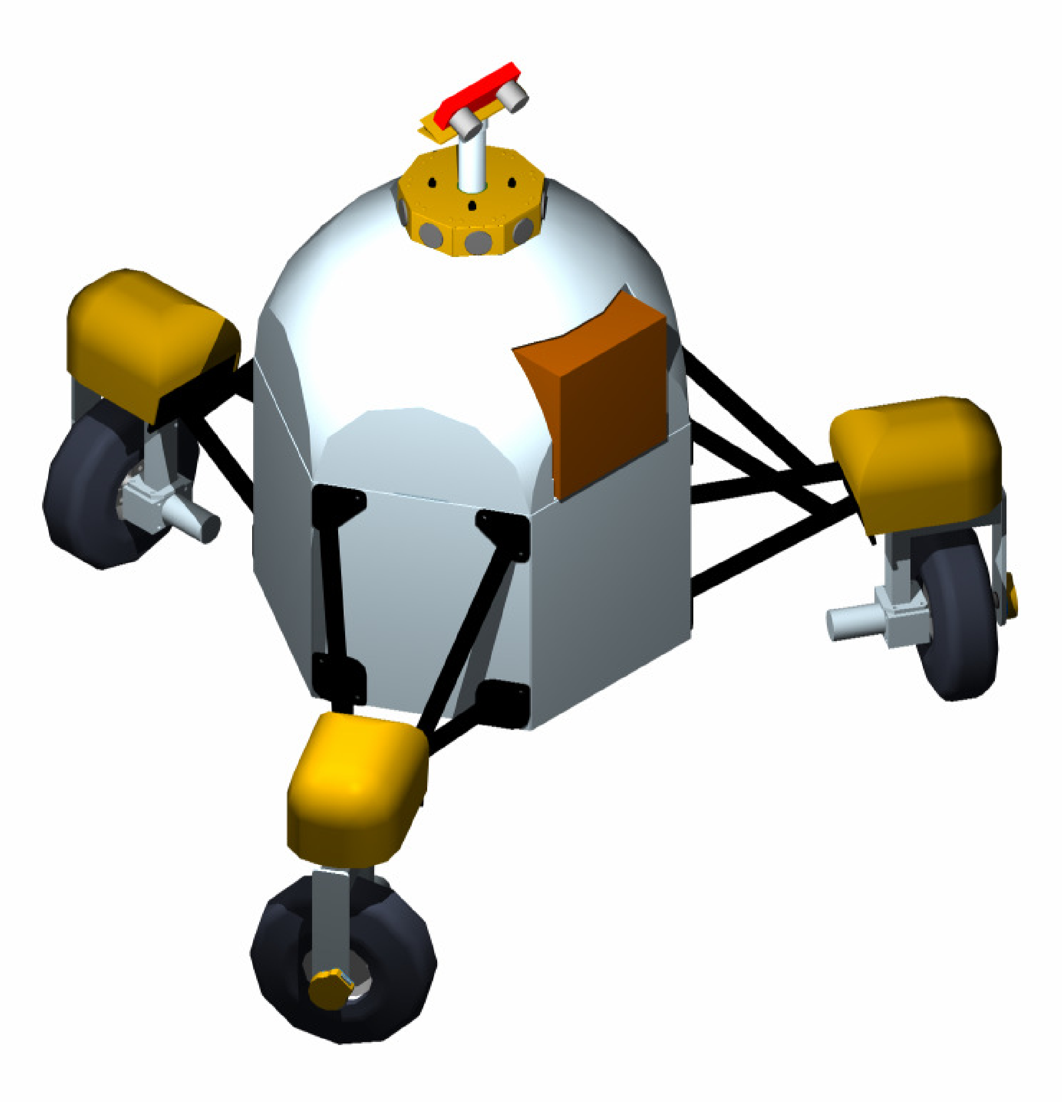
14th Annual Intelligent Ground Vehicle Competition

Optical Prime



University of Missouri-Rolla

Robotics Competition Team

2006 Design Report

1.0 Overview

Optical Prime is a continuation of the 2005 design from the University of Missouri-Rolla (UMR). The three-wheeled platform was very stable and the sensors were effective, but much of the rest of the robot had to be redesigned. Optical Prime has new motors, control systems and new artificial intelligence. This will allow it to be competitive despite its problems in 2005. This robot has been designed over two years to be maneuverable, stable, intelligent and fast.

2.0 Innovations

The 2005 design had many innovations that were carried over into this robot (Figure 1). First, it is inherently stable due to its three-wheeled design. Second, it is omni-directional; each wheel can turn 270° and the motors can run forwards or backwards. Third, it uses the minimum amount of sensors. Stereovision cameras are the primary sensors and they were found to be less expensive and easier to use than LIDAR sensors. Infrared sensors are used on the tips of the wheel pods to detect close obstacles that may be missed by vision sensors.

The improvements over last year's design are also innovative. The new environment model and path planning algorithms are simpler and faster than last year's version. The computer vision is three times faster due to the use of a filtering algorithm instead of a feature detection algorithm. The new control system is more modular and efficient. Special microcontrollers were implemented on printed circuit boards for this competition to improve the reliability of the control system.

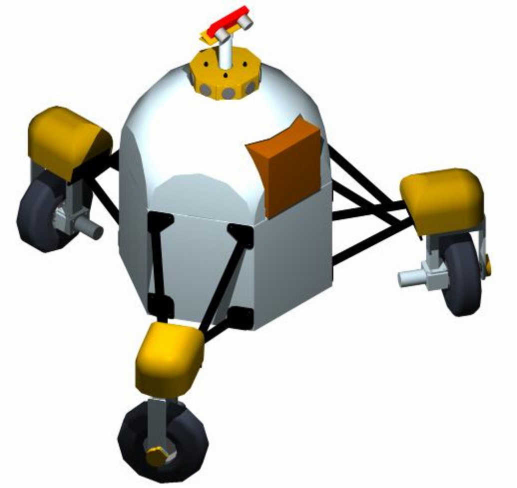


Figure 1: AutoCAD Render of Optical Prime

3.0 Design Process

Optical prime was initially designed for the 2005 IGVC but was redesigned following the process in Figure 2 to compete this year.

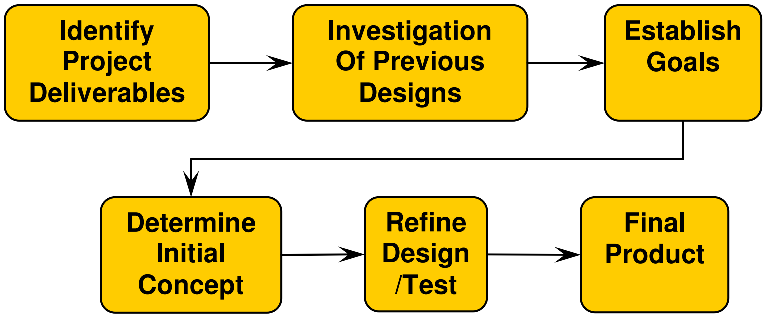


Figure 2: Design Process

Given that the rules for the competition are stated very specifically, it was easy to determine the project deliverables. For the primary competition the robot must be able to find its way through an obstacle course composed of construction warning objects (barrels, netting and cones) while staying between two white lines. The same robot must be able to run through a navigation challenge that is similar to the traveling salesman problem.

The team analyzed last year's design before commencing work on the current robot. The major limitations of the older model were found in the motors and control system, but it was also found that some software systems could be optimized and reorganized. In addition to this analysis, the team studied the successful robots from the 2005 IGVC and created a list of possible improvements for the 2006 design.

This list of improvements was compiled into a statement of goals to make the new robot fit for the competition. The major goals were to replace the motors, engineer an efficient and reliable control system and optimize the software from 2005.

The remainder of the design process was applied to the software and control system since the physical structure was already built. Many different algorithms were tested to determine the best solution.

4.0 Team Organization

The team consists of members from nine different majors including mechanical, electrical and computer engineering. It is divided into a group responsible for the mechanical and structural components of the robot and a group responsible for the intelligence and controls of the robot. Each group has a project manager who is responsible for managing their section and reporting progress to the Vice President of the team. A group of officers is responsible for business and administration of the team as a whole. See Figure 3 for the hierarchy of administration. See Appendix A for a full list of team members. Both robots entered by UMR this year were designed and built by the same team.

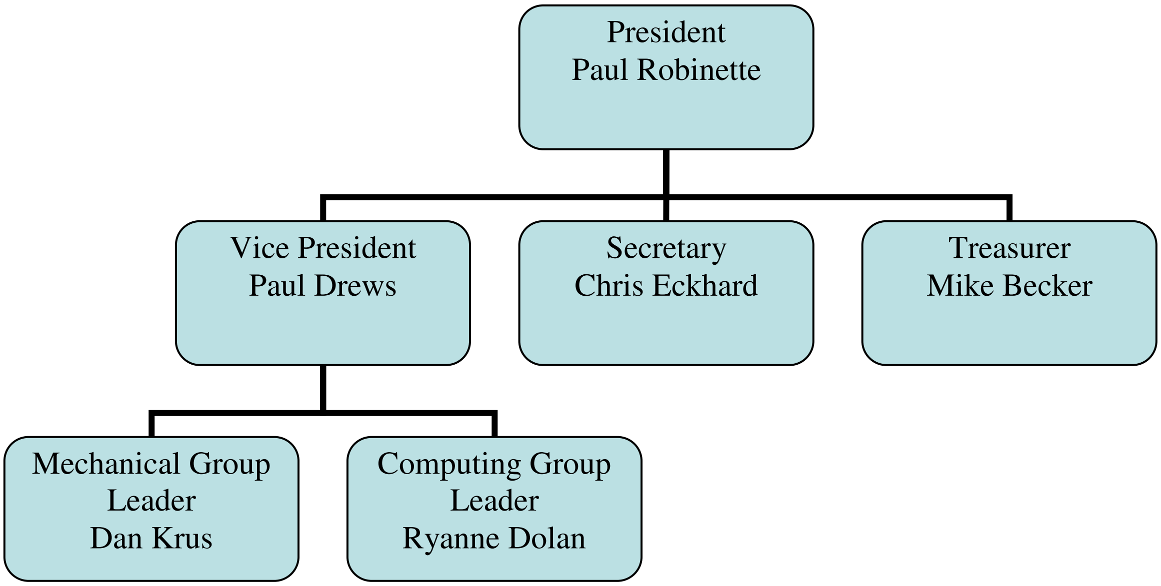


Figure 3: Organization Chart

5.0 Mechanical Design

5.1 Chassis

The main chassis is hexagonal in shape, and constructed primarily out of 1" square steel tubing. There are three support legs for the wheels, fabricated out of 1" round steel tubing and ¼" steel plates. The 0.065" thick wall steel tubes are welded together, creating a very rugged and durable frame. The frame is covered in a Plexiglas and fiberglass shell for increased visual appeal.

One of the main goals for this robot is to be as modular as possible. Each leg is attached to the main frame with ¼" bolts. The legs are identical, which allows them to be interchangeable and easily replaced if one should be damaged. Detaching the legs is the primary method of making the robot more portable. The upper support structure is also removable from the main chassis, which allows for easier access to the internal components. See Figure 4 below for an AutoCAD rendering of the robot disassembled into its primary components.

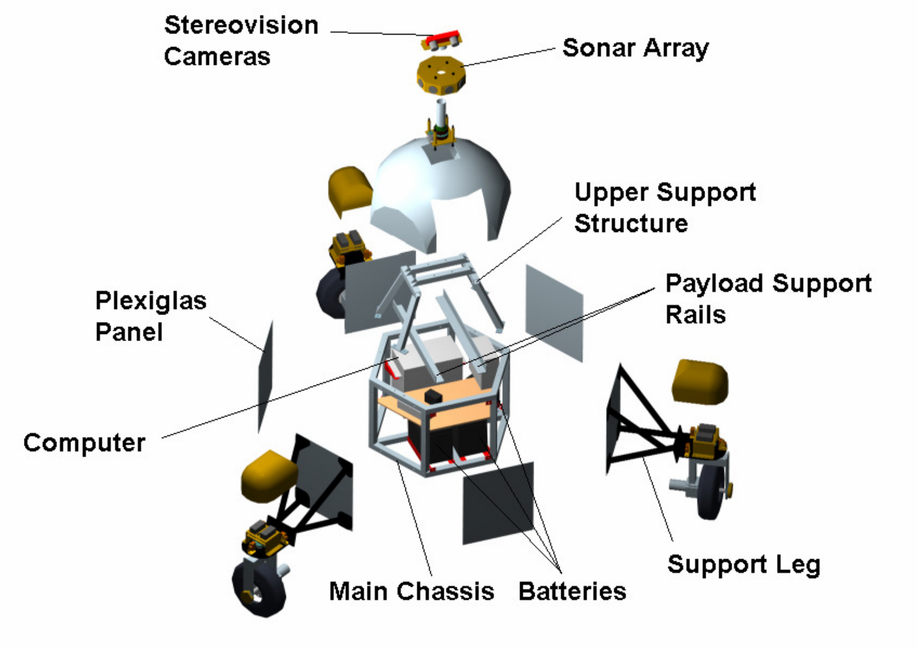


Figure 4: Disassembled Render of Optical Prime

5.2 Drivetrain

Each wheel is powered by an NPC gear motor running at 12 volts. The mechanical group selected this particular motor because of its compact size and strong construction. The team decided that the robot should be able to accelerate to 5 mph in under 2 meters, and the maximum speed of the motors should result in a speed of the robot at about 5 mph. The ideal motor for these parameters would have a power of 93.1 watts and would turn at 17.6 rad/sec. The selected motors have a power of 56.69 watts and turn at 9.74 rad/sec which results in about 3 mph. The output shaft of the gearbox is attached directly to the wheel shaft. This assembly is mounted to the yoke, and can be easily detached if the motor or gearbox ever needs replacement. An optical encoder is mounted to the opposite end of the drive shaft and provides feedback to the motor controller about wheel rotation rate.

5.3 Steering

Each wheel is steered via two ¼ scale, HiTec HS-805BB servos. A single-turn potentiometer is mounted to the end of the rotation shaft, allowing for precise control over the direction of the wheel. See Figure 5 below for a close-up of the steering mechanism. Each wheel is capable of rotating 270°, which allows for the entire robot to move in any direction and even turn about its own axis. The 270° limit prevents excessive twisting of the wires running to the motor and encoder around the yoke. The steering mechanism in Figure 5 was designed and built by UMR in 2005 for the IGVC.

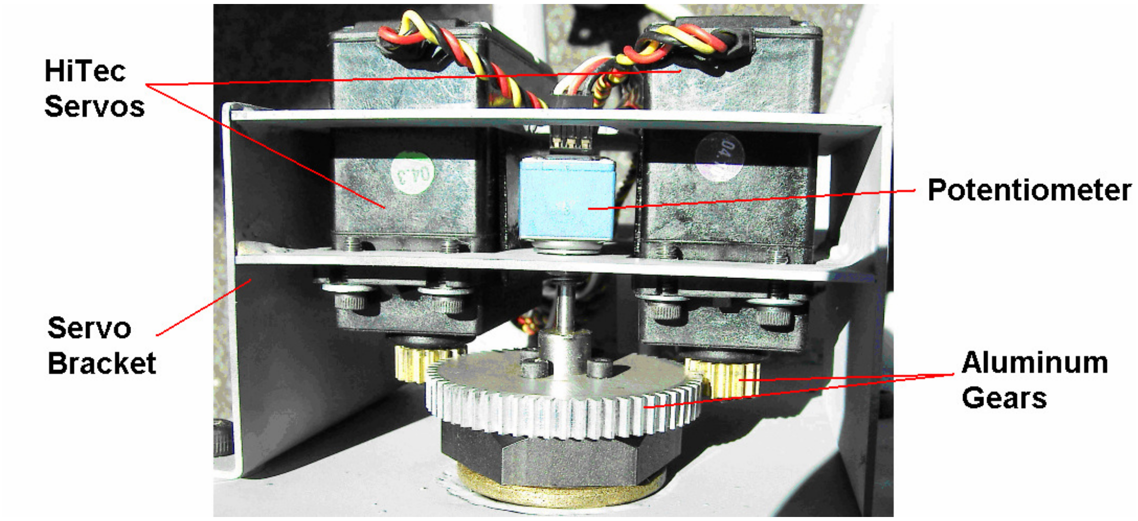


Figure 5: Steering System

5.4 Stability and Vibration Isolation

The minimum mechanically stable platform consists of three contact points with the ground, so the team selected a three-wheel configuration to reduce the number of physical components. A three-wheel design also has the advantage of all wheels being in contact with the ground at all times. With any vehicle it is desirable to have the lowest center of gravity possible in order to handle quick acceleration or deceleration, especially on a slope. As a result, the robot has a wide wheelbase and a compact central chassis. The most crucial weight factor in the stability of the platform is the weight of the batteries, which have a combined weight of 60 lbs. Therefore, the batteries are placed as low as possible inside the main chassis.

The only component on the robot sensitive to vibrations is the computer. The inflated wheels absorb most major vibrations, given the 5 mph speed limit, and can be inflated/deflated to change the affect of vibrations on the computer. In addition, the computer itself rests on rubber shock absorbers, thus further reducing vibrations.

5.5 Physical Parameters

The robot has a total weight, including the 20 lb payload, of 195 lbs, which is 25 lbs under the target weight of 220 lbs. After completing the vehicle, the team noted that the frame could be significantly lightened by using more expensive, thinner walled steel without compromising structural integrity. However, the low cost and ease of fabrication of the steel used on this platform served as the main material selection criterion.

The robot has an overall width of 52.7 inches at the widest point, and has a height of 44.26 inches. The main chassis has a ground clearance of 10 inches. See Figure 6 below for the dimensioned vehicle.

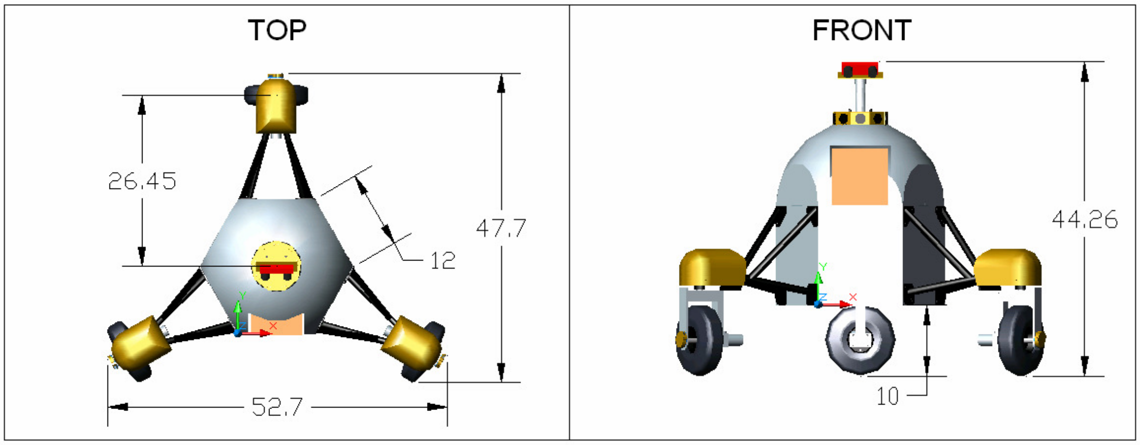


Figure 6: Dimensions of Optical Prime

6.0 Electrical Systems

Power for all motors and servos is provided by two Helios 12 VDC, 35 A-hr, deep-cycle, sealed lead acid batteries. These two batteries are connected in parallel to a terminal block, providing a total of 70 A-hr. The computer is powered by a similar Helios 12 VDC, 12 A-hr battery. Having a power supply for the computer separate from the motors provides a cleaner source of power and enhances the robot’s modularity, allowing the computer to continue to run while the main batteries are charging. The main power batteries were designed to run the robot at full speed for 1 hour. Very little power is drawn while the robot is standing still, so the batteries would last a very long time on an idle robot. The computer battery has been tested up to 1 hour of continuous use and has doubled that while idling. The sonar and infrared sensors receive their power from the 5 VDC power supply, which is connected to the 12 VDC terminal block. The stereovision cameras get their power from the computer over the IEEE 1394 cable.

7.0 Sensors

Sensors must be used by the robot to detect its environment. To keep the overall cost of the robot low a minimal amount of sensors are used. Stereovision cameras were chosen over a LIDAR system due to their low cost and ease of use. A LIDAR system would still require a vision component to detect the lines, so the problem was actually simplified by using computer vision. The Videre stereovision camera was determined to be the best fit for this competition since it can give accurate distance data up to 15 feet. The camera has trouble detecting objects within 2 feet of the robot, so low cost infrared rangefinders are used. Sonar sensors were implemented in the 2005 robot but in the redesign they were determined to be redundant.

Figure 7: Videre Stereovison Camera

The position of the robot is determined by the GPS receiver. If necessary, the wheel encoders can be used in conjunction with the digital compass to track the movement of the robot when a GPS fix in unavailable. When the robot starts in the Navigation Challenge it can use the digital compass to determine which heading it should turn to before the GPS has a chance to receive accurate data. The GPS is a Garmin GPS 18 1 Hz unit which has WAAS differential capability so that it can give the position within 1.5 meters. This should be accurate enough for the competition.

8.0 Software Design

Four main principles were used when designing the software for Optical Prime: modularity, rapid development, prototyping and hierarchical design. The software was designed to be modular so that each component could be developed and tested independently before the whole system was integrated. See Figure 8 for a diagram of the software. Rapid development was necessary given the one year design-build-compete schedule. Development time was reduced by using the simple syntax of interpreted languages (mostly Lua). High level logic was used to prototype algorithms quickly in simulation and in the field. Hierarchical design is necessary for a robot of this complexity; this can be seen in Figure 8 where high-level logic and low-level control systems are built out of simple modules. The major components of the software system are the controls, the computer vision, the model of the environment and the path planning. Given the similarities between UMR's two robots, the software systems are nearly identical.

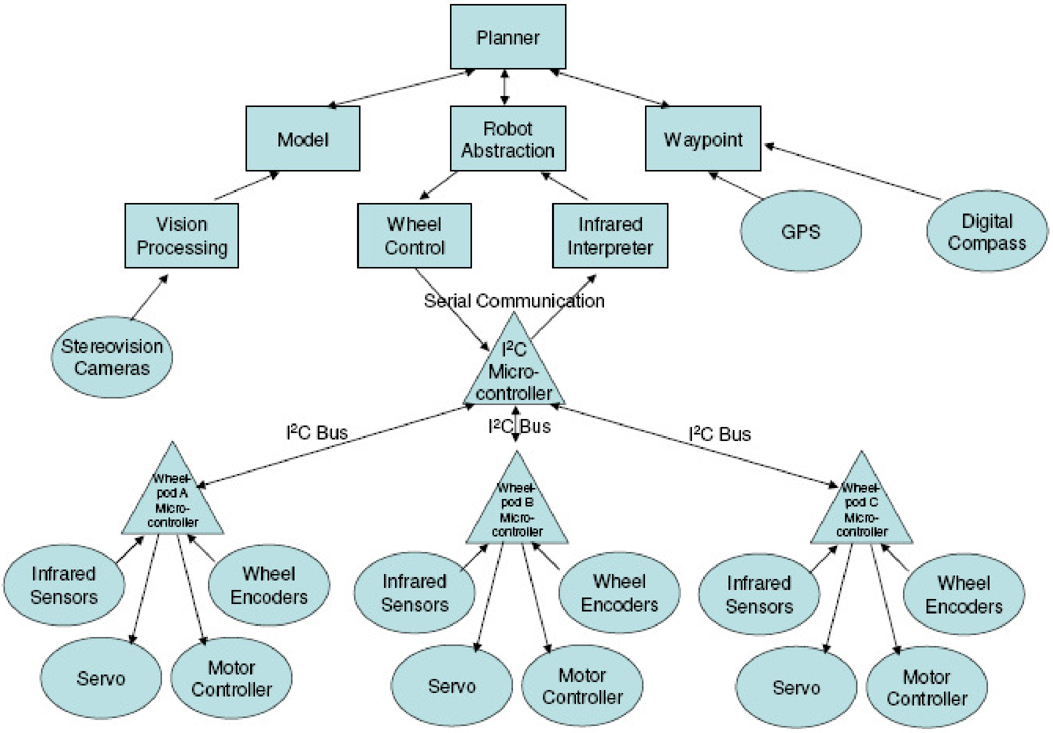


Figure 8: Software Organization Chart

The robot abstraction module provides an interface to each individual sensor and actuator on the robot. These interfaces are abstracted with simple commands that control the robot as a unit. This allows the planner to use simplified commands for navigation and also allows for interaction with individual components during testing.

The infrared interpreter module feeds data to the robot abstraction module so that the robot can react to obstacles too close for the cameras to see. The wheel control module takes high-level commands from the robot abstraction and converts these into simple movement commands to send through serial and I^2C buses to the wheelpod microcontrollers.

8.1 Computer Vision

Optical Prime uses stereovision to detect its environment. To interpret this data it uses a combination of the Small Vision System Library, the Open Computer Vision Library and in-house software developed at UMR. Instead of a feature based system as was implemented last year, a simple filter method was used for Optical Prime. Last year, the image was filtered for color but then objects were found and lines were detected. This was determined to be inefficient and was replaced with a system that does not waste time finding obstacles but instead finds regions of interest and sends them to be stored in the environment model. Since the robot only needs to pay attention to white and orange objects, it was logical to ignore everything else.

First, the image is filtered for white and orange colors. This creates regions of interest which correspond to lines on the ground and physical obstructions like barrels. Then the distance to each region of interest is found using disparity data from the stereovision cameras. Finally the model is updated with the three dimensional position of the region.

The performance of the cameras has been increased from 5 Hz in 2005 to 15 Hz in the current model due to these changes. This speed will help the robot to react quickly and efficiently. Below are two images from the vision system. Figure 9 is the raw image of a barrel in our shop and Figure 10 is the image after the filter. Notice how the filter ignores irrelevant components in the image and focuses on the orange barrel and the white board.

Figure 9: Raw Sample Image of Barrel

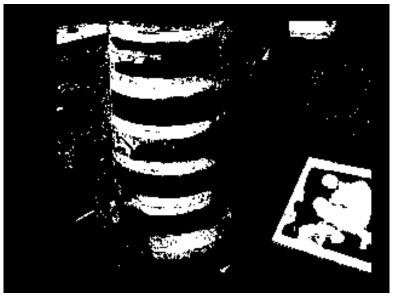


Figure 10: Image in Figure 9 Filtered for Orange and White

8.2 Environment Model

Using the data gathered by the stereovision cameras the computer develops a model of its environment. Since the computer only interprets data in the form of regions of interest, it creates a map of these regions and gives each a cost. The cost refers to how traversable it is for the robot. White lines are stored and avoided as if they were physical obstacles. The current model was simplified from an obstacle based system implemented last year to a cost map. This cost map takes in three dimensional data from the vision system and converts it to a two dimensional, pixel-based representation of the world. The three dimensional component was kept in order to make the model functional for other projects. After updating the cost map with new data the model is smoothed with a Gaussian blur so that the robot will see an increasing risk as it approaches a high cost area.

At the request of the planning module, the model will evaluate the cost along a linear path between any two coordinates. It does this by summing each pixel's cost with a weight towards pixels closer to the robot. This allows the robot to see nearer (and more certain) obstacle with a higher cost than obstacles at a distance.

The model updates each time the vision system receives a new frame, which has been measured at 15 Hz. Since the controls are able to receive commands at this rate also, the robot has a response time of 1/15 of a second. At 5 mph this corresponds to just 5 inches of travel for the robot. Below are two different screenshots of the model. Figure 11 is the model after it received data from the setup in Figures 9 and 10. Figure 12 is a simulation of many obstacles placed randomly around a robot within a 15 foot radius.

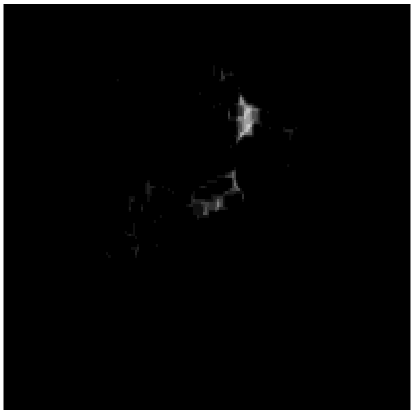


Figure 11: Cost Map From Figures 9 and 10

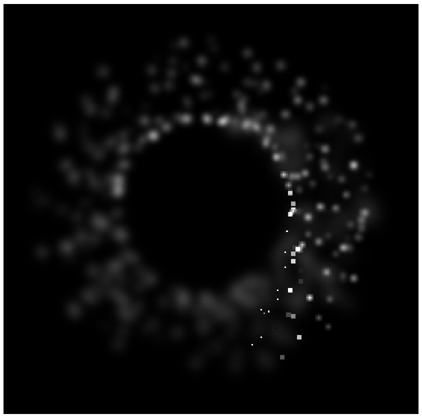


Figure 12: Simulated Cost Map

8.3 Planner

The environment model is used by the planner module to plot a course through all visible obstacles. The planner starts by asking the model to evaluate a linear path from the robot to a target. If the direct path to the target is obstructed, then the planner will bend the path until an unobstructed one is found. If no clear path is found then the path of least resistance will be chosen. The robot will follow this path while updating its cost map to see if any new high cost regions have been detected in front of it. If an obstacle is detected along the planned path, then the path is recalculated.

For the Navigation Challenge the targets will be the waypoints provided. For the Obstacle Course the targets will be set at the furthest unobstructed point that the robot can see. This target will be reevaluated periodically to keep the robot progressing along the forward path. Since the only data considered is the forward view, the robot has a strong bias towards moving forward. Due to this implementation no line following is necessary; the robot will naturally find its way around the circle. It is possible to trap the robot in a corner or make it turn around and drive the wrong way; however, this situation is highly unlikely in practice. The system should avoid any traps that it can see, so the robot will not be caught in any dead-ends that can be seen from 15 feet away.

8.4 Controls

Once an appropriate path is chosen the control system is responsible for moving the robot using low-level commands sent to the microcontrollers on each wheel pod. The control system for the IGVC robot uses an innovative distributed processing design that allows the use of inexpensive nodes connected to an inter-robot bus instead of a traditional centralized controller. This design provides several important advantages. It is modular, allowing simple component interchange and replacement, inexpensive due to distributed processing, and organized to use a single serial bus instead of multiple wires.

Each node consists of a Phillips 89LPC938 microcontroller module (Figure 13) and supporting connecting hardware. The 8051 module was designed by the team and incorporates programming hardware for the 8051 and simple connections to a serial bus and I^2C bus. This module is inexpensive, in-system reprogrammable, and can easily be swapped between wheel pods to facilitate quick hardware debugging and replacement.

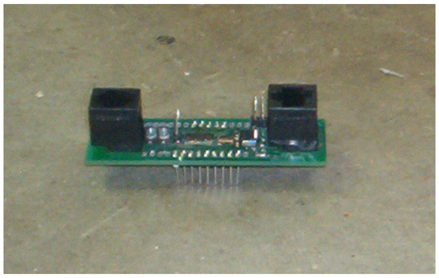


Figure 13: Microcontroller Board Designed by UMR for IGVC

Since these modules are small, easily programmable and utilize a simple I^2C bus, a single 4-wire bundle can carry power and data from a central controller to different nodes on the robot. This greatly simplifies wiring complexity and makes designing and debugging a much easier task.

Because these modules can be easily integrated into different parts of the robot and connected via a common bus, much less powerful, less expensive, and less complex processors can be used. Instead of having one powerful central computer trying to run 6 feedback loops in real time, this task can be offloaded to several less expensive slave processors that are controlled from a central node.

In our current robot, there are 4 separate modules controlling the low-level drive systems of the robot. On each wheel pod, there is one microcontroller module that controls the steering motor, drive motor, and infrared range finders. Robot Power's Open Source Motor Controllers are used to directly control the motors' speeds. The microcontroller communicates with these and the servos by generating pulse width modulation signals. These three microcontroller modules communicate via an I^2C bus to another microcontroller module. This fourth module acts as the interface between the wheel pod controllers and the central PC. This architecture allows for simple and flexible wiring within the robot and efficient distributed processing.

9.0 Integration

It is not a trivial process to combine several sets of sensor data with many outputs to motors and servos. Using the inherent advantages of our modular and hierarchal design the task was greatly simplified. First, the data is taken into the computer via the vision, waypoint and robot abstraction modules. Next, it is processed into a cost map of the world around it. Then, a course is plotted through the least costly areas so that the robot can travel safely to its target. For the Navigation Challenge, the robot finds the best path in the direction of a given waypoint. For the Obstacle Course, the robot picks the best point within sight and sets a waypoint there. Finally, the control system is told where to move the robot.

10.0 Safety

In addition to the safety of the electrical system, the robot has two emergency stop buttons. One is mounted next to the camera within easy arm reach and the other is a wireless device. Each of these is connected to the disable input on the motor controller. When the input is triggered the motors will stop. This will halt the robot in just a few feet, even at 5 mph. Using the microcontrollers, each motor is hardware limited to 5 mph.

11.0 Vehicle Cost

When Optical Prime was originally designed for the 2005 competition the cost estimate was $8000. This year, the team budgeted $1000 in repairs to purchase new motors and components for a new control system. Other than that the physical structure did not change so there was no additional cost. The robot's total cost would be less than $6000 to duplicate and cost the team less than $5000 due to donated parts. In addition to the monetary cost, more than 3000 man-hours were required to retrofit Optical Prime this year, in addition to the 3000 hours spent designing and fabricating it in 2005.

Component	Cost
Mechanical	
Steel Components	$245.75
Drive Motors	$478.13
Servos & Gears	$430.66
Wheels	$49.09
Fiberglass/Plexiglas	$171.34
Bearings	$32.51
Bolts/Fasteners	$200.15
Subtotal	$1,607.63
Sensors	
Cameras	$1,415.00
Digital Compass	$51.00

Accelerometer	$20.00
Optical Encoders	$158.60
Sonar	$425.70
Infrared	$140.04
GPS	$94.99
Subtotal	$2,305.33
Computing System	
Pentium 4 Processor	$158.00
Motherboard	$70.00
RAM	$134.25
Hard Drive	$78.97
Other Hardware	$167.72
Subtotal	$608.94
Electrical Systems	
Batteries	$153.54
Electronics	$559.81
Wire	$469.25
Connectors	$221.82
Subtotal	$1,404.42
Total	**$5,926.32**

12.0 Conclusion

A simple and elegant design was stressed this year due to the difficulties experienced in 2005. The control system was redesigned to be distributed and hierarchal, the model and planning algorithm were written to be more efficient and the computer vision system was re-implemented to be three times faster. This was added to the already stable frame, retrofitted with new motors, to create a robot which will perform well in the 2006 IGVC.

Appendix A: Team Roster

First Name	Last Name	Major	Grade Level
Robert	Adams	Mechanical Engineering and Biological Sciences	Freshman
Ryan	Arlitt	Mechanical Engineering	Freshman
Kent	Barnett	Electrical Engineering	Junior
Mike	Becker	Mechanical Engineering	Senior
Ben	Bethge	Physics	Junior
Dave	Brown	Computer Science and Electrical Engineering	Junior
Ryanne	Dolan	Computer Engineering and Computer Science	Junior
Paul	Drews	Electrical and Computer Engineering	Junior
Matt	Duncan	Mechanical Engineering	Sophomore
Chris	Eckhard	Computer Science	Senior
Ramaprasad	Eshwarahalli Lakshminarayana	Mechanical Engineering	Graduate Student
Kerry	Geisz	Mechanical Engineering	Senior
Steven	Gullen	Mechanical Engineering	Sophomore
Jia	Guo	Mechanical Engineering	Graduate Student
Venkatesh	Hariharan	Manufacturing	Graduate Student
Matt	Hawkins	Mechanical Engineering	Senior
Mike	Hibbeler	Mechanical Engineering	Senior
Igor	Izyumin	Electrical Engineering Computer Science	Junior
Chris	Jacobsen	Applied Mathematics	Senior
Daniel	Krus	Mechanical Engineering	Graduate Student
Joel	Logue	Electrical Engineering	Junior
Joe	Lombardo	Electrical Engineering	Junior
Amanda	Luellen	Electrical Engineering	Freshman
Matt	Marsh	Computer Engineering	Freshman
Ryan	Meuth	Computer Engineering	Graduate Student
Justin	Miller	Computer Science	Senior
Phillip	Ponzer	Computer Engineering	Junior
Casey	Porta	Mechanical Engineering	Junior
Jeff	Pretz	Electrical Engineering	Freshman
Paul	Robinette	Computer Engineering and Physics	Junior
Zachary	Royer	Materials	Junior
Stuart	Salvador	Mechanical Engineering	Freshman
Gerard	Sequeira	Electrical Engineering	Graduate Student
Chris	Solomon	Aerospace Engineering and Business Administration	Senior