|  |
| --- |
| **Rose-Hulman Institute of Technology** |
| **2012 IGVC Design Report** |
| **Moxom’s Master** |
|  |
| Ander Solorzano;Ruffin White;Kyle Green;Michael Pauly;Trent Tabor |
|  |

Rose Hulman Robotics Team CM 5000

5500 Wabash Avenue

Terre Haute, IN 47803

@ Team PiC?

This report and the described vehicle were designed and constructed by the Rose-Hulman Robotics Team during the 2011-2012 school year.

|  |  |
| --- | --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Dr. David M. Mutchler (Advisor) | Date |

|  |
| --- |
|  |

# Introduction

The RHIT Robotics Team comprises of undergraduate students from Rose-Hulman Institute of Technology in Terre Haute, Indiana. Named after a short story by Ambrose Bierce, this year’s design entry is an improved version of Moxom's Master. It reflects our growing experience with the IGVC, having attended the 2008, 2009, 2010, and 2011 competitions. This report documents the design decisions and rationale for this year’s vehicle design.

# Team Structure and Design Planning Process

## Describing the team’s hierarchal structure and organization

This year, IGVC design team is structured into two sub-teams, the mechanical team and the electrical team. The sub-teams focus on the hardware, electrical, and software components of the robot by accomplishing various tasks from design concept to finalized product. To accomplish their objective, each of the sub-teams holds meetings two to three times a week. Overseeing the sub-teams’ functions, the team’s president is in charge and responsible for setting goals and milestones for the project, contacting sponsors, and managing the administrative tasks of the team. Once a week, the two sub-teams get together in an all-team meeting, presided over by the president, to give updates on the project’s current conditions. These weekly updates include informing the team about accomplished tasks, setting future goals, and occasionally reporting minor setbacks. Aside from the president, the team also has several other officers in charge of running the team’s administrative and financial tasks. Refer to Figure 2.1 for the team’s overall structure.

|  |
| --- |
|  |
| Figure 2.1: The figure above demonstrates how the team is structure and organized into its two IGVC sub-teams with a team leader assigned to each sub-team and the administrative branches of the officers. |

## Envisioning the overall design approach

From the beginning, our team established that the design entry must be modular for any further redesign or last minute changes, robust so that the robot is able to operate on off-road terrain or handle unexpected conditions, intelligent so that the robot can correctly react to its environment, mobile so that the platform can handle tight turns with minimal effort, and safe for operation.

This year, the team accomplished more objectives in ten weeks than in the past three years. Instead of using open-source and General Public License components, the team decided that the software compatibilities should determine the hardware and electrical components present in the making of the design. This change was taken to facilitate the communication between various components and the computer, and to allow the use of more reliable sensors and more accurate devices. Finally, we concluded that the design process should consist of creating the mechanical infrastructure that would support the entire system, install and configure all the necessary electrical devices, and to create the software system that would integrate the path following, obstacle avoidance, line detection, and GPS navigation algorithms.

# Mechanical Systems Report

The primary goal of the mechanical sub-team was to build and maintain a safe, durable, reliable platform on which electrical and software tasks could be executed. Moxom’s Master incorporates many components of previous years’ designs, but several elements have been improved, with the primary objective of improving the robot’s mobility. Our approach to the improvement of our design was inspired by the product development process outlined by Ulrich and Eppinger in *Product Design and Development* which ensures a structured and consistent process. Conceptual designs and redesigns are engendered through the process of brainstorming and evaluated with considerations to the criteria of safety, cost , ease of construction, and sturdiness, as well as the needs of electrical an software aspects.

Currently, the frame sits upon a steering system comprised of two rear drive wheels and a caster situated in the front. The chassis contains two cases for the storage of electrical components, and also includes a rear mast for mounting sensors and housing external interfaces. Continued testing showed that the mechanical system had to be redesigned to improve the platform’s durability, stability, and modularity.

