 2008

This report and the described vehicle were designed and constructed by the Rose-Hulman Robotics Team are the work done during the 2007-2008 school year is of equivalent caliber to that which would be given credit in a Senior Design course.

_____
Wayne T. Padgett

1 Introduction

The RHIT Robotics Team is comprised of undergraduate students from Rose-Hulman Institute of Technology in Terre Haute, Indiana. We have a long history with the International Aerial Robotics Competition, having built autonomous helicopters for the last fifteen years. However, due to a recent shift in interests it was decided that a ground based competition would be more appropriate for the team's main project and the Intelligent Ground Vehicle Competition was selected as a primary competition.

Due to the sharp contrast between the two projects and the current state of our software at the beginning of the 2007 school year it was decided that, while some existing sensors could be reused, the the rest of the ground vehicle would need to be built from scratch. This paper will discuss the various challenges, opportunities and methods used to design and construct a completely new vehicle, including new electronics and software.

2 Design Process

The Rose-Hulman academic year is divided into Summer, Fall, Winter, and Spring quarters. The decision to switch projects and the selection of a new competition was completed near the end of the Summer Quarter. The remaining three academic quarters were used for planning, construction, and integration with testing preformed during both construction and integration.

2.1 Design Decisions

Use of Free/Open source components

Based on our previous experience using proprietary protocols and software we made the decision to use Free and Open Source software and hardware wherever possible. The software for the robot consist of entirely free software and is released under the GNU GPLv3. Several electrical components, such as our primary camera, also have the schematics and software available under the GPL.

Modular and expandable design

Our entry this year is intended largely as a platform for further growth, particularly in the software arena. Thus our design is composed of generic components designed to be reusable in future years. For example, our motor controllers are designed to handle currents well in excess of the maximum required by our motors to assure the possibility of reuse given hardware changes.

Off the shelf components

When possible we have attempted to use off the shelf components for our vehicle. However, Rose-Hulman's primary focus is on education and not research. This focus on education has led to several components being designed from scratch in order to further the education of the participating team members. Such components include the control software and sensor drivers, the vehicle chassis, camera mount, and motor controller boards.

2.2 Administrative Facilities

2.2.1 Lab space

Lab space and equipment has been provided by the Rose-Hulman Student Government Association. At the end of the winter quarter the club moved lab spaces.

2.2.2 Computing facilities

The team provides computing facilities for club members to use while working on the project. Access is set up using a LDAP/Kerberos server so that users can log into any services using the same account.

Wiki
: The main use of the wiki is for documentation and communication within the team. It also serves as a starting point when new members wish to join the club.

Subversion
: Primarily used by the software team, subversion serves as a source code management system. We chose to use SVN due to the ease of use and familiarity incoming club members have with it.

File storage
: A DFS server currently provides storage space for team files. This has been used for some documentation but is mostly being replaced by the Wiki. Archives and old files are also stored here.

Workstations
: There are two workstations provided for general use in the lab. One runs Ubuntu Linux and the other runs Windows Server 2003.

Mailing list
: A mailing list is provided by Rose-Hulman Institute of Technology and is used to send meeting agendas and reminders.

Jabber and IRC
: During the summer Jabber and IRC are often used for team communications. IRC has also been used with increasing frequency this year for team discussions between official meetings.

3 Hardware Platform

The chassis was constructed from 1 inch T-slot 80/20 because of its ease of use as a construction material. It allows us to build a sturdy platform, and modify it as the design evolves.

The chassis consists of two tiers of 80/20 that measure 37” long and 32” wide. On the bottom tier, two crosspieces run from front to back, providing a 10” gap on each side to mount wheels and drive train. Crosspieces near the front support the motors, motor controllers, and bearing mounts. Six vertical supports attach the upper tier, which consists of a frame of 80/20 covered by 1/4 in. plastic.

Figure 1: Early chassis, showing the basic structure of the robot

3.1 Wheels

Wheel Criteria

The wheels must support the 200 lb. robot and provide traction on a variety of surfaces, including grass, sand, and pavement. They must also be able to slip sideways since the robot uses tank steering. Pneumatic tires were chosen to provide a smoother ride over rough terrain. The diameter needed to between 9" and 12" to support the robot and provide ground clearance while still fitting within the frame. We also wanted a hub that could easily be attached to a sprocket.

Wheel Chosen

Manufacturer: TD Industrial
Product: 10" Pneumatic Tires / Wheels, 5/8" ID (2 pc Set)

Technical details

- Set of 2 - TD Industrial 10 in. pneumatic tires / wheels
- Great tires for Hand Trucks, Wagons and Carts
- 10 in. diameter, 4 in. width
- 5/8 in. ID ball bearing hub, rated for 300 lb. loads
- Tube: nylon, rating: 2 Ply, Size: 4.10 / 3.50 - 4

Product Description

Knobby all terrain tread. 30 PSI inflation pressure. Net Weight: 3.7 Lbs. (ea.)

Figure 2: Wheels

Figure 3: Motor

3.2 Motors

Motor Criteria

Wheels must be able to move a 200 lb robot. Each wheel is to be driven by its own motor for simplicity.

Motor Chosen

Manufacturer: CCL Industrial Motors (CIM)
Product: CIM Motor FR801-001

Technical Details:

Voltage: 12 V
Free Load Speed (RPM): 5342
Stall Torque (Oz-in): 346.9

3.3 Gearing

Gearing Criteria

The gearing was calculated by taking the RPM and torque of the Chiaphua motor at max power, 2671 RPM at 173.4 oz-in, and finding the ratio required to step it down to the RPM needed to go no more than 5 MPH; 168 RPM.

Design

The gearing ratio is 20:1 to well overtake any incline without stalling. A chain and sprocket system is being used due to ease of setup, assembly, and low cost.

The intermediate shaft is a half inch so that a 14 tooth and 48 tooth sprockets could be mounted on the same shaft. Other shaft sizes didn't include a combination to achieve the 20:1 ratio. This intermediate shaft is supported in metal bearings mounted in pillow blocks. The shafts are drilled and threated 1/4"-28 on both ends to bolt 1.5" OD washers to keep the shafts from sliding out of the bearings.

Figure 4: The 72 tooth 5/8" ID sprocket has 4 1/4" holes to mount to the intermediate shaft. The set-up allows for gear balancing to minimize wobble.

Figure 5: Assembled gears and wheels

4 Electrical Systems

4.1 Computer

As image processing is very CPU and memory intensive, the computer on RATT is an AMD Athlon x2 5000+ with 4GB of RAM. This provides for ample computational power for doing both the image processing and other robot control. In particular, the dual-core processor was chosen to allow image processing code to run on one core, and the general control code to run on the other core.

The operating systems and control software are installed on a 2GB Compact Flash card connected through an IDE→CF converter. A solid-state storage method was chosen for two purposes: to reduce the power consumption and eliminate moving parts. The computer and Compact Flash card are powered using an off-the-shelf 160W DC-DC converter.

4.2 Motor Controller

The motor controllers were constructed modularly to allow for flexibility in control methods. The H-Bridge for each motor is on its own PCB, and the controller for both H-Bridges is on a separate PCB. The controller consists of an Atmel microcontroller which communicates with the main computer over USB. The microcontroller receives commands from the computer and sets the speed and direction of the motor accordingly.

4.3 Wireless Emergency Stop

The emergency stop was designed to stop fulfill contest requirements of a remote kill switch. An Atmel microcontroller on the robot and on a battery powered handheld controller communicate using an off the shelf wireless transceiver. If the connection between the two devices is lost, or if the power is lost to the emergency stop board, power will be cut to the motors and the robot will be unable to move. If the transmitter is activated, power will be cut to the motors.

4.4 Accelerometer

The accelerometer on RATT is a MicroStrain 3DM-G which interfaces with the computer over the serial port. This accelerometer has a three axis angular accelerometer, as well as a three axis linear accelerometer. This allows the accurate calculation of current heading and position relative to the starting point by integration.

4.5 Camera

The camera used for line-detection and obstacle avoidance is an Elphel 353. This camera was chosen for a number of reasons: it allows for user-selectable framerate and resolution, it is accessible over Ethernet, and the hardware and software are licensed under the GPLv3. This last point was especially important; since the source code for the camera's onboard FPGA was available, we have the option to perform image processing directly on the camera itself.

4.6 Power Distribution

The power system of the robot consists of two separate 12V sealed lead acid batteries. One is dedicated to the computer, the other is responsible for powering the rest of the robot. This was done to reduce the possibility of the electrically noisy motors from impacting computer stability.

After the battery is a 180A circuit breaker, which then connects to a bus bar to distribute power to the motors and other components.

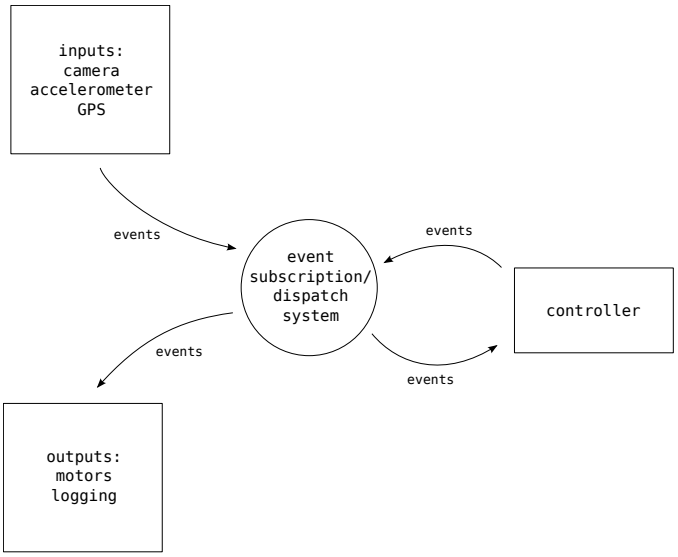

Figure 6: Diagram of the basic event loop

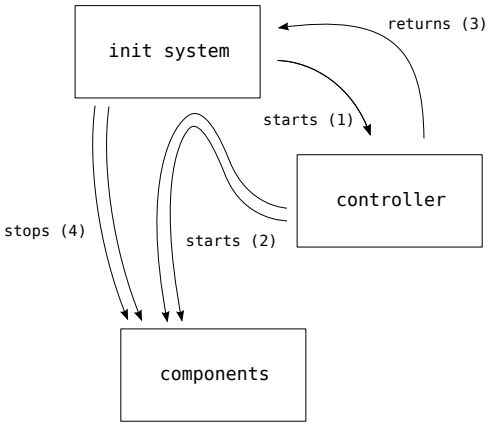

Figure 7: Diagram of Controller and Component initialization

5 Software Systems

In keeping with the open nature of our project the decision was made to use Open Source software throughout and release our own work under the GNU General Public License, version 3. We leveraged a wide variety of Open Source programs and libraries, visible in everything from RATT's operating system to the LaTeXsoftware used to typeset this document.

5.1 Operating System

The current stable release — 'etch' — of the Debian Linux distribution was chosen as RATT's operating system. The primary reason for this choice was Debian's strong reputation for stability. However, in order to improve hardware compatibility we upgraded the Linux kernel to the latest stable version.

5.2 Programming Languages

Python and C were chosen as the languages RATT's custom software is written in. This decision was largely based on their mutual compatibility and broad library support. Python is used for non-speed-critical portions of the code, such as serial and GPS communications, where the bottleneck is I/O. C is used for components that must run as fast as possible, such as image processing. Libraries and programs used include OpenCV and gpsd, a Linux GPS device daemon.

5.3 Architecture

At the core of our architecture is the event system as shown in figure 6. This system allows the various components of the robot to communicate asynchronously in a loosely-coupled manner. Outside of this system there are two major types of object:

Controllers
: The code responsible for processing input events and producing output events. Implements high-level functionality like obstacle avoidance or JAUS remote control. Only one controller may run at a time.

Components

A generic chunk of functionality. Most are responsible for gathering data from sensors and dispatching it as events or relaying event commands to hardware. May also implement event processing, for example translating an image event into a series of obstacle coordinates.

Finally, the init system is responsible for loading Controllers and their Component dependencies, as shown in figure 7.

The init system runs the Controller (1) which requests that any number of Controllers be started (2). When the Controller is done (3) the init system stops any Components that it started (4). This process may loop if the Controller indicates another Controller should take over after it shuts down.

5.4 Algorithms

For this first year of the competition most the time spent on software went into designing overall architecture of the software. This was done because the previous versions of the software proved to be difficult to learn and work with. Most of the algorithms needed for the competition are currently very simple and will be optimized for better performance during the coming year.

Lane Following

Lane following is accomplished by using custom edge detection algorithms to locate the positions of the bordering lines or dashes in the camera images. This data is combined with the orientation data from the 3D accelerometer in order to locate positions of the lanes with respect to the vehicle. In the case of dashed lines the lines is extended and connected to the best matching dash in order to obtain a "solid" line. Once the lane boundaries are obtained the outside is marked as "off-limits". The actual lane following is done with a wall following algorithm.

Obstacle Avoidance

High level obstacle avoidance is done by the lane following algorithms. Like the outside of the lane, obstacles are marked as "off-limits". This allows the wall following algorithm to route a path around the objects. In certain cases the wall following algorithm can also be switched to follow the opposite wall. That is, if an obstacle is found on the right side of the lane the wall follow algorithm will switch to follow the left side of the lane, thus easily avoiding the obstacle.

Actual obstacle detection will be more difficult for the RATT. Due to funding issues we have been unable to obtain a LIDAR unit so all obstacle detection and avoidance must be done using computer vision. The RATT will use a variety of redundant algorithms to detect and avoid obstacles. The control software will be using the OpenCV library for most of the image processing because it contains built in functions for object recognition, motion tracking, and stereo vision.

Waypoint Navigation

Waypoint navigation will be done using a Differential GPS. Obstacle avoidance algorithms employed during the Waypoint navigation will be designed to simply go around obstacles and continue on the previous course. In the case of a very wide obstacle it will also search for gaps in the obstacle that can be passed though.

6 System Integration

During the Fall and Winter quarters the Hardware, Electronics, and Software teams worked independently in order to design and build their respective components. Each team scheduled meetings independently and worked at their own pace with an administrative meeting held once per week in order to determine the status of the various portions of the vehicle.

By the start of the Spring quarter the vehicle was starting to come together and we shifted the meeting schedule to better facilitate integration of the components. In addition to the weekly status meeting regular weekend "work days" were also held so that the teams could work together on tasks that did not fit nicely into one specific topic area.

6.1 Electronics Assembly

The major integration task was to mount the various electrical devices onto the frame constructed by the hardware team. This required interaction with the majority of the team. For example, the hardware team actually did the construction and mounting but with the position of devices such as the accelerometer and camera were determined by the software team. Finally, the electronics team was responsible for determining how the different components would be wired together.

6.2 Communication Protocols

By Spring quarter the electronics team was nearing completion on the motor controllers and specifications were needed so that they could be accessed from the software. For simplicity this was done with a simple protocol over RS-232 serial.

6.3 Integration Testing

Before attempting to complete all the requirements for the competition a much simpler test was devised in order to test that everything was properly working together. This test was a simple object following task. This task is carried out by first specifying an object such as a orange or a dot from laser pointer. Next the object is placed in the RATT's field of view. If successful, the RATT will in some way reposition itself with respect to the object.

7 Vehicle Purchases Summary

Component	**Regular Cost**	**Total cost for team**
Mechanical		
Wheels	$\$16 \times 2$	\$32
Motors	$\$30 \times 2$	\$60
Gears	$\$30 \times 2$	\$60
Electronics		
Elphel 353 camera	\$1,000	\$0
Computer	\$315	\$315
Batteries	$\$119 \times 2$	\$200
Motor Controllers	$\$80 \times 2$	\$160
Kill Switch	\$80	\$80
Power Distribution	\$50	\$50
Software		
Everything	\$0	\$0
Total	\$1,957	\$957