EMBRY-RIDDLE
Aeronautical University
ROBOTICS ASSOCIATION

MOLLEBot

Embry-Riddle Aeronautical University's Modular Light-weight, Load-carrying Equipment Bot

Statement of Effort: I certify that the engineering design of the vehicle described in this report, MOLLEBot, has been significant and equivalent to the effort required in a senior design project. Areas of modification include, but are not limited to, vehicle chassis, sensors and mounting, software design and construction, and electrical implementation.

Charles F. Reinholtz,
Professor and Chairman,
Mechanical Engineering Department,
Embry-Riddle Aeronautical University

5/10/2011

Table of Contents

Table of Figures

1. Introduction

MOLLE is an acronym for “Modular Lightweight Load-bearing Equipment,” coined by the military for their infantry backpack. The typical MOLLE pack is a versatile polymer pack frame that can be augmented with nylon bags and totes of various sizes, depending on the equipment being carried. The MOLLE pack shares many conceptual characteristics with the ideal portable intelligent ground vehicle: it should be lightweight, customizable, and durable. MOLLEBot is a simple, robust, and safe ground vehicle, while conforming to IGVC rules and exceeding customer expectations.

MOLLEBot was designed to be an inherently small, lightweight, back packable vehicle. This represents a substantially different design paradigm compared to approaches used to develop previous entries in the Intelligent Ground Vehicle Competition (IGVC). Having a footprint of just over three feet in length and two feet wide, MOLLEBot has the smallest platform (the shape viewed from above) allowed by competition rules. Weighing less than forty pounds, MOLLEBot is dynamically nimble and easy to transport. Focusing on a compact, lightweight design resulted in a vehicle that is intrinsically safer and easier to operate than larger vehicles. Compact size and light weight generally equate to lower costs. More importantly, minimizing vehicle weight results in more compact, less dangerous drive motors, smaller battery packs with less on-board energy storage, and lower kinetic energy in operation. It is clear from the IGVC rules that safety is the primary concern of the judges, sponsors and organizers of the competition, hence developing a functional vehicle of minimum size and weight should be a fundamental object of the student design teams. In addition to its new mechanical design, MOLLEBot also has a sophisticated software and electrical system vastly different from previous years. New solid state hardware and a robust and highly cohesive software system will make MOLLEBot an effective competitor in the 2011 IGVC.

2. Innovations

MOLLEBot is a fundamentally innovative platform in that it breaks the long running paradigm of large and complex vehicles entered into IGVC. While our commitment to innovation should be apparent throughout this report, this section will highlight a few of the unique innovative aspects of our design.

2.1. Light-Weight, Modular Vehicle

From years of experience with other, larger robots, the team decided that the optimal solution was to aim for a more compact and lighter weight vehicle. Size and weight of the vehicle directly affect utility and dynamic performance. MOLLEBot can fit in the trunk of most modern cars, making it easily portable and versatile in use. MOLLEBot was designed such that it complies with Mil-std-1472F. This standard defines the maximum weight for an object the size of MOLLEBot to be 44 pounds. Weighing in at less than 40 pounds, MOLLEBot complies with this standard and thus more than 95% of people can lift MOLLEBot to a height of 3 feet. A compact platform also allows for the greatest latitude in planning and executing autonomous maneuvers and in solving the configuration space problem as the robot translates and rotates. Stated simply, the smaller the piano, the easier it is to solve the piano-mover's problem. Given the complex, tortuous maze of obstacles present in recent IGVC competitions, it is not adequate to plan paths by assuming a single point (zero-sized) vehicle translating in a 2-dimensional plane.

2.2. Unit Body Frame

The previous iteration of MOLLEBot used a frame and box design. The frame structure of the vehicle took up a large amount of space, and the electronics compartment was formed and attached based on the constraints of the structural frame. In this iteration we adopted a monocoque construction design, also called unit body construction, from modern automobile design techniques. The structure of the vehicle is now the external walls of our electronics compartment instead of being internal to the frame. By doing this, we have more than doubled our internal capacity for electronics and other components. This monocoque design not only increases internal space and maintains structural integrity, but it is also lighter than the original frame.

2.3. PRO Mast: Single-Unit, Mast-Mounted Sensor Suite and E-stop

A complex aspect of designing a platform like MOLLEBot is to account for mounting of various sensors along with the required emergency stop button and the safety light. It is logical and common practice to mount the GPS antenna, camera, compass, and emergency stop onto a single mast. This allows the GPS antenna a clear view of the sky, it provides the best perspective view for the camera, it removes the compass from the magnetic field generated by the drive motors and it gives an elevated location for the emergency stop button. Like most vehicles in competition, MOLLEBot also uses a scanning laser range finder for exteroception. Most vehicles mount this rangefinder near the front of the vehicle and low to the ground to give an unobstructed field of view. Because of MOLLEBot's size and remarkably low profile, we were able to mount the rangefinder on the same mast as the other sensors. By mounting the laser range finder on the mast, MOLLEBot has a single mount for the entire sensor suite. All sensors can be removed at once by removing the PRO Mast (Perception, Recognition, and Observation). This design makes MOLLEBot easy to break down for transport. It also helps to facilitate efficient wire runs and rain proofing of the vehicle electronics.

Figure 1: MOLLEBot Fully Assembled

2.4. Modular, Highly Cohesive Software

The MOLLEBot software system is simple in its design and elegant in its implementation. The goal of the design was to produce a highly cohesive software system with low coupling. The implementation of asynchronous message passing aided in making this software system simple and incredibly robust. Software was broken down into modules such that each sensor, actuator, and control loop had its own executable file. Using asynchronous messages, these software modules broadcast messages to other modules without being directly coupled. Since there are strictly typed message sets for each module, a software module can be easily modified, removed, or completely replaced without any changes in the other modules.

3. Electrical Design

MOLLEBot uses sensors that have become commonplace in the IGVC, including a SICK LMS 151 scanning laser range finder, a NovAtel GPS, a Sparton SP3004D compass, and a Minoru stereovision camera. A comprehensive list of components is provided in section 5.1, Component Costs. Since many of the other vehicles at competition use similar sensor packages, this report will focus more on the integration of the sensors and the design of the supporting electrical system.

The electrical system is often the most complex part of an autonomous system, and has the highest concentration of potential failure points. For this reason, the team spent substantial time working to design and document the electrical system of MOLLEBot before implementing it in hardware. Every detail of the electrical system must coorespond to a design requirement . If the details do not line up into a solution that meets the design requirements, changes must be made, either to the requirements or the system.

3.1. Selection of Operating Voltage

One place where the design did not meet the requirements was in the selection of an operating voltage. The best voltage for a majority of MOLLEBot's electrical components was 12V. However, the motors can operate any any voltage between 9 and 24, and each voltage corresponds to different torque output. At 12V the motors would barely be capable of moving the vehicle over the ramp in competition, and would be insufficient to propel the vehicle over the ramp with the payload attached. Therefore, the design had to be modified for an appropriate operating voltage to power the motors. The design was updated to operate of a 24V power source. The 24V power source enabled the vehicle to accommodate the power requirements of the motors and laser range finder and also eliminated the need for a boost converter to provide 19V to the Xi3 computer hardware.

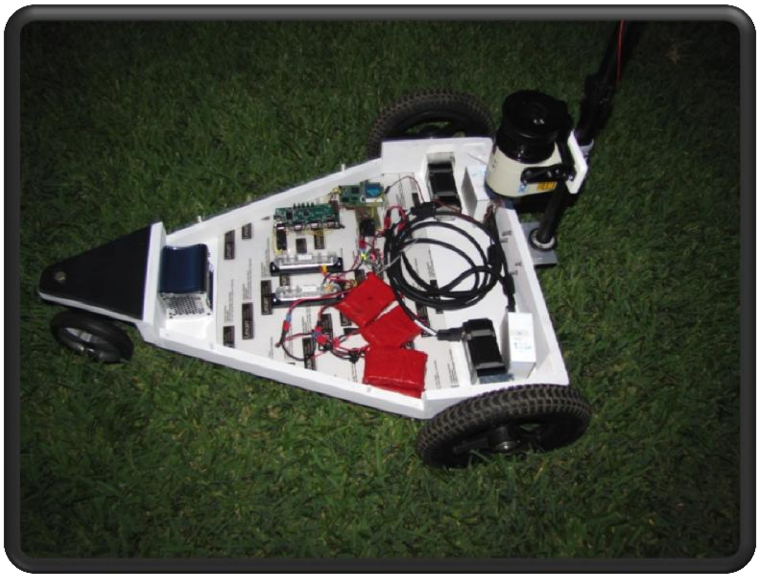

Figure 2: MOLLEBot's Operational Electrical System

3.2. Simplification of Motor Interface

MOLLEBot used RS232 serial lines to communicate to the motors in the previous iteration of the vehicle. Now, for simplicity and safety, the remote control solution has been integrated into the chain of command for the motors. In **Figure 3**: Primary Electrical System, the National Instruments Data Acquisition device is used to control the motors over an analog interface rather than a serial line. This allows RC data and autonomous signals to be transmitted over the same line, rather than requiring both a serial line and an analog line going to the motors.

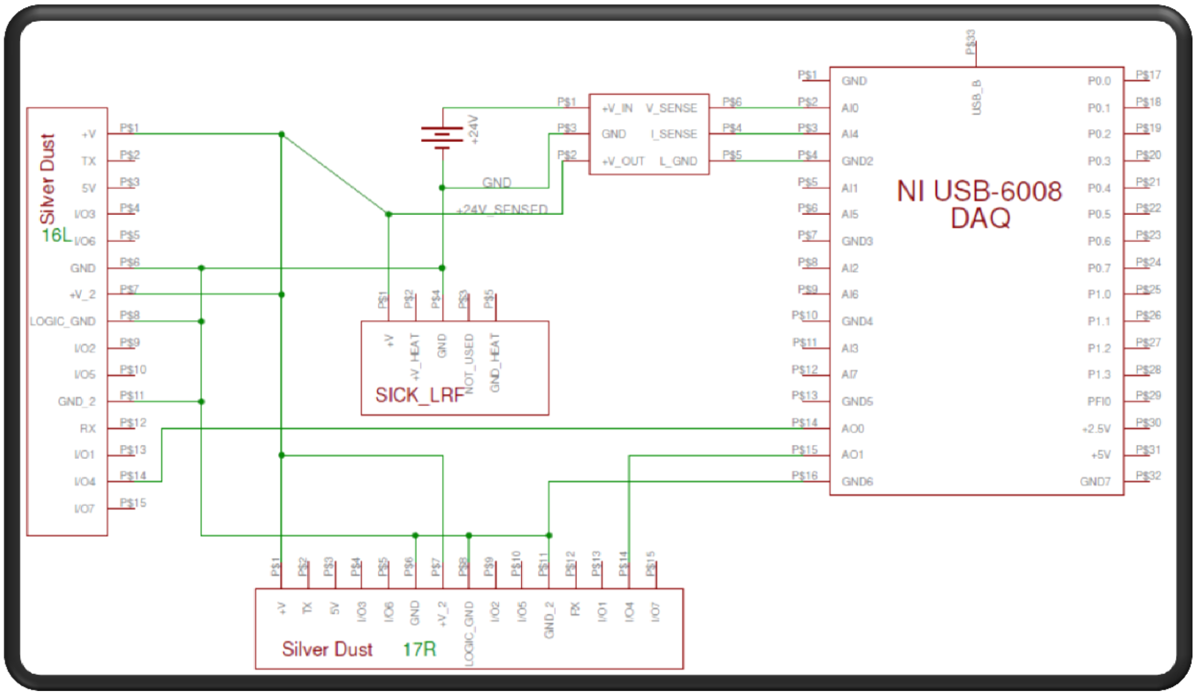

Figure 3: Primary Electrical System

3.3. Battery Type

During the design phase of the project, the team looked at multiple battery chemistries for use on MOLLEBot. Lithium Polymer (LiPo) batteries have the best energy density, which is important for a light weight vehicle like MOLLEBot. However, LiPo’s can be potentially dangerous if not handled properly. For this reason, the team selected A123 Nano phosphate Lithium Ion battery cells due to their safety and similar energy density to LiPo’s. Thanks to a donation from DeWalt our team was able to construct custom battery packs, seen in **Figure 4**: Custom A123 Battery Packs, from A123 cells to get the 24V operation voltage discussed above.

Figure 4: Custom A123 Battery Packs

4. Software Design

MOLLEBot had a group of five Software Engineering students dedicated to designing the software system from the ground up. By writing a formal software requirements specification and ending with a full implementation of the design, the software team followed the full life cycle of the software.

Many of The Robotics Association's industry partners develop software in C++ under a Linux based operating system (OS). Use of C++ and Linux seems to be prevalent in the autonomous systems industry. Given this preference, the team decided to undertake the MOLLEBot software development using the same development environment. MOLLEBot's onboard computer uses the Ubuntu OS and the g++ compiler to operate efficiently and effectively. MOLLEBot also utilizes the Robot Operating System (ROS), a meta-operating system that provides functionality common to robotic applications and OpenCV, the open source computer vision library.

4.1. Software Architecture

MOLLEBot was designed to have a modular software system that is easily modified. As such, the software could be used for many years to come, in the changing competition environment, without a significant overhaul of the software. The software was broken down into independent modules that could be written and operated completely independently of each other. The initial design of modules is shown in **Figure 5**.

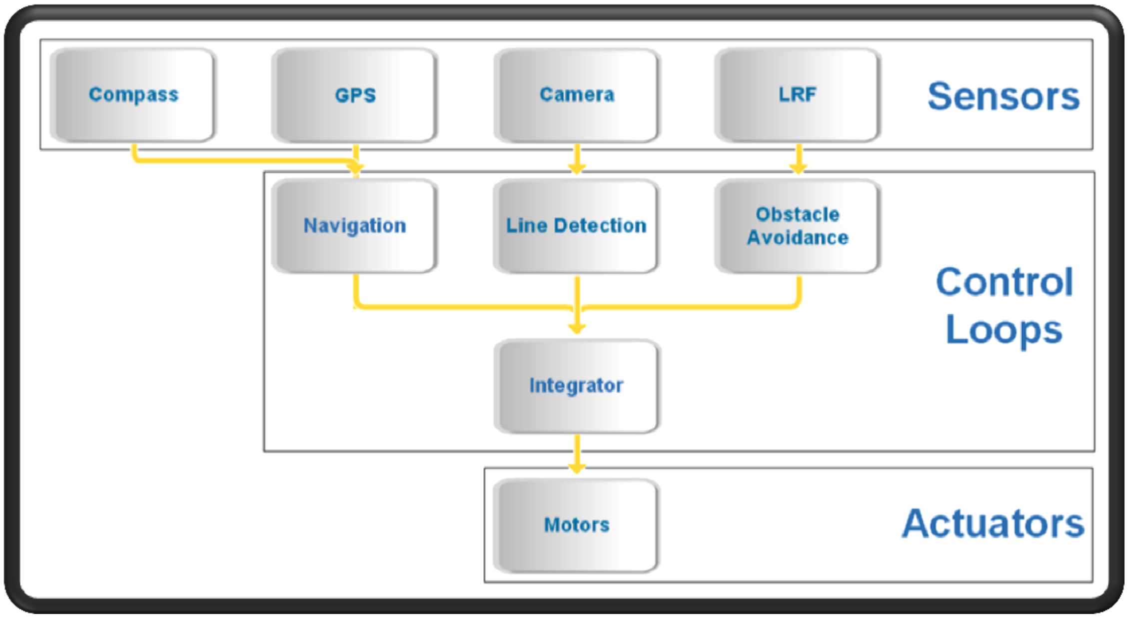

Figure 5: Software Modules

Each sensor has a software executable that will run independently of all other software in the system. When new data is received by the module, it will broadcast, in a strictly typed message, what was received over a private communication channel to a control loop module that will then process the data. This makes it easy to modify or replace the communication to sensors because it does not affect any of the other software. A control loop module is an event based system. Upon receipt of a message, the module will execute once, updating necessary information and broadcasting it, and then sleep until another message is ready. It will then calculate a vector to its desired direction. This vector is then passed to the integrator, which is also event based, like a control loop, but instead of calculating a vector it will calculate motor speeds and broadcast the commands accordingly.

As previously discussed, MOLLEBot's software is optimized for high cohesion and low coupling. Using asynchronous message passing, provided by ROS, each module will act as if there is nothing on the receiving end of sent messages. Coupling of the software is reduced because the software modules are not aware of each other; communication is simply accomplished through continuous broadcast of messages. This also makes it possible for a developer to write an entire module without knowing about the rest of the software, just understanding the interfaces which are strictly typed and well defined.

4.2. Mapping Technique

Various types of mapping algorithms, such as occupancy grids and simultaneous localization and mapping (SLAM) algorithms, have been tried on student projects at Embry-Riddle. The MOLLEBot team decided that sophisticated mapping algorithms, while useful, were not necessary for the competition. Focusing more on robust local mapping algorithms, MOLLEBot builds a highly localized map that only contains obstacles and data from acquisitions within the last few seconds. This local map is updated so that obstacles that disappear from the field of view are assumed to move backwards over time, thus having less of an effect on the integrator algorithm, but still contributing so that the rear of the vehicle does not hit the obstacle. The path of the vehicle around an obstacle resumes its initial path and therefore looks like the diagram in **Figure 6**.

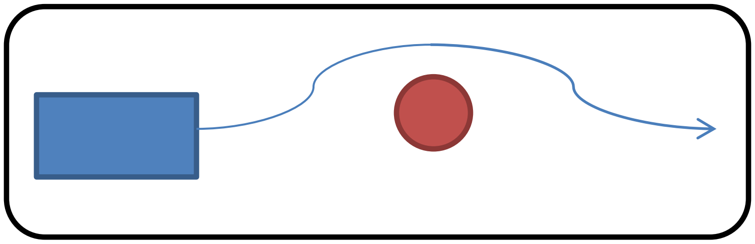

Figure 6: Vehicle Path around Obstacle

4.3. Lane Following

In the Line Detection module, MOLLEBot uses the simple brightest pixel algorithm, shown in **Figure 7**, for detecting lines and identifies a single point on the line using ground plane interpolation of the pixel coordinate.

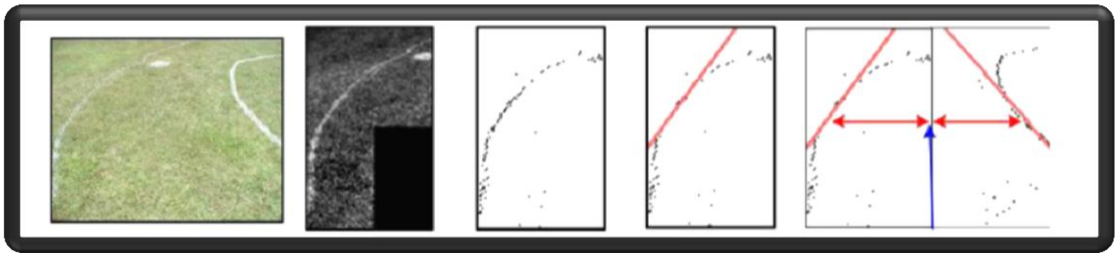

Figure 7: Line Detection

Once the lines are detected, a single point on each line is selected at a specified distance from the vehicle. These two points then become virtual obstacles in the software. The Line Detection module broadcasts a vector that points directly between the two points identified. An example of the point identification is shown in **Figure 8**.

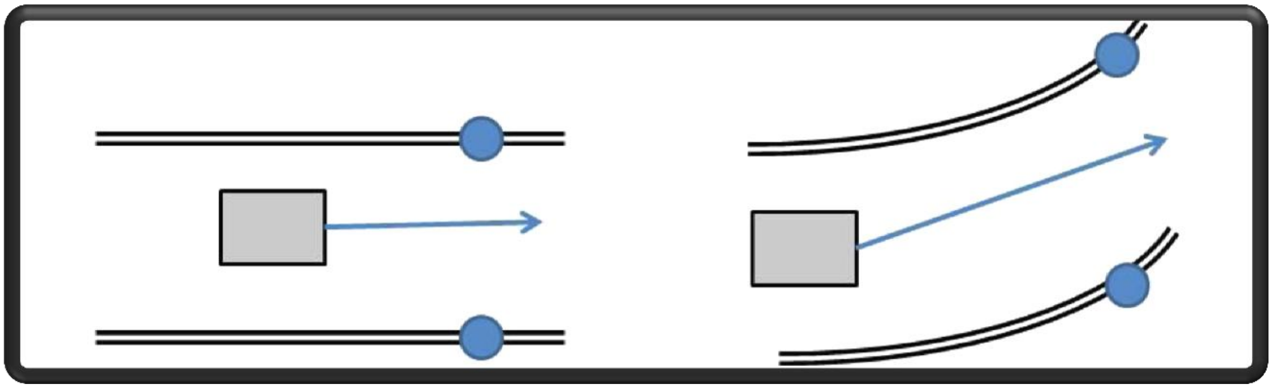

Figure 8: Vector Generation from Line Detection

4.4. Obstacle Detection

The Obstacle Avoidance module takes data in from a scanning laser range finder as an array of vectors. MOLLEBot selects a subset of these vectors that are closest to the vehicle, the most hazardous to the vehicles current location and calculates the shortest path away from these obstacles.

4.5. Waypoint Navigation

The Navigation module keeps track of the waypoints that the vehicle is required to navigate through. This module identifies its position and heading relative to true north from data that it receives from the GPS and compass broadcasts. Using this information and the recorded information about the waypoint path, MOLLEBot calculates the distance and angle to the waypoint. The module then broadcasts this information for receipt by the Integrator.

4.6. Data Integrator

Once any of the control loop modules have completed their tasks and sent their data over the broadcast channel, the Integrator module will read the data and update the motor values accordingly. To do this, the Integrator module is broken down into sections that are cascading in sequence.

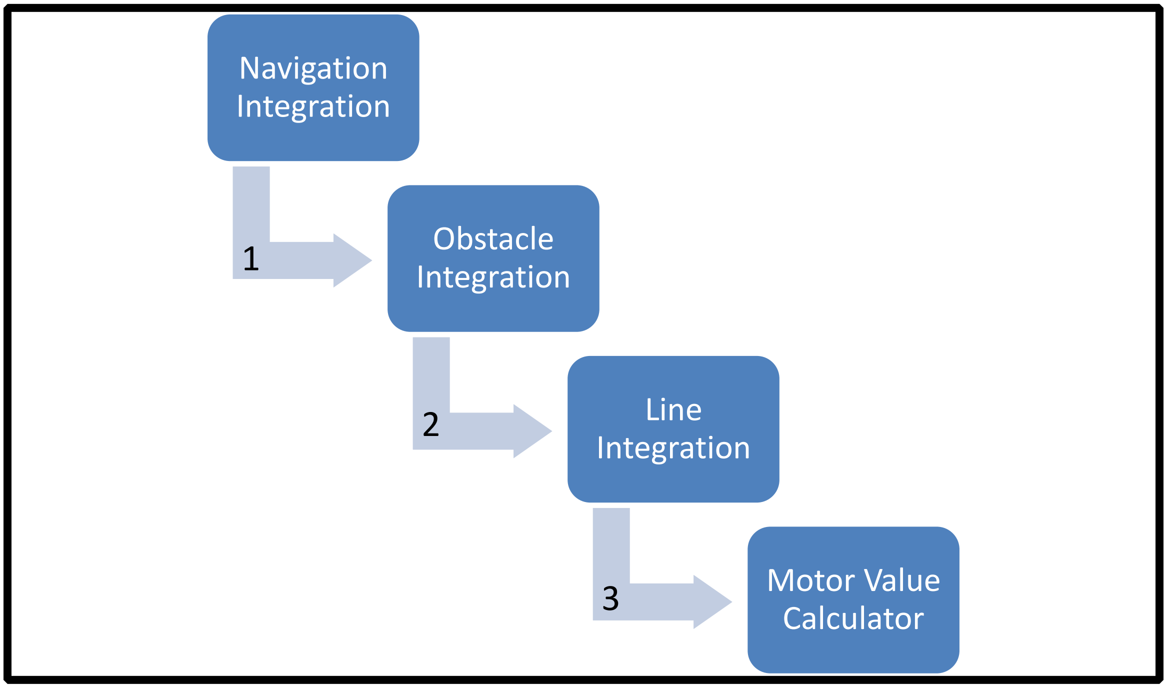

Figure 9: Waterfall Sequence

If the Line Detection module completes an execution of its control loop and broadcasts a new vector before the Obstacle Detection or Navigation modules finish, then the Integrator will only execute the Line Integration function, making the assumption that the vector produced by transition 2 is still valid from the last execution of the Integrator. This flow of logic is shown in **Figure 9**.

5. System Integration

The challenge posed by the Intelligent Ground Vehicle Competition does not pose a problem that can be solved through Mechanical Engineering, Software Engineering, or Electrical Engineering practices. The problem is fundamentally one of Systems Engineering. No modern robotic vehicle can be successful without careful consideration given to the systems integration. Each team member needs to be cognizant of how their decisions will affect the design and engineering approach of the other team members.

Positive team dynamics helped the group find a unified solution to almost any problem. An example of this would be when the electrical design called for an Arduino microcontroller in the RC system, but the software was designed for a NI DAQ device. The two groups came to a compromise by incorporating a multiplexor to switch control without interfering with each other.

5.1. Component Costs

Due to the sponsors of our organization, MOLLEBot was able to be constructed at minimal cost to the team. **Figure 10** shows a list of components that were used in MOLLEBot, with the cost to the team and estimated market value.

Component	Cost to Team	Market Value
A123 Battery Packs (4 Cell)	Donated by DeWalt	$108.00
Wooden Frame	Donated by Dr. Reinholtz	$80.00
Caster Wheel	$15.00	$15.00
Keyspan Serial to USB	$114.00	$114.00
Xi3 Modular Computer	Donated by Xi3	$800.00
Sensor Mast	$82.00	$82.00
NovAtel GPS	$5,000.00	$5,000.00
Quicksilver Motors	$1,550.00	$2,200.00
SICK LMS151	Donated by Michael Coleman	$7,632.90
Skyway Wheels	$60.00	$120.00
Sparton SP3004D Digital Compass	Donated by Sparton	$700.00
Minoru Stereovision Camera	$50.00	$50.00
Wires and Misc.	$200.00	$200.00
TOTAL	$7071.00	$17,101.90

Figure 10: Component Costs

6. Safety and Robustness

Safety is a primary concern at all times while working on MOLLEBot, or any robotic project in the robotics lab. Laboratory rules were strictly enforced for any student participating on the MOLLEBot team. In order to join the team each member must watch a safety training video that explains how to safely operate equipment in the lab, how to handle an emergency, and general safety rules. Safety goggles were worn throughout the build process, except for during software construction.

6.1. Emergency Stop Functionality

The hard-wired electronic emergency stop button is located on the mast at the rear of the vehicle. This button disengages a relay, cutting power to the motors and rapidly stopping the vehicle. The remote control receiver and processor provide a wireless estop that can be engaged at any time.

6.2. Remote Controller

The remote control system used in MOLLEBot enforces that the controller must be on and in range for operation of the vehicle to continue. If the signal from the controller is disrupted the system goes to a state where motors are continually commanded to zero movement. This will prevent the vehicle from rolling if on a slope, as well as, prevent the computer from commanding new motor values. The wireless receiver will operate up to .25 miles away. Should the vehicle loose the wireless connection with the controller, it will immediately go into emergency stop mode.

6.3. Reliability

MOLLEBot uses many commercial, off-the-shelf components and hardware. As this was a systems engineering problem, these components fit well into the design and provide a high level of reliability. Each item was carefully integrated with voltage regulators, power switches, and mounting hardware. With different voltage levels inside MOLLEBot, the electrical system was implemented such that no device could be plugged into the wrong voltage source. Each different voltage level has a different connector that will not interface with the others. This ensures that anyone can set MOLLEBot up and it will run the same way that it has every time in the past.

7. Design Process

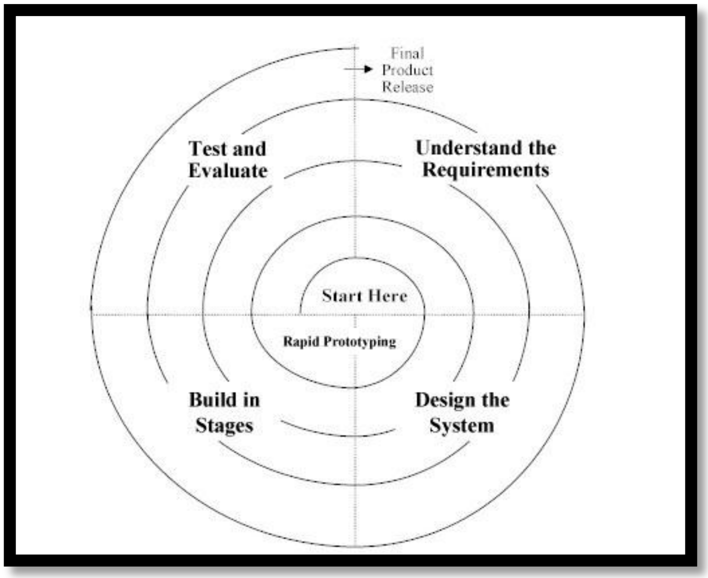

Figure 11: Spiral Design Process

The team followed a true systems engineering process for creating MOLLEBot, going through three iterations of development using the Spiral Method, **Figure 11**, for designing and implementing the system.

This development cycle is highly effective for any systems engineer. Starting with rapid prototyping, engineers can get a firm understanding of what is feasible. Moving into developing requirements, followed by design, implementation and testing, the engineers gain a solid understanding of the system and develop a prototype by the end of iteration one. After this is done once, the development cycle starts again in a new iteration by revising the requirements and design. In all following iterations the system is expanded and improved. This process ensures a quality product at the end of the development cycle.

MOLLEBot was developed with a bottom-up approach, starting with hardware interfaces and moving towards intelligence throughout the development cycle. Since sensor integration was a constant in any development cycle or iteration of requirements, the team implemented the sensor software layer as part of the rapid prototyping phase, as well as, thoroughly designing the electrical system.

In the first iteration of development, the build phase was when a majority of the vehicle was built and sensors mounted. The control loops layer of software was designed and the interfaces to the sensor layer were defined during the design phase. The build phase included integrating the three lower levels of control loops into their respective sensors and the testing phase was primarily testing the communication interface between sensor and control loop layers.

The second iteration requirements and design phases were focused on improving the control loops layer. The build phase consisted of refining and tuning the software control loops layer. In the third design iteration, the integration and motion control modules were designed and interfaces defined. Once the integration layer and motion control layers were written they were tested for communication with the control loops layer and full vehicle testing followed.

8. Team Composition

The development of MOLLEBot's system required a multidisciplinary engineering team that consisted of everything from freshman to graduate students. The nine team members, listed below, put more than 2,000 person-hours into the design, manufacturing, and implementation of MOLLEBot. While our team was made up of a specific group of students that met at weekly meetings, there is large group of students and faculty that provided continuous support for the MOLLEBot project. Without them, the creation of MOLLEBot would not have been possible. Student team members are listed in alphabetical order and the contribution column is the primary area of concentration in development. A student's contributions are not limited to the single area listed below.

Name	Degree	Class	Contribution
Randy Breingan	Software Engineering	Junior	Software
Katrina Corley	Mechanical Engineering	Graduate	Mechanical
Catherine Cruz-Agosto	Software Engineering	Sophomore	Software
Gregg Leonard	Mechanical Engineering	Junior	Mechanical
Lin Lin	Mechanical Engineering	Senior	Mechanical
Christopher McKinley	Computer Engineering	Junior	Electrical
Alaric Payne	Computer Science	Sophomore	Software
Jameson Pietrowski	Aerospace Engineering	Freshman	Software
Christopher Sammet	Mechanical Engineering	Senior	Electrical
Matthew Standifer	Mechanical Engineering	Freshman	Electrical

Figure 12: Team Members and Contributions

9. Conclusion

In a class of its own, MOLLEBot is breaking down the preconceived notion that autonomous ground vehicles need to be large, heavy and operationally complex. Designed through a spiral systems engineering process, MOLLEBot is a simple, robust, and elegant solution to the problem posed by the IGVC. The practicality of an autonomous vehicle that weighs less than fifty pounds is enormous; this vehicle can scout indoors as well as outdoors and access many environments which larger vehicles cannot. For these reasons, and many others, MOLLEBot will not only be a major competitor in this year's competition, but this prototype platform is the future of autonomous ground vehicles.

MOLLEBot, inherently compact and lightweight, is a substantially different design paradigm compared to approaches used to develop previous entries in the Intelligent Ground Vehicle Competition (IGVC). By minimizing vehicle weight, MOLLEBot is compact, with less dangerous drive motors, with smaller battery packs with less on-board energy storage, and lower kinetic energy in operation. With these innovative modifications, MOLLEBot will be an effective competitor in the 2011 IGVC.