## Increasing mechanical durability for extended runs

There have been substantial improvements upon the previous design in regards to durability and reduction in the risk of mechanical failure. To protect against corrosion, all steel parts are now stainless or painted. Three main sources of mechanical failure were revealed during testing and operation. The first source of failure was the loosening of bolts in the drive train assembly. To minimize this problem, lock washers and Locktite now secure all bolts in the drive train assembly. Secondly, the chains connecting the encoders to the drive wheels would commonly come dislodge. This issue was lessened by the installation of adjustable encoder mounts, designed to maintain chain tension. The third source of mechanical failure was the shearing in the wheel hubs due the substantial weight of the robot; the team’s solution is discussed in greater detail later in Section 3.5

## Restructuring the frame for improved stability

The robot originally had a 4 wheel skid steer drive.  While this was an extremely stable setup, it was also unnecessarily heavy and restricted mobility, and the high center of gravity caused the robot to lurch upon stopping. This year, the robot’s design was modified to be a balance between the stability and mobility.  By replacing the mast with lighter angled aluminum, the team reduced weight by 15 lbs. and moved the robot’s center of gravity lower to the ground. Additionally, the removal of the internal battery charger shed an additional 35 lbs. from the robot. The final weight saving measure was replacing the two front wheels with a single castor wheel, decreasing the weight of the robot by about 30 lbs. with minimal losses to stability. To minimize vibration for electrical components, the robot remains on a suspension system which is formed by two springs mounted between the wheels and the chassis. Additionally, the choice was made to use a pneumatic castor, which would absorb some of the forward jolts.

## Adding modularity to the infrastructure

Moxom’s Master is designed to be modular to allow for easy installation of new parts and the ability to modify the robot with minimal downtime. The team used T-slot aluminum for the main frame of the robot because of the easy-to-use prefabricated tabs and brackets for mounting. Minimizing the weight necessary to support the few components in the mast while facilitating possible installation of external devices, we selected to use bolted angle aluminum for the rear mast as a weight-saving measure. All of the plastic panels are easily removable to provide quick access to the robot’s internal components. The on-board computer and many of the other electronic components are mounted in sliding server drawers that can be removed from the robot for replacement or repair. The tower mast provides the electrical sub-team a place for installing additional electrical sensors and external computer hardware like keyboards, a USB hub, and a computer monitor.

## Redesigning mechanical features following observations and analysis

The mechanical team originally chose tractor tires and wheels because they were inexpensive and readily available. After several months of use, one hub failed catastrophically, and two other hubs had deformed to the point of breaking their paint and neared failure. The team already had spare wheel hubs available and wanted to use these to replace the broken hubs.

The proposed redesign involved welding an additional doughnut-shaped plate to the hub to reinforce the joint. This effectively makes the portion of the hub structure that is under the greatest stress three times thicker. Additionally, the weld provides a fillet with a greater radius than the existing design, reducing the stress concentrations at the joint where failure occurred.

|  |  |
| --- | --- |
|  |  |
| Figure 3.1: Stresses at joint for original hub (left) and redesign (right)  Increasing the base thickness and fillet radius greatly reduced shear stress. | |

As shown in Figure 3.1, analysis reveals a reveal a large stress concentration around the center of the hub where failure occurred on the actual part. Red indicates an area of expected failure, with a maximum equivalent stress at the joint of 116,000 psi. Next, the analysis was repeated for the redesigned hub. As can be seen in Figure 3.1, the stresses around the hub are greatly reduced. The maximum stress on the redesigned hub is 32,000 psi, which results in a safety factor of 2.8.

As mentioned previously, a new castor wheel was installed to replace the front two drive wheels. Doing so resulted in significant gains in maneuverability, due to the elimination of skid steering. The switch in steering type also reduced stress in the hubs, removing the outward friction force caused by turning, an inherent aspect of skid steering. Losses in speed due to the loss of the forward drive motors were negligible, as the elimination of a large amount of weight from the front shifted the robot’s center of gravity rearward. This caused greater normal force against the rear wheels, allowing the drive motors to operate more effectively.

# Electrical Systems Report

The electrical team decided to follow a strategy that focused on coming up with a design concept that would meet all required objectives, build the robot based on the design, run various performance evaluation runs, and redesign or restructure components in need of reconsideration. For this year’s entry, the electrical team made significant adjustments to last year’s design, improving the overall performance and reliability of the electrical systems.

The team decided that to meet all goals and objectives for the competition, the robot’s design must meet several characteristics. The robot’s electrical structure must be modular to allow the team to replace components or reposition some components with ease. The robot’s electrical components must be robust and sturdy and capable of handling off-road terrain. The components must be reliable, user-friendly, and rarely experience serious or minor malfunctions.

## Improved electrical objective to meet design concept

From an electrical standpoint, the team decided that the robot must contain circuit protectors to prevent components from burning in case of an unexpected high current spike. A power supply should deliver power to the motors and electrical components of the robot individually. The robot must be able to determine the direction of travel, the speed of travel, and intelligently decide when to stop with minor problems or malfunctions. Using an array of several sensors, the robot should determine white lines that indicate the boundaries of the track, its current position and current operator mode, and allow the team to read the state of various critical components via some feedback. The robot must also be able to be safely shut off via wireless and wired communication in case any unexpected problems arise.

The electrical team proceeded to incorporate the electrical components after the robot’s mechanical frame was constructed. To achieve the team’s overall objective, the electrical team split into several individual groups that would accomplish a specific task.

## Reusing certain features of the original design

Moxom’s Master derives its power from two 12-volt, 60AH Power-Sonic sealed lead acid batteries connected in parallel. To achieve a high-level of modularity and adaptability, the computer and on-board electronics are mounted in 3U rack-mount server cases.

|  |  |
| --- | --- |
| C:\Users\solorzaa\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.Outlook\28316QXF\Figure1ElectricalLayout.png | C:\Users\solorzaa\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.Outlook\28316QXF\Figure2ElectricalLayout.png |
| Figure 4.1: The flowchart on the left shows how the power was distributed to the motor and electronics in the original design entry. The figure on the right shows how several electrical components received power. | |

Mounted in one of the server cases, the team uses an Intel i7 860S for the processor that delivers a power budget of 65W TDP. The operating system and control software are installed on a 60 GB SSD which help reduce power consumption and eliminates the effect of moving parts.

Mounted in another of the server cases includes a 12V 30A battery charger, an AC-DC power supply, a main battery breaker, and a main cutoff switch as shown in Figure 3.1.

Previously mounted in another of the server cases, four Texas Instruments Black Jaguar motor controllers provided control to the motors of Moxom’s Master. The motors are controlled via a CAN bus over an integrated CAN to RS-232 bridge. Optical quadrature encoders attached to the wheels help provide a read of the rotational speed and direction that the wheels are turning and allow the robot to control and adjust the speed of the motors. If a problem or malfunction is observed, the communication to the motors can be safely interrupted via a handheld 100ms relay-controlled kill switch that transmits packets to an Arduino microcontroller mounted on the robot.

Previously, the robot also contained an array of several sensors and a computer monitor that is located along a vertical mast situated in the back of the robot. The sensors included a Garmin GPS 18LVC OEM connected via RS-232 interface, a SICK LMS 291 LIDAR, a 5-megapixel Elphel 353 camera, and the master relay-controlled kill switch.

## Redesigning electrical components to meet design constraints

The team performed several tests runs on the original design to observe the performance of the robot. Critical areas of concern were turning ability, turning speed, power consumption, average travel speed, responsiveness to the kill switch, overall performance of electrical components, and the structural stability.

Due to the lack of performance and reliability from the original design, the team re-organized and re-evaluated some areas that decreased the overall performance of the robot. After evaluating the robot’s performance with respect to the areas of concern, the team concluded that several improvements could be made to improve speed and performance, such as decreasing weight and buying new, more reliable components.

Overall Electrical Component Weight Displacement Restructure

Since weight played a major role towards the maneuverability of the robot, the team decided to redesign the robot by removing one of the server cases that included the battery charger, a DC-DC 160 W converter, and the main power switch. The power switch was moved to one of the two other server cases while the battery charger unit and the DC-DC converter were allocated to an external and portable case. The team redesigned and reconstructed the external battery charger case in a manner that would facilitate access and ease of operation.

|  |  |
| --- | --- |
| C:\Users\solorzaa\Desktop\IMG_0579.JPG | C:\Users\solorzaa\Desktop\IMG_0578.JPG |
| Figure 4.2: This figure shows the external battery charger case when removed from the robot unit (left) and when attached to the robot to provide battery charge (right). | |

Power Supply and Distribution Restructure

To increase performance, and help reduce the overall weight, the team considered replacing the original power supply of the robot. Several options were considered and some tested, however these tests showed that the current design is the best fit for the robot currently.

|  |  |
| --- | --- |
| C:\Users\solorzaa\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.Outlook\28316QXF\Figure1ElectricalLayout.png | C:\Users\solorzaa\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.Outlook\28316QXF\Figure2ElectricalLayout.png |
| Figure 4.3: The figure on the left shows the improved power case with the charger removed from power tether case. The figure on the right shows the new power distribution to various sensors mounted on the robot. | |

Motor Controller Restructure

Another important factor reviewed and changed to improve the performance of the robot consisted of changing the motor controller. Previously, the robot used 4 TI Black Jaguar Stellaris Motor Control Units (MDL-BDC24) to control the turning speed and direction of each of the four wheels previously mounted. After the replacement of the two front wheels with a single caster wheel, the electrical team decided to remove the TI motor controllers and replace them with the RoboteQ Motor Controller MDC 2250. This is a two channel, high power motor controller, which is more reliable and has better documentation than the Black Jaguars.

|  |
| --- |
| C:\Users\solorzaa\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.Outlook\28316QXF\Capture.PNG |
| Figure 4.4: Motor controller tests showing wheels changing direction due to the direction of current flow. |

The objective of this change was to add reliability, accuracy, and performance to our motor controllers. By replacing the previous motor controllers with the new model, the team managed to achieve improved communication between the motor controller and the computer via RS232 to USB communication.

Emergency Shut-Off Switch Restructure

After numerous tests, the team observed that the Arduino kill switch that was previously used lost reliability due to signal interruption or unaccounted power fluctuations. As a result, the team replaced the entire emergency shut-off switch with a long range wireless 2 channel relay box that responds well at distances of 150 feet as long as the robot is in the line of sight.

The team connected the relay to the digital I/O pins available on the new motor controller so the motor controller correctly and safely cuts the power to the motor. The handheld emergency switch device was also redesigned to make a more robust and reliable device than the previous design. After continued testing of the new safety features, the robot constantly responds well.

Improving the Vision and Object Detection Capabilities

After careful reconsideration, the team concluded to swap the 5-megapixel Elphel camera with a Logitech HD 1080p webcam that connects via USB. Since the camera will only detect traffic barrels and white lines, we decided to sacrifice unnecessary high resolution in order to consume less processing time.

Lastly, the team also swapped the 24-volt SICK LIDAR sensor with a 12-volt Hokuyo UTM30LX LIDAR sensor. Aside from decreasing the power consumption, the new LIDAR also increased the robot’s scan angle to 270o with a 30 meter range. The improved scan angle gives the robot more peripheral vision, allowing the robot to see what is on its side without turning.

Sensors that Provide Travel Direction and Waypoint-to-Waypoint Travel

This year, the team managed to incorporate the IMU, or Inertial Measurement Unit, that allows the robot to read its current angle of travel, due to the digital compass feature in the IMU. In addition, the team also installed the NAVCOM SF-2050 GPS that allows us to acquire the robot’s position within a 50-cm radius as long as the robot is in an open-field and the WAAS feature is enabled. The WAAS feature allows the GPS to improve accuracy by measuring small variations due to the geometry of earth, the ionospheric conditions, and electromagnetic disturbances. By analyzing the variations, the GPS allows us to acquire highly accurate data.

# NI Logohttp://www.ni.com/images/coreblock/large/lvlogo_vert.gifSoftware Systems Report

Using National Instruments' LabVIEW programming environment, our team developed a robotic platform implementing the basic principles of sense, think, and act. The general implementation of these principles involved performing every process of sense-think-act in parallel. Using this method is more advantageous and contrary to traditional sequential structures due to the fact that many robotic applications involve small and mobile computational architecture, relatively lengthy computation times, slow sensor refresh rates, and mission-critical safety requirements. By observing the everyday behavior of living creatures, the characteristics of sense-think-act are performed in parallel as well, whether it involves the subject as simplistic is a fruit fly, involving electromagnetic sensing of light, image and range recognition of obstacles, under-actuated control and in-flight stabilization. All of these amazing processes and wonders of biology are elements of controls and logic which we pursued to imitate with a form intelligent autonomous locomotion.

## Top-level module instantiates variables, initializes sensors, and flows performs needed actions

Using a top level module to encapsulate the subsequent subprocesses, we're able to structure the startup initialization, continuous operation, and shutdown procedures. Upon startup, our code first initializes all the sensor acquisition loops, which include the GPS, LIDAR, IMU, and motor telemetry acquisition loops. Every sensor loop runs independently and continuously while acquiring and writing sensor data to share global variables. This allows for different acquisition rates that are dependent on each sensor type. For instance, the GPS roughly updates once every second, while the LIDAR data updates roughly at 50 Hz. This enables for each sensor to operate at its maximum frequency, allowing for the environmental sensor database to be updated as periodically possible. All sensor interfaces (except the vision) communicate with computer platform through virtual serial RS-232 ports emulated by USB adapters. These ports are opened and configured using LabVIEW's VISA tool.

|  |
| --- |
|  |
| Figure 5.1: Shown left is the front panel view of the top level module showing various tabs that the user can select to switch between sensor outputs and operator modes. |

## Describing the “sense” loop logic

The LIDAR sensor loop continuously broadcasts polar coordinates that contain the angle and magnitude of object surfaces that intersect the LIDAR's plane of sight. The GPS sensor loop acquires a longitude and latitude along with additional information such as altitude and absolute time. Using a known location approximation we can improve the startup time for the GPS by setting a close estimate of the GPS's physical location. The IMU uses a combination of accelerometers, magnetic Hall Effect sensors, and Kalman filters to continuously report its own orientation in space with respect to the Earth's magnetic field. We can then use this as a magnetic North for our compass in order to guide our robot to a command heading.

Once the entire sense network is initialized, the top level module continues to start up the logic VI, or "think" portion of the code. This VI takes in the current sensor data along with the current set mode that defines the robot's behavior, which include a “tele-op” mode for remote operation and drive control with external Wii mote or any other handheld remote and an autonomous mode for intelligent course navigation.

|  |
| --- |
|  |
| Figure 5.2: Start up procedure that initializes each loop. |
|  |
| Figure 5.3: An example of our simple LIDAR acquisition loop. |

   

Figure 5.3: This figure shows all the key sensors used for sensing the environment around the robot. These include (from left to right) the Hokuyo LIDAR, the NAVCOM GPS, the IMU microstrain, and the Logitech webcam.

## Describing the “think” loop logic

While the LIDAR data is operating in sync, the vision loop captures the white lines and stores an array of several line vectors that contain start and end points. We transpose the line vectors on top of the LIDAR distant readings to create a new obstacle histogram. Using our current GPS location and the location of the next GPS waypoint, we use spherical coordinates to calculate the current distance and directional bearing to the waypoint. Using the obstacle histogram and desired bearing, we constructed a simple bearing controlled algorithