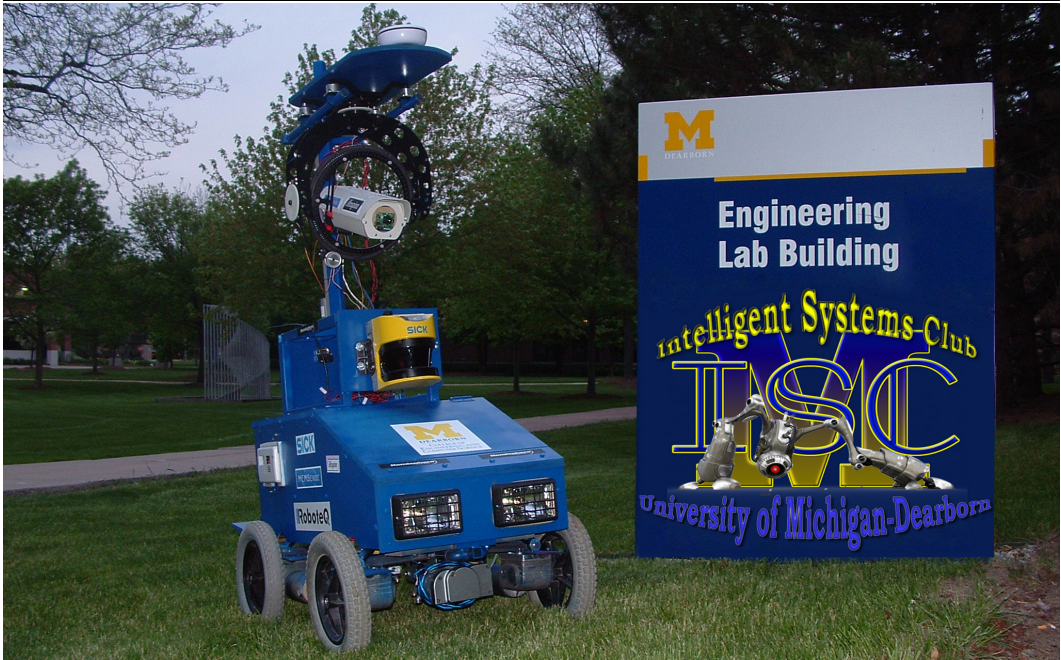


University of Michigan-Dearborn

RAEPTOR



Intelligent Ground Vehicle Competition 2008

TEAM MEMBERS

Allen Akroush, Gregory Czerniak, Edward Klacza, Lisa Linna, Anthony Lucente, Jason Smith



Chapter 1

Design Report

1.1 Introduction

For the 2008 IGVC, the University of Michigan - Dearborn Intelligent Systems Club has designed Raptor, an all-new robot platform. This autonomous robot is the work of a group of students, both undergraduate and graduate, who combined the strong points of previous IGVC attempts with innovations designed to mitigate the weaknesses exhibited in previous years.

1.2 Innovations

The Raptor has many innovative design features over the previous robots, including a gimbal-assisted camera stabilizer (see Section 1.3.4), a new chassis design with suspension (see Section 1.3.2), a fault-tolerant multithreaded software architecture (see Section 1.3.4), a powerful simulation software package (see Section 1.3.4), and GPU-accelerated image processing for line detection (see Section 1.3.4).

1.3 The Design

1.3.1 Early Planning

Early in the design phase of Raptor, many different strategies were considered. Very early in the planning phase, a diamond-shaped design was considered, which would have used an omni-wheel on the front and rear ends for steering. This design was eventually rejected due to the procurement difficulties in obtaining omni-wheels, as well as the unpredictability of the chassis's ability to turn. Over time, the design converged to a more conservative but reliable rectangular design that would still use omni-wheels to implement assisted differential steering.

This design, minus the omni-wheels, is what the Raptor eventually evolved to become. As Figure 1.1 shows, the early CAD design of the Raptor is very similar to the finished product.



Figure 1.1: A CAD model of an early Raptor design.

1.3.2 Mechanical Design

The robot is a four wheeled/four motor vehicle with skid steering and is constructed in two halves, the chassis and the roll cage. These are constructed of welded square steel tubing with the roll cage as a removable piece which bolts to the chassis. The roll cage is protected by wood panels which allow for easy access through two doors, one for quick access to electronic components and the other for its main laptop control. Additional panels can be easily removed by unbolting them for further access. Components of the robot are also easily replaceable should they fail. The motors, motor controllers, batteries, and electronics can be removed with either a set of wrenches or Allen keys. Even though items can be removed with ease, items are secure enough to withstand heavy usage without any shifting whatsoever. It has a simplified four wheel independent suspension along with two jacks bolted beneath the chassis. This allows easy debugging of motor control by lifting the wheels and entire body off the ground. The front of the chassis also has two headlamps to aid the vision system by eliminating shadows and allowing night time control.

The chassis body holds the power supplies and the robot controls with switches, including the emergency stop (pushbutton and wireless), and battery indicators. Mounted on top of the robot is another steel structure which accommodates the payload through a front access door which also supports a SICK laser range finder. In addition, this structure holds a linear actuator that supports and adapts for the systems 3-axis gimbal. Located on top of the gimbal is a flat panel which supports our global positioning systems primary and secondary GPS units.

1.3.3 Electrical Design

Electrical Systems

Raptor is equipped with two 12 volt 85 amp-hour deep cycle batteries connected in series, which provide 24 volt DC power to all robot subsystems. This system supplies power to the robot's motor controller, and two 24-to-12 volt DC/DC converters. The two DC/DC converters provide regulated power to the many robots components that need less than 24 volts to run. All three power distribution systems safely power everything on the robot, with separate protection circuits. This provides simultaneous short circuit, overload, over-voltage, and over-temperature protection. These three separate systems allow new modules to be incorporated into the robot with ease and affording peace of mind when adding new subsystems. The batteries are recharged through an on-board competition-grade 2-bank battery charger.

Motor Controller

Raptor uses a two-channel Roboteq AX2850 motor controller, with channels powering two sides of the robot. Each channel is capable of supplying a maximum of 120 amps for up to 30 seconds. Two 120 amp circuit breakers are placed on each channel to prevent over-current from destroying the motor controller. Two industrial waterproof optical encoders are used for speed and odometry. All information is then sent back to the robot's main computer through a serial cable.

Furthermore, the motor controllers are extremely efficient. They can be configured to ramp up speed on the motors to eliminate high start currents. Without a ramping of the motor speed, startup currents can reach up to 300A. By controlling the ramping speed, we limit the maximum startup current to 50A. Power distribution to the motors is controlled. If the motors on channel 1 run a little slower than the motors on channel 2, the motor controller is programmed to send a little more power to the first motor so that the motors on both channels will spin in sync.

Motors

After much research on what would be the best motor for the robot, the team decided to use the widely-available wheelchair motor. These motors run at approximately 140 RPM and have a 12.5-inch wheel connected to them, allowing Raptor to achieve the IGVC's 5 MPH speed limit. Two styles of these motors are created so that they can be mounted on both the left and right of the robot. Below are the ratings of the motors:

- Permanent magnet, reversible, rated for operation on 24 VDC.
- The no-load speed is approximately 140 RPM with a no-load current of approximately 3.6 amps.
- The RPM is about 130 RPM with a load of 60 in-lb at 7.9 amps.
- The max rated torque is 120 in-lb with an output of 94 RPM at 13.2 amps.

- Gear ratio of 32:1.

Sensors

The Raptor obtains information regarding the outside world from several inputs. These inputs are the odometry coming from the wheel encoder, a SICK scanning laser range finder, a Fire-i digital camera stabilized with a gimbal, a highly-accurate Hemisphere VS-100 Differential GPS, a 3-axis gyro, and a 3-axis accelerometer. Many of these sensors go directly into the decision-making algorithms of Raptor, while others (such as the gyro and accelerometer) are used to control the Intelligent Gimbal System (see Section 1.3.4).

To perform dead reckoning in environments where GPS is not available, the Raptor has a set of optical wheel encoders. These encoders count the angular distance travelled by each wheel, which can be used to perform odometry calculations to approximate the present location and heading of the robot. However, this information is used only to augment the GPS information.



Figure 1.2: A photograph of the wheel encoder.

The SICK scanning laser range finder is used by Raptor to detect obstacles. It has a field of view above 190 degrees, making Raptor much more aware of the environment than robots using hardware such as infrared or ultrasonic sensors. It communicates over a serial interface to the computer.



Figure 1.3: A photograph of the SICK range finder.

As mentioned in Section 1.3.4, Raptor has GPU-assisted image processing capabilities. However, this software is used in conjunction with a camera to supply a steady stream of images. The camera

used on Raptor is the Unibrain Fire-i digital camera with a wide-angle lens.



Figure 1.4: A photograph of the Unibrain Fire-i Camera.

Another considerable addition to the Raptor comes in the form of Differential GPS. Many IGVC contenders rely on inaccurate traditional GPS units, which are only accurate to 3 meters without correction. However, this year Hemisphere was extremely generous in supplying the Intelligent Systems Club a Hemisphere VS-100 Differential GPS accurate to 60 centimeters 95% of the time. This GPS unit has greatly increased the accuracy of the Raptor in both the Navigation and Autonomous Challenges.



Figure 1.5: A photograph of the Hemisphere VS-100 Differential GPS.

The Intelligent Gimbal System, as described in Section 1.3.4, uses a combination of a 3-axis gyroscope and a 3-axis accelerometer to obtain the information necessary to automatically stabilize the camera. These sensors are attached to an embedded ARM board through A/D converters. Figure 1.6 and Figure 1.7 are photographs of the accelerometer and gyro.

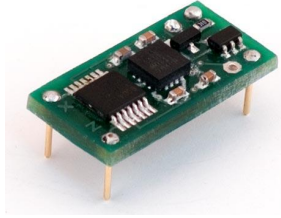


Figure 1.6: A photograph of the accelerometer.



Figure 1.7: A photograph of the Memsense gyro.

Power Distribution

All modules of Raptor are connected to one of the three power distribution systems inside of the robot. All wires use Anderson connectors with adequate current rate. The Anderson connectors provide easy connect/disconnect capability. The robot is activated by switching on power to the wireless E-Stop and pressing the channel that is connected to the main robot relay. This relays power to the motor controller and both DC/DC power supplies, thus activating all systems inside of the robot. There are also two more switches located on the outside of the robot that provide power to the SICK scanning laser range finder and Hemisphere VS100 DGPS. The schematics below (Figures 1.8 and 1.9) shows the robot's main power distribution systems.

1.3.4 Software Design

Strategy

Based on our previous experience, the University of Michigan Dearborn Intelligent Systems Club realized that a new software strategy would have to be devised in order to create IGVC-winning robot control software. To begin this process, the main Raptor programmers audited previously-written source code to determine where improvements could be made. From this audit, the Raptor engineers decided to build a new software foundation that would keep code organized and expandable well into the future. The result of this effort was a new software body entitled NGRP, or Next-Generation Robot Platform. Figure 1.10 shows a diagram of the NGRP data flow.

The primary goal of the NGRP system was to rigorously enforce modular programming into

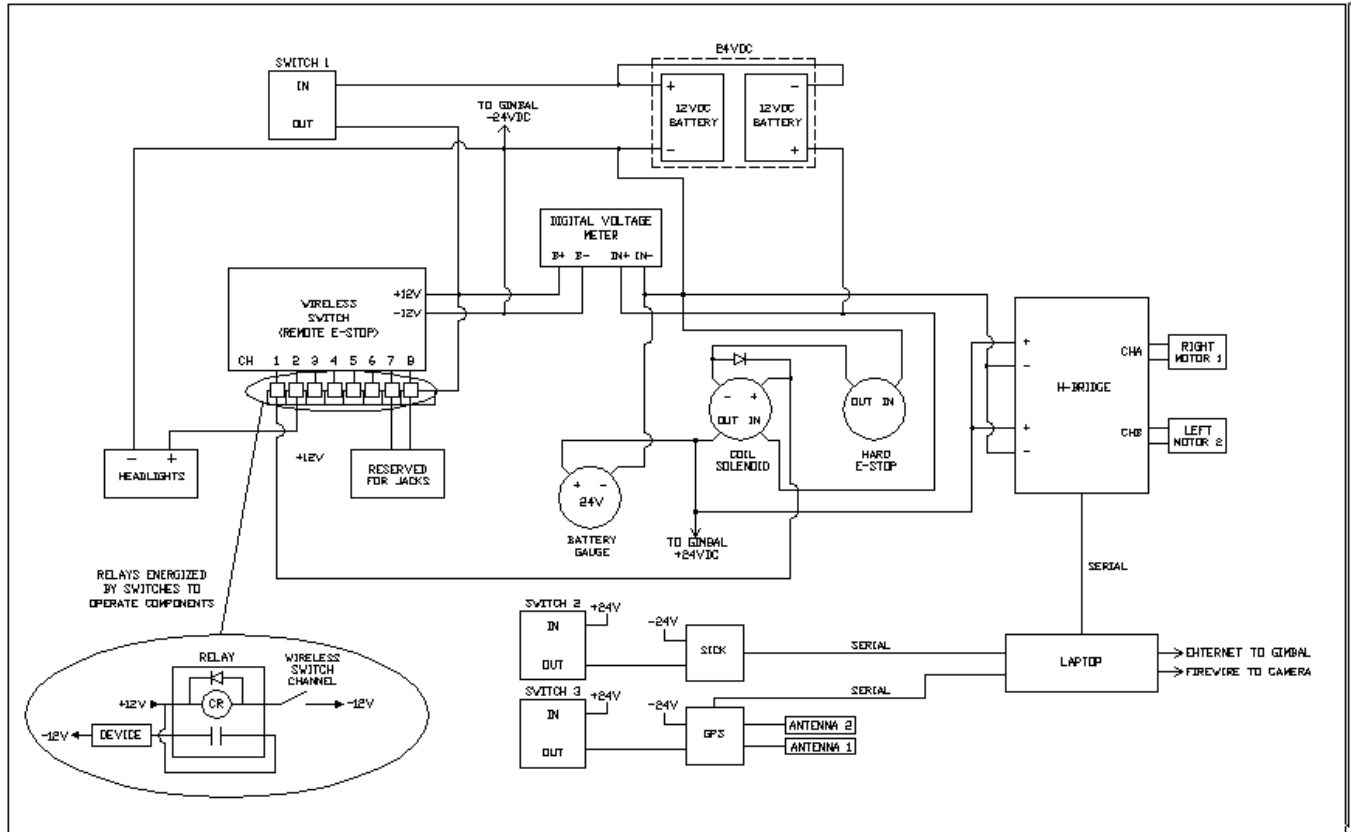


Figure 1.8: A schematic of the overall Raptor electrical system.

the design. The NGRP framework is a C++ application that defines a base object entitled NGRP-Module. All modules inherit the NGRPModule object and conform to its standard interface. Each module is compiled as a dynamically-linked library, and they are loaded at runtime by a "kernel" application obtaining its directives from a configuration file. These modules run as separate threads, and communicate with each other through UDP or shared memory depending on the speed requirements. By adopting this methodology, the software gained many benefits:

- Each module has well-defined inputs and outputs conforming to well-documented interfaces.
- Modules could be individually tested, and bugs are easier to pinpoint to a specific subsection of the software.
- By running each module as a thread, the system is ready for advances in multi-core processor technology.

Intelligent Gimbal System

To improve the robot's vision control a 3-axis gimbal has been implemented to stabilize the robot's camera. A multi-axis gimbal is a device designed to keep an object level as its surroundings change

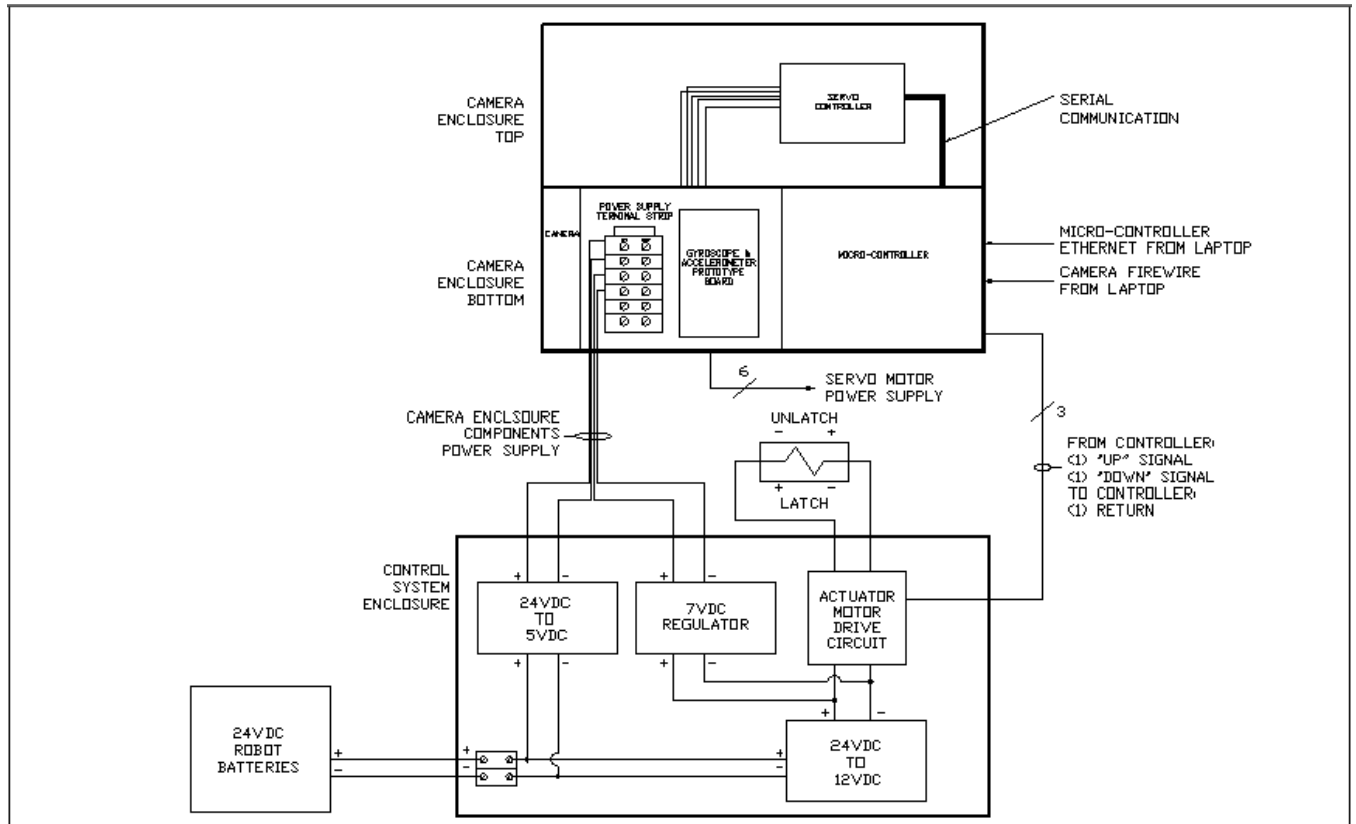


Figure 1.9: A schematic of the overall Intelligent Gimbal electrical system.

by rotating the object about multiple axes. The body of the gimbal is a circular camera mount able to rotate for roll, pitch and yaw via 3 servo motors. The servos are controlled by a servo motor controller that communicates with the system's microcontroller via an RS-232 connection. Each of the servo motors rotate based on A/D input from a 3-axis gyroscope and 3-axis accelerometer. Temperature is also monitored because overheating of the sensors can cause data error.

The gyroscope is used to sense change in velocity for each of the three axes. The accelerometer is initially used in order to give each axis a reference point; from this reference, it detects velocity change in the X, Y or Z coordinates and also monitors tilt from the accelerometer. An accelerometer is used to level the gyroscope which works by giving a reference to the direction of gravity; however, it is then required to fix any angular errors that will inevitably occur. Additionally, a linear actuator is used to support the camera mount and when the robot is in manual control, allows the gimbal to rise and fall in order to see over obstacles at the operator's command. The program was written in C and developed as an embedded, hard real-time system which depends upon the response of the servo motors from data received to occur within a given time frame in order for the system to work as required.

This control system has 2 feedback sensors, each providing feedback for 3 axes in order to stabilize the camera through 3 servo motors. The servo's speed and direction is determined based

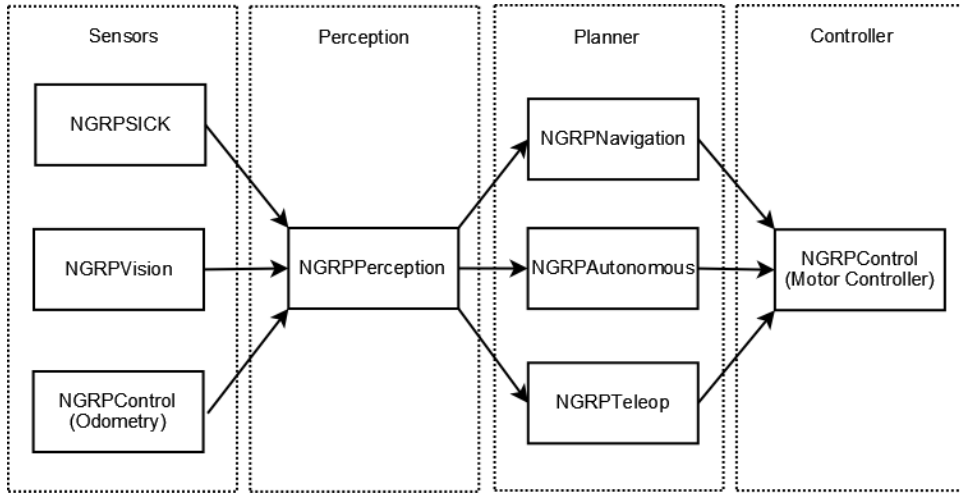


Figure 1.10: A block diagram of the modular NGRP design.

on data received from these sensors using a competency scale. Since the robot will be accelerating/decelerating, the data received from the accelerometer will not include only the tilt from the accelerometer; therefore the system bases its output to the servos on compensated/weighted gyroscope and accelerometer values. The further the overall value from the accelerometer is from $1g$, meaning the robot is accelerating or decelerating, the less competency the system puts on the accelerometer values and the more the system uses the gyroscope values. When there's no change of speed, the data equates to $1g = 9.81 \frac{m}{s^2} = \sqrt{x^2 + y^2 + z^2}$. The overall speed and direction sent to the servos are based on a value in which $(gyro\ data) * K1 + (accelerometer\ data) * K2 = (servo\ speed)$.

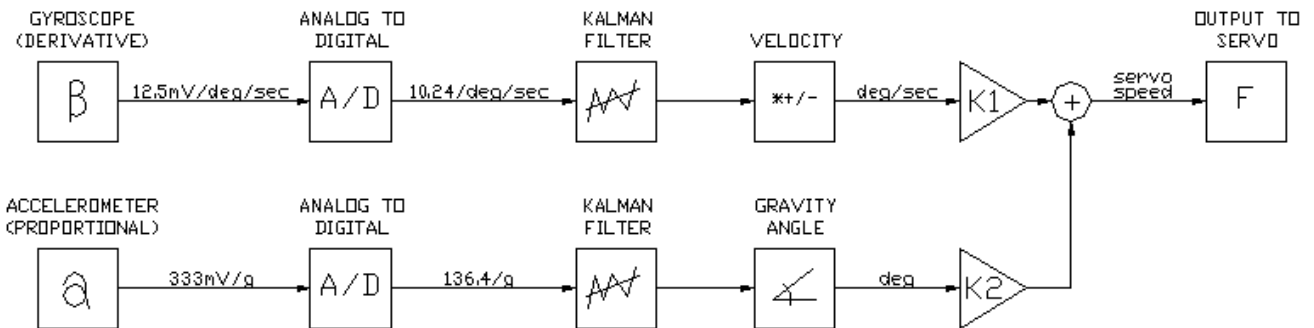


Figure 1.11: A diagram of the Intelligent Gimbal control system.

Computer Vision

A major innovation in Raptor this year is an enhanced image processing library designed to run on a Graphics Processing Unit (GPU). Typically, GPUs are used for visualization purposes, such as displaying CAD images or computer games. However, with the advent of readily-available pixel shader technology, modern GPUs can now be reprogrammed as low-cost, high-performance general-purpose vector processors.

In the past, most image processing has been done using a computer's central processing unit (CPU) using image processing libraries such as OpenCV. With modern graphics card technology, the onboard graphics processing unit (GPU) can out perform even the best dual-core CPU at the types of computation necessary for image processing. As robots become more sophisticated, an increasing number of sensors are competing for valuable CPU time. Since CPUs are busy processing so many other tasks, it follows to send the image processing tasks to a processor that can handle the task faster than the CPU and in parallel. Diverting image processing to the GPU simultaneously improves performance and uses the laptop's computational resources more efficiently.

A line detection algorithm using image morphology and a Hough transform was implemented using pixel shaders, which, according to internal Intelligent Systems Club benchmarks, ran up to six times faster than the CPU with a low-end laptop's embedded video card and up to 35 times faster on a GeForce 8600 GTS. However, the true benefit to the technology is that more CPU time is now available for the highly computationally-intensive tasks the robot must do such as navigation, path planning, and obstacle avoidance.

Simulation

The robot simulator was designed to meet the needs of the Intelligent Systems Club (ISC) for testing algorithms for controlling mobile robots. Testing new algorithms on simulated vehicles has many advantages. First, real vehicles can be potentially dangerous. Second, testing with a real vehicle requires access to the vehicle and space to use it. Setting up a large course like the ones used in competitions (roughly 100 by 100 meters for the IGVC) is not feasible. Also, the vehicle isn't always available to run.

The simulator solves these problems by providing an easy-to-use virtual testing environment with fairly realistic vehicle behavior. The simulator can run on any reasonable computer running Windows with a 3D-accelerated graphics card, which makes it readily available. It can also simulate the behavior of the vehicle faster than real time and allow many situations to be tested after any change to the control algorithms. Also, the simulator is capable of sending its simulated sensor data to NGRP in the same format as real sensor data, making it possible to test NGRP without a real robot present.

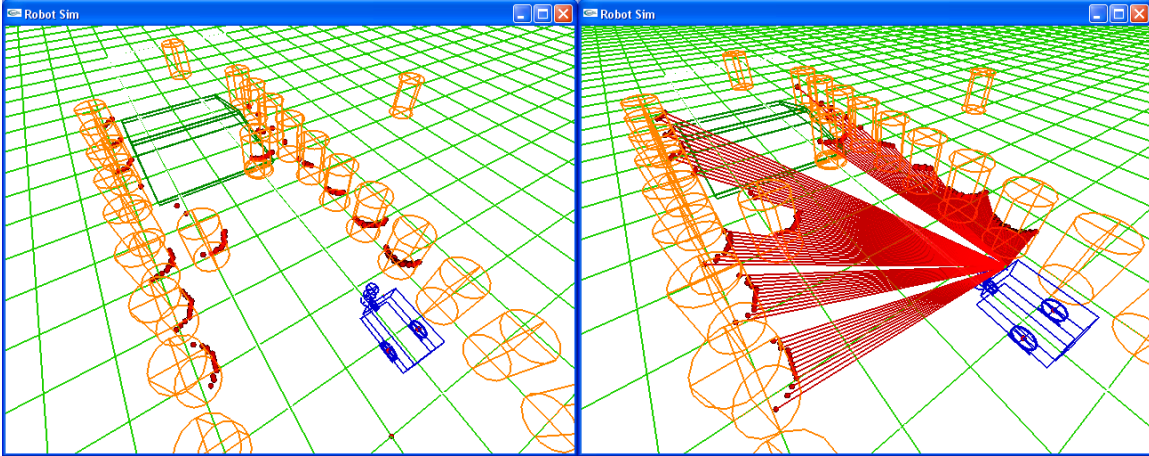


Figure 1.12: Screenshots of the Robot Simulator.

Systems Integration

A major goal this year was to improve the Intelligent Systems Club systems integration plan, especially the software. The NGRP software system was designed with systems integration in mind. As NGRP is a modular architecture, each subsystem was written as a dedicated module running as a separate subprogram (essentially a thread). However, without adequate communication between subsystems and their modules, system integration is impossible. To allow for communication, a universally-available cross-platform UDP object was made available to all NGRP modules. This, augmented with well-documented protocols described on a Wiki page created by the University of Michigan through their CTools database, allowed all NGRP programmers to communicate between their respective software modules with relative ease. Since the vehicle is controlled by UDP packets, there are many devices that can operate it: a handheld 802.11 remote, a laptop or PDA with wireless capability, or even a computer hardwired into the network, located directly on the robot.

At startup, NGRP's kernel code loads a configuration file, which contains a description of all relevant modules to load, along with all hosts, port numbers, COM ports, local settings for specific hardware, and any other information that may change rapidly. This feature, coupled with the modular/dataflow architecture used by NGRP, creates an environment which facilitates easy systems integration. An additional benefit is rapid reconfigurability – during the Raptor's testing phase, the NGRP configuration file allowed the testers to change settings within seconds and without recompiling the source code.

Lane and Obstacle Avoidance

To stay within the lanes, Raptor uses a combination of image morphology functions to isolate the white lines painted on the grass from the rest of the image. This pre-processing is performed using the GPU Image Processing subsystem by performing an erode operation on the image, and then subtracting the unaltered image from the eroded image. After some de-noising operations, a Hough

transform is applied, which is then sent to NGRP to be used in its decision-making algorithm.

For obstacle avoidance, Raptor mainly uses a scanning laser range finder to obtain information. With this information, Raptor uses fuzzy logic to avoid obstacles. During testing, many situations arose where the robot made incorrect decisions. To account for this, Raptor contains trigger situations that cause "instincts" to occur within the Raptor. These instinct functions are designed specifically to handle hard situations that occur, such as sand traps and switchbacks, by determining that the robot is indeed in one of these situations, and then performing specialized functions.

Computer

The innovations designed for this year require a powerful computer. The team chose the HP Pavillion DV6700T. The reasons for this choice were the Intel Core 2 Duo processor, which allows the NGRP multithreaded system to run more efficiently, 3 gigabytes of RAM, and the GeForce 8400M GS graphics card.

A critical choice was the graphics card. It was determined that the GeForce 8400M GS would provide an order-of-magnitude performance increase in image processing without severely compromising the computer's battery life. As a result, the laptop is capable of running for 6 hours per charge.

1.3.5 JAUS

This year, the Intelligent Systems Club has decided to participate in the JAUS Challenge for both Level 1 and Level 2.

Learning Process

The first task for an adequate JAUS integration was to learn the underlying purpose, procedures, and protocol for JAUS. To do this, the programmers read through the JAUS Reference Architecture Specification documents, which supplied a comprehensive picture of the entire JAUS specification. However, to achieve practical compliance with the specification, a trusted reference implementation was required. The Intelligent Systems Club solved this by obtaining a copy of the JAUS Compliance Test Suite (JCTS), which supplied the complete picture necessary to implement a JAUS parser.

Integration

As the Raptor's new software system uses UDP for inter-module communication, the new software framework already supplied the low-level protocol support for integrating JAUS into the software. Integration of JAUS was mainly a matter of parsing the incoming JAUS packets into signals recognizable by the rest of the framework. To efficiently write code to parse the packets, the team received the UDP packets coming from the JCTS software and then analyzed them in a hex editor. Studying the raw packet in conjunction with the Reference Architecture made writing an effective parser a much easier job.

Enforcing the new software architecture present in Raptor, JAUS support was added to the design as an additional module entitled NGRPJAUS. As other NGRP modules communicate with a custom-written UDP protocol, NGRPJAUS was designed to act as a JAUS emulation layer, converting the JAUS packets into NGRP packets that the rest of the system could understand. Due to the message-passing modular software architecture, integration of JAUS required a module to "translate" the JAUS messages to the right format.

Challenges

Due to the excellent documentation of JAUS in conjunction with example packets from the JCTS software, the difficulty of writing a JAUS parser was greatly reduced. To the contrary, the main challenge with implementation and integration of JAUS on Raptor was translating the packets to the internal NGRP protocols. Several features, including activation of autonomous and navigation modes from an external module, required some modifications to the data messages.

1.4 Analysis

Table 1.1 provides the cost analysis of Raptor. Table 1.2 provides the performance analysis.

A major concern while building Raptor was safety. To ensure safe driving conditions, the Raptor has two completely separate emergency stops. One is a large red button on the Raptor's chassis, which can be pushed to completely disable power to the electric motors. The other emergency stop is wirelessly activated, which allows Raptor's operator to disable the vehicle even if it drives somewhere hazardous to humans. The emergency stop is designed to automatically disable the robot if it leaves the effective range of the radio signal. Also, in the event of a bug disabling the onboard computer control system, Raptor's motor controller is configured to automatically stop after 1 second of no longer receiving a signal. This means that hardware faults and software faults do not put the operator (or spectators) of Raptor in danger.

Reliability was another major design consideration of the Raptor. As mentioned in Section 1.3.3, large amounts of protective measures were added to the electrical system to prevent electrical shorts that may damage components. On the software side, the modular architecture of NGRP means that even if one module of the program fails, other portions will continue to run. The NGRP kernel also has the ability to monitor the status of modules and restart modules that may have crashed.

Raptor is built with sturdy medium-density fiberboard, which was specifically used to aid in the durability of the Raptor's body. Also, the incorporation of shocks into the chassis was an intentional design decision to aid in durability by giving a smoother ride, reducing the vibration that can destroy electrical and computer components.

1.5 Acknowledgements

The team would like to thank Larry Sieh for his help and support for the project.

1.6 Signature

I certify that the design and creation of Raptor has been significant and is equivalent to what might be awarded credit in a senior design course.

Dr. Narasimhurthi Natarajan

Professor of the Department of Electrical and Computer Engineering, University of Michigan - Dearborn

Date

#	Description	Actual Cost	Cost To Team
Chassis Construction / Mechanical Design			
1	Square Metal Tubing and Wood Panels	288.50	288.50
4	Electric Wheelchair Wheels 12.5 Wheel	743.80	60.00
4	Wheelchair Electric Motors	1,864.00	240.00
2	Two Ton Jacks	120.00	120.00
4	Shock absorbers (off-Go Kart, ATV, Mini Bike, Etc)	80.00	80.00
Motor Control Circuitry			
1	Roboteq Motor Controller Ax2850	620.00	575.00
Robot Computer			
1	Dell Laptop Computer	999.00	999.00
Microcontrollers			
1	Camera Gimbal, Embedded Arm Board	332.00	332.00
Sensors			
2	Wheel Encoders: Quad Optical Encoders, same ones as on WOLF	338.00	316.00
1	Sick Laser Sensor	5273.00	389.00
GPS			
1	Hemisphere VS100 Series DGPS	4,195.00	0.00
Camera			
1	Fire-i Digital Board Camera-Camera with Wide-Angle Lens	130.00	130.00
1	CopterWorks 3-Axis Gimbal - Pan, Tilt, Yaw	2,520.00	650.00
1	MEMSense 3-Axis Rate Gyroscope	810.00	0.00
1	3-Axis DE-ACCM3D Accelerometer	35.00	35.00
1	Servo Controller	39.95	39.95
1	Linear Actuators	109.00	109.00
1	Camera Enclosure	54.95	54.95
Controller Transmitter and Receiver			
1	Wireless Emergency Stop Receiver	60.00	60.00
1	Wireless Emergency Stop Transmitter (1500 feet)	37.00	37.00
Accessories			
2	Transcontinental Deep Cycle Batteries 12V 85AH	130.00	130.00
1	Multi-Bank Battery Charger: 5/5amp 2 Bank Charger	189.99	189.99
1	Mean Well USA, 24v to 12v DC/DC Converter	185.00	0.00
1	Mean Well USA, 24v to 5v DC/DC Converter	185.00	0.00
1	Exterior lights for eye catching appeal	100.00	100.00
1	Random Switches, Wires, Connectors, Electronic Parts, Enclosures, Etc	500.00	500.00
6% Sales Tax		\$1,196.34	\$326.12
Total		\$21,135.34	\$5,761.52

Table 1.1: The table of expenses for Raptor.

Attribute	Design Prediction
Maximum Speed	5.0mph
Climbing Ability	30 degree ramp
Nominal Power Consumption (Watts = Amps x Volts)	500 watts
Battery Operating Time (24v 85AH Battery System)	4 hours
Distances at which objects can be detected	5.5 meters
Waypoint accuracy (DGPS)	<60cm 95% of the time
Reaction Times	60-120 ms depending on configuration
How vehicle deals with complex obstacles	Reactive Fuzzy Logic approach

Table 1.2: Performance attributes of the Raptor.