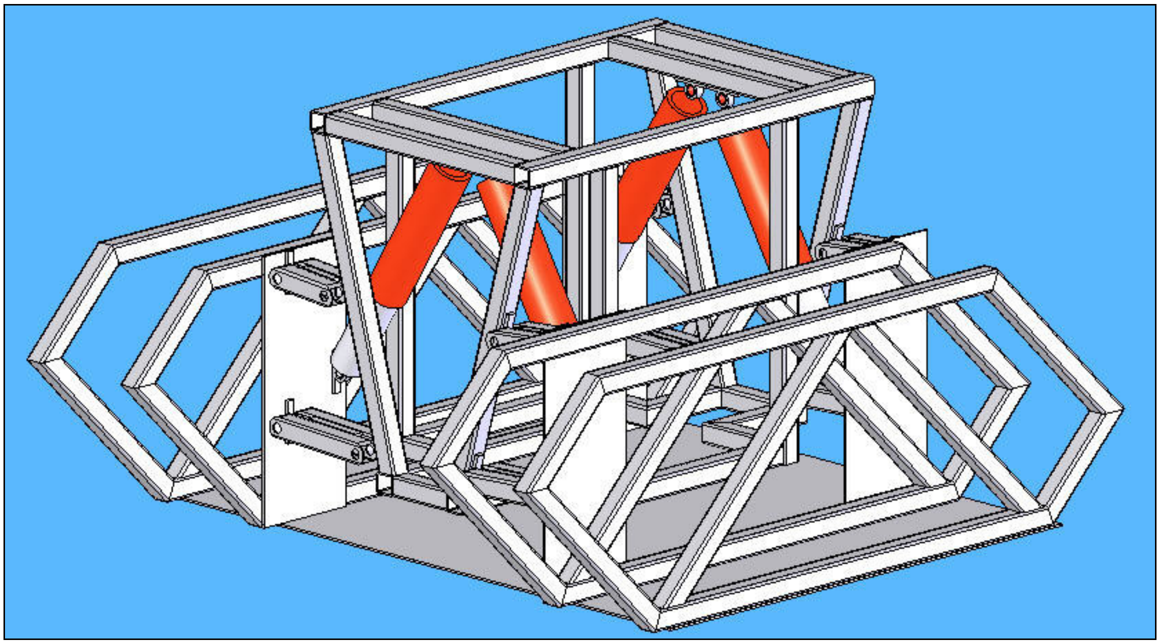
OVERLORD

Autonomous Vehicle Team

Multi-Disciplinary
Robotics Club



Rochester Institute of Technology

Alexander Sojda, Jason Stanislawski, Alan Meekins,
Ziyan Zhou, Joshua Karpoff,
Nick Valerio, Josh Watts, Jose Torres.

I, Dr. Daniel Phillips of the Electrical Engineering Department, Rochester Institute of Technology certify that the engineering design of the vehicle and systems by the current student team has been significant and equivalent to what might be awarded credit in a senior design course.

Signed,

Dr. Daniel Phillips
Advisor, RIT Multi-Disciplinary Robotics Club

TABLE OF CONTENTS

1. Introduction

The Rochester Institute of Technology Multi-Disciplinary Robotics Club (MDRC) is proud to present OVERLORD, our newly designed autonomous robotics platform. In addition to our newly designed robot, our team has created an extensible robot core (ERC), which coordinates sensor input and directs motor output. Within this environment, we have developed autonomous algorithms that will allow our robot to navigate the IGVC track.

2. Design Process

2.1 Design Methodology

Our team was committed to building a robot with value, modularity, robustness, and stability. Our team strove to follow the ideals of "Lean and Agile" development techniques; below are the seven principles that we used as our project's framework.

Eliminate Waste – As an extra-curricular activity with a limited budget – financially and in human resources, we needed to reduce waste in our projects. We did this by focusing on tasks that were decided to be necessary and important to the completion of the project. Our "litmus test" of whether it was important was whether it was needed to complete the competition [1].

Amplify Learning – One of the defining points of our robotics club is that we nourish learning. Our team followed a "try-it, test-it, fix-it" design cycle when developing. We have found that this approach worked because we were continuously learning about new algorithms and ways to make our design environment work more effectively and efficiently [1].

Decide as Late as Possible – Our software and hardware team both are small, and allow work to be done concurrently. This allows us to work in groups to reduce programming errors. On the mechanical work, it allows us to complete work on time and reduce errors. Important decisions were decided at meetings only after we were certain that we had enough information to responsibly make that decision.

Deliver as Fast as Possible - Our team works to release early and often. We have gone through three releases of our environment, ERC, to get to the present one. Our mechanical team has gone through numerous solid model revisions allowing input from the whole team each time. This has resulted in a robust, stable platform for our autonomous algorithms [1].

Empower the Team – Our team decisions, in a large part came from the regular team members, rather then the sub-group leaders or the project leader. This allowed the each team to work to the internal human resources available to each sub-group. Our group leaders serve the purpose of a guide, to provide focus, and expedite administrative work, rather then dictate every aspect of work [1].

Build Integrity In – Our software is developed to be modular, allowing for simple use, and for additional features to be added as needed; allowing the team to be agile and responsive. The mechanical team worked to create a robot that is easy to maintain, and repair [1].

See the Big Picture – Our team strived to keep in mind the whole team, and not just the interests of the sub-group. This was achieved through formal and informal meetings, emails, phone and other modes of communication [1].

2.2 Team Structure

Our Team was split into three components, the Mechanical, Software, and Electrical. Each of these subgroups was responsible for specific subsections of the robot; the group leader of each subgroup ensured that each group was able to fully integrate their section into the completed robot. The team structure is outlined in figure 1.

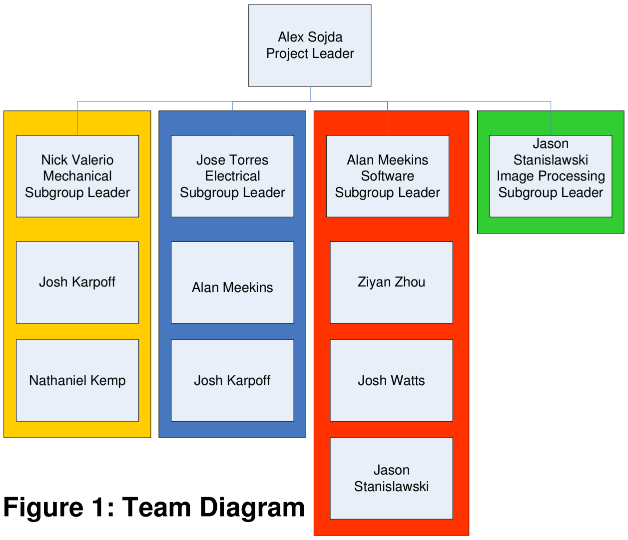


Figure 1: Team Diagram

3. Mechanical Design

3.1 Overview

The mechanical team was successful in meeting all design and functional requirements for the 2007 IGVC competition. Early in the brainstorming process several design and functional requirements were established.

Functional Requirements:

- Have ability to turn in place
- Be able to achieve 5 mph in 10 seconds or less
- Have ability to traverse rough terrain
- Conform to all rules outlined by IGVC

Design Requirements:

- Must fit through a standard 32” doorway
- Suspension geometry must produce little or no camber
- A suspension must produce sufficient vibration isolation
- All rotating shafts keyed as a point of failure
- Drive train must incorporate belts
- Use of a permanent magnet DC motor

3.2 Chassis

The Overlord chassis is constructed from 1” square structural steel tube. Approximately 110 feet of steel was required for the chassis construction. In addition to the 1” square steel, a 3’x 4’ 0.125” thick steel plate was used to fabricate all of the brackets and mounts. Both MIG and TIG welding was utilized wherever fasteners were not ideal and in areas where extra strength was required, such as in the idler gear brackets (Figure 2).

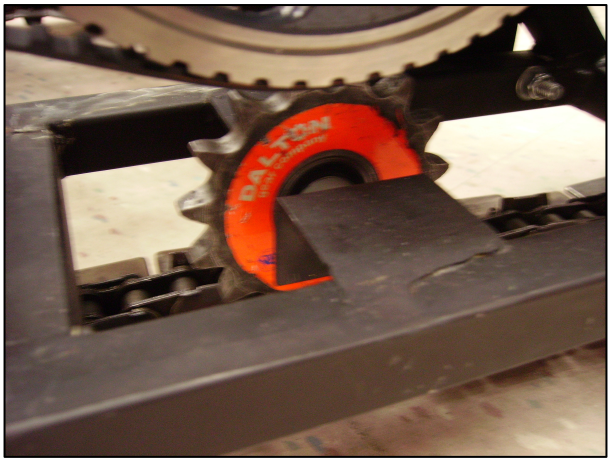


Figure 2: Close-up of idler gear bracket

The chassis design consists of two major components: the electronics box and the track modules. The track modules move independently of the frame on a four point suspension system. Separating the suspension from the track modules has several benefits. The first benefit of separating the track modules from the suspension is ease of track tensioning. Previous autonomous robotic efforts aimed to combine suspension and tracks into the same mechanical system. This was unsuccessful as track tensioning effectiveness became a function of position of the suspension within its travel. The new design suspends a rigid track with its own tensioning system. Greater control over track tensioning is achieved in this manner. The second benefit of track module and suspension separation is a lower center of gravity. Previous efforts placed heavy items such a batteries and motors above the center of gravity creating an imposing looking, but unstable robot. With the relatively light weight electronic compartment placed above the much heavier track modules containing all motors, sprockets, and steel tracks, the robot has a lower center of gravity, resulting in increase in stability. The third benefit of separating the track and suspension system is the total isolation of the mechanical components from the electrical components. This design approach improves the serviceability of the mechanical components. In previous efforts, it was necessary to fully remove electrical components to unbolt drive train components. Now, with the drive train in separate modules, it is possible to remove one or both track modules without removing electrical components. These design changes will make field repairs shorter and easier.

One design requirement of the robot is the elimination of vibration from the sensitive electronics. This is achieved through two mechanisms: rubber feet on the tracks and oil dampened struts. Prior to the installation of rubber feet on the tracks, the vibration from the mechanical components alone was great enough to loosen nylon secured nuts. Adding simple

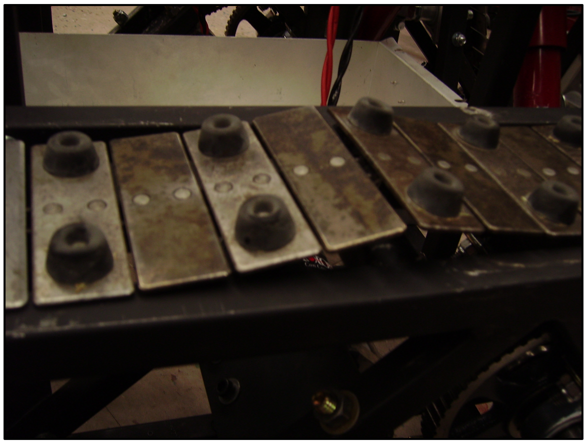


Figure 3: Detail image of rubber feet.

rubber pads eliminated this problem with less cost and complication than isolating each mechanical component individually. The rubber feet have been found to be effective in reducing shock resulting from the tracks impact with the ground. The purpose of the oil dampened struts is to maintain the robots degree of level while traversing rough terrain and limit shock from the electronics compartment induced by the drive train and terrain.

Suspension geometry was researched early in the design process. Different designs were weighted based on their manufacturability and effectiveness. A common McPherson style suspension was settled upon for its manufacturability and proven design. Other designs considered were aircraft landing gear style suspension in addition to some original designs.

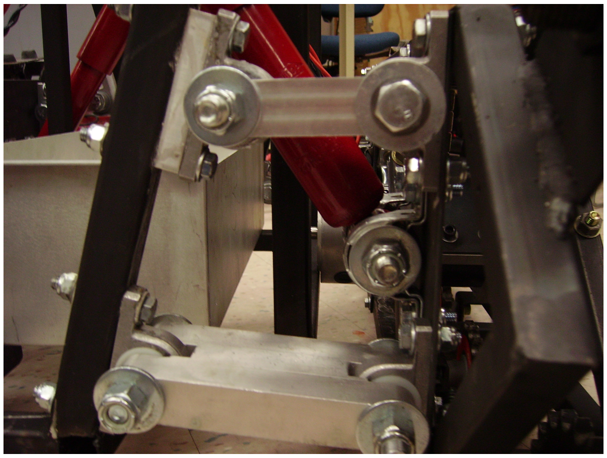


Figure 4: Detail image of suspension and suspension linkage.

3.3 Motors/Drive-Train

We use high torque, 4000rpm, 3.8HP MagMotors, to drive a 2 stage reduction synchronous belt drive train. The tracks consist of custom modified case hardened steel industrial blast furnace conveyor belts. The tread modules attach to the main compartment via an independent McPherson style suspension.

The double gear reduction utilized in the robot was designed around several factors before the drive train was fully realized. The team used prior knowledge to estimate the robots weight, friction coefficient, and known maximum speed within IGVC rules to design the drive train components.

In order to satisfy the needs of the software team's algorithms it was determined the robot needed the ability to turn in place. Previous robotic efforts were unsuccessful in rotating in place in an acceptable manner. Numerous existing tank designs were researched for their length to width ratios. Although the ratios were not all the same between the designs, consistently all tank geometries were longer than they were wide. The team settled upon a length to width ratio of 1.25 to 1 as an average of the researched designs.

4. Software Design

4.1 Overview

The software team worked to create a robust, modular, stable control environment in which to develop our algorithms. In addition to this, our software team also developed the algorithms that control the robot as it transverses the course.

4.2 Extensible Robot Core (ERC)

The environment that was developed for our autonomous algorithms is called Extensible Robot Core (ERC). This is a fully modular environment that allows components to be on different computers and communicate through TCP/IP. This feature allows distributed computing, adding additional value to the software.

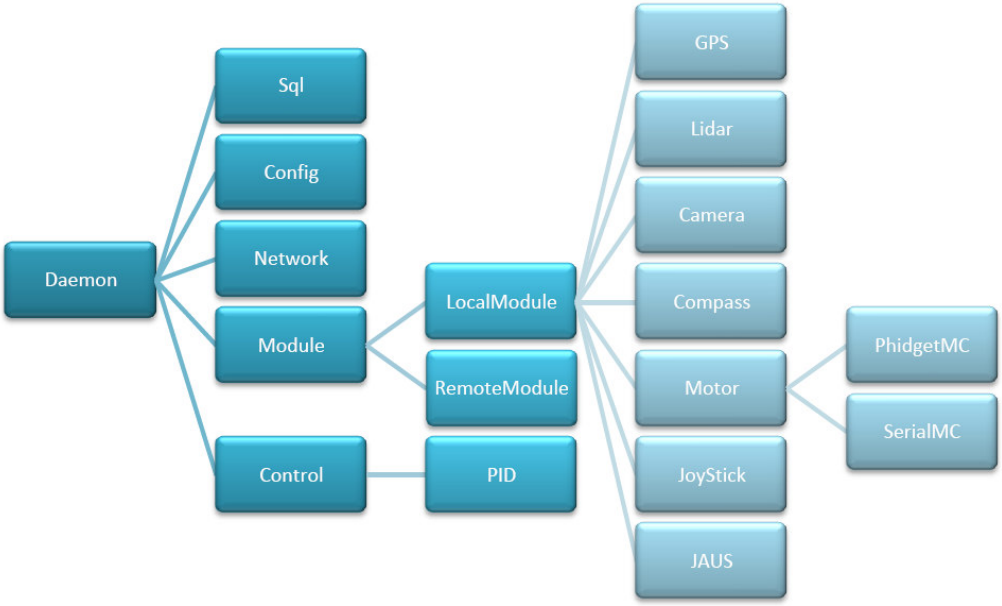


Figure 5: Block diagram of ERC.

4.2.1- LIDAR Module - To interface to the LIDAR, a custom API was written using LibSerial, a serial interface library written in C++ for Linux systems. An ERC module was then written using the API to interface it to our system.

4.2.2 – GPS Module – The GPS module is implemented using the open-source software GPSd. This is the pseudo-standard Linux interface for GPS. Using this approach, we are able to use any NEMA compatible GPS unit on the robot. The GPS module interfaces to GPSd using a TCP socket.

4.2.3 – Compass Module – The compass module is interfaced using the I^2C drivers built into the Linux Kernel. The sensor is polled on a continuous basis and data is the main sensor responsible for navigation.

4.2.4 – Image Processing Module – The Image processing module is used to detect the lines and ensure that the robot stays within them. This was achieved by applying a bilateral filter, then converting the image to grey-scale and using a Canny edge detector on the result. Lastly, a Hough transform was applied to certain regions of interest.

Figure 6: Original image for processing.

The purpose of a bilateral filter is to smooth the image and remove noise, but still preserve edges. The bilateral filter is an improvement on the Gaussian filter because it analyzes and filters on the spectral range as well as on the spatial domain.

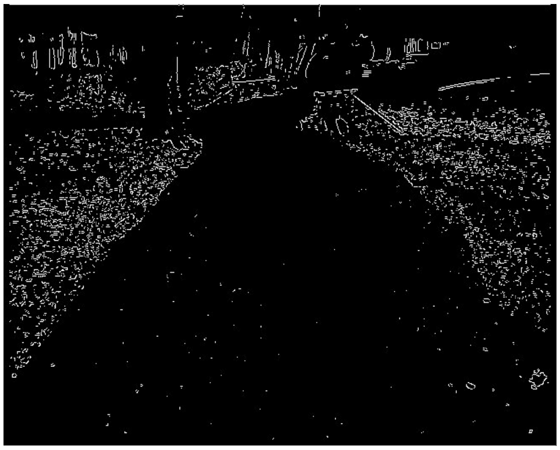


Figure 7: Image after canny edge detector and shift to grey scale.

Then, the image was converted to gray-scale and the canny edge detector was applied. (Figure 7) The Canny edge detector first uses Gaussian filtering to smooth the image. Next, the detector uses the Sobel operator to determine the spatial gradient of the image. From the spatial gradient, the edge directions are found. Then, the maximum values along each traced edge are kept while all others are set to zero to obtain thin edges. Finally, hysteresis is used to threshold the edges to obtain an unbroken edge.

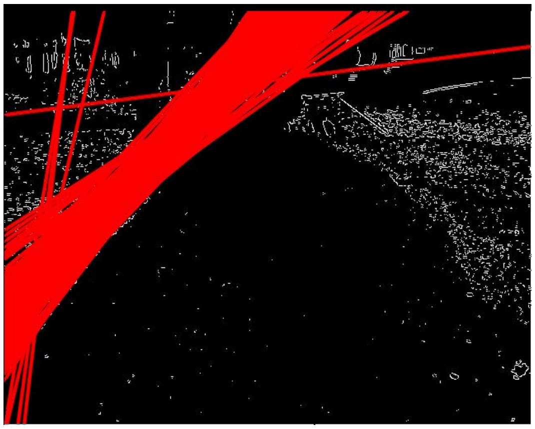


Figure 8: Hough transform on left side

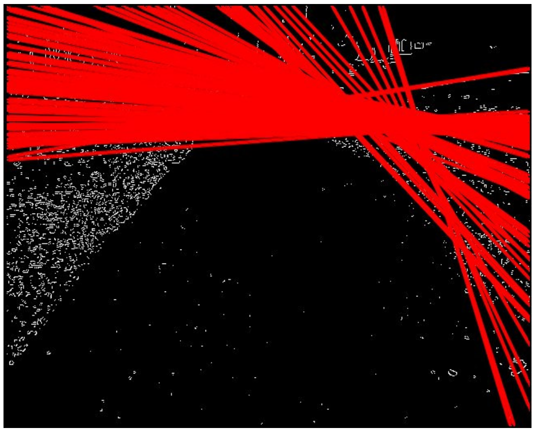


Figure 9: Hough transform on right side

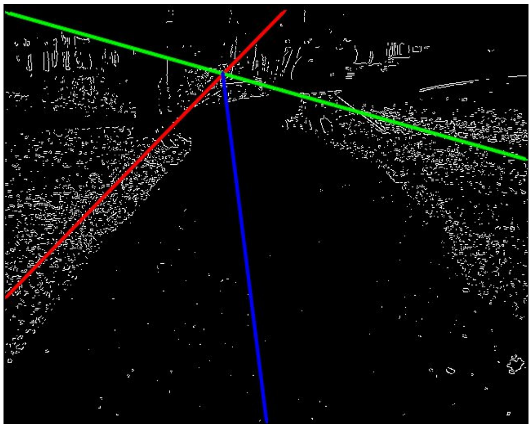


Figure 10: Processed image with calculated vector

The Hough transform was applied to find all the lines in both halves of the image.(Figures 8 & 9) The Hough transform weighs the points in the image to find the lines that fits the most points. Finally, the lines are averaged to find the dominant lines, and a vector to the center point is calculated if both lines are present (Figure 10). If one or both lines are found to be not present, then the path planner module is notified about the invalid data so it can take appropriate corrective action.

4.2.4 – JAUS Implementation – JAUS messaging system is built as a module in our control environment. Since ERC was designed based on our own network communication protocol, JAUS is treated as an additional sensor that constantly listens for incoming JAUS messages. The autonomous algorithm will pause or resume according to JAUS sensor status. On the hardware level, OVERLORD integrates IEEE 802.11g wireless networking.

4.3 Autonomous algorithms

The algorithms for this robot operate in a dynamic environment. Below are the specific strategies for each challenge of the competition.

4.3.1 - Autonomous Challenge – In the autonomous challenge the main sensors used will be the video camera and the LIDAR. The image processing module will return an "optimal vector". This vector is passed to a function that locates the best heading. A high-level diagram of the operation of the robot is shown in figure 11.

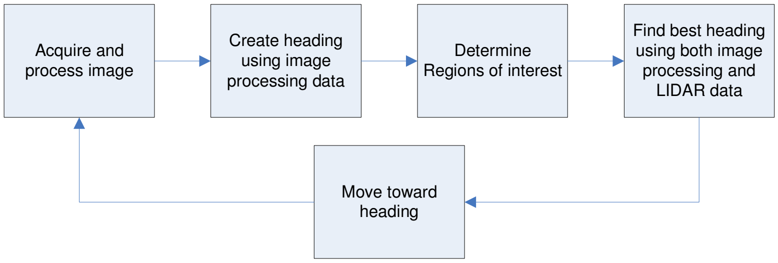


Figure 11: Flow diagram for autonomous challenge.

4.3.2 - Navigation Challenge – In the navigation challenge, the main sensors that will be used are the GPS and the LIDAR. The process for navigating to all GPS locations is outlined in figure 12.

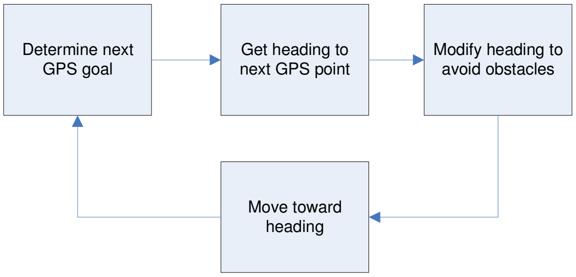


Figure 12: Flow diagram for navigation challenge.

5. Electrical Design

5.1 Overview

As with the rest of our robot, we strived to create a modular electrical system. This was done by using off-the-shelf components when available, and developing systems using standard interfaces such as USB and RS-232.

5.2 Sensors

5.2.1 LIDAR – Our robot is equipped with an LMS-291 SICK LIDAR. This unit allows the robot to detect obstacles in a 180 degree radius in front of the robot. The LIDAR is one of the most important sensors on our robot because of its precision and range.

5.2.2 GPS – The Global Positioning System (GPS) is used to determine the position of the robot. The robot uses an AgGPS132 unit with Omnistar differential corrections. With this capability, the position of the robot is known to within a meter. This system connects to the robot through a Linux daemon and a custom ERC module.

5.2.3 Digital Compass – The CMPS03 digital compass is connected to our motherboard via the onboard I^2C connection. This compass has an accuracy of 4 degrees. An I^2C 5V to 3.3V level shifter had to be fabricated in order to facilitate the signal difference. As a backup, and for testing our system on hardware that does not have onboard I^2C, we have developed a USB interface to the compass using the USB I2C/IO board by DevaSys.

5.2.4 Digital Video Camera – Our robot can be equipped with either a USB webcam or an IEEE-1394 (Firewire) camera. They both connect to our motherboard and have a standard Linux driver interface (video4linux). The images are then processed in our custom developed video processing ERC module.

5.3 Computer Systems - The robot is equipped with an ITX mini-motherboard with a 1 GHz processor and 1 GB of RAM. The unit is powered by a highefficiency DC-DC power supply; this extends the runtime of the robot and allows the use of lead acid batteries. Since there was concern about the reliability of using a traditional hard drive on an outdoor robot, a USB flash drive was used in the place of a hard drive. Using Debian Linux as an operating system gives our team the ability to leverage the use of many open-source projects in our favor.

5.4 Motor Control – The extremely high-current motors require the use of two IFI Victor-883's per motor. Each pair of Victor-883 motor controllers is interfaced to a port on a four port Trossen Phidget motor/servo controller. A custom motor controller using an ATmega168 was also developed, but is reserved for emergency use since the USB connectivity of the Phidget is more convenient and modular.

5.5 Emergency Stop Systems – As stated in the IGVC rules, our robot has two emergency stop mechanisms: a hardwired Emergency Stop (E-Stop), which is connected directly to a relay that controls power for the motors, and a wireless emergency stop. The wireless e-stop uses two dedicated ZigBee development kits from Freescale Semiconductor. Both systems are designed to be failsafe; if there is a power failure in any of the sections of the E-Stop circuit, the robot motors will power down.

6. Final Cost

Part	Vender	Part Number	Quantity	Total Cost	IGVC Cost
Frame materials	Metal Supermarket	1" Sq. Tube 1018 Steel, 14 Gauge.	110ft	$52	$52
Suspension Arms	Metal Supermarket	Aluminum Bar	6ft	$30	$30
DGPS	Trimble	AgGPS132	1	$500	-
LIDAR	SICK	LMS-291	1	$5000	-
Digital Compass	Acroname.com	CMPS03	1	$52	$52
Tracks	Custom	-	-	-	-
Pulleys, belts, bearings, shocks, bushings	Grainger, McMaster-Carr	-	-	$800	$800
Motors	Robot Marketplace	MagMotor C40-300	2	$634	$634
ITX Motherboard	EPIA	MII	1	$200	$200
Electronics, Connectors	Goldcrest Electronics	-	-	$40	$40
USB 4GB Flash stick	Newegg.com	-	1	$49	$49
Total				**$7357**	**$ 1887**

7. Conclusion

We are confident that we value engineered a robust, quality robot that meets our design and functional requirements. This system should provide a solid platform to achieve a high standing at the 2007 Intelligent Ground Vehicle Competition. We look forward to demonstrating our robots capabilities; OVERLORD is a truly impressive robot.

8. Acknowledgments

Multi-Disciplinary Robotics Club (additional members):
Jeremy Abel, Mark McKann, Kevin Egan, Mike Egan, Stewart Wadsworth, April Diak, Chris Kaczmarek, Sean Croteau, Mike DeFrancis, Nathan Pendleton, Zach Harvey, David Longo, Iheanyi Umez-Eronini, Daniel Raugalis, John Floren, Patricia Heneka, Chris Armenio.

Sponsors:
Impact Technologies, Kate Gleason College of Engineering, RIT Student Government, Goldcrest Electronics, Seabreeze Amusement Park, Leonard Alvin, Esq., Rochester Steel Treating Works, Mahany Welding Supply.

Faculty Advisors:
Dr. Daniel Phillips, Dr. Ferat Sahin.

9. References

[1] Poppendieck, Mary, and Tom Poppendieck. <u>Lean Software Development, An Agile Toolkit.</u>, 2003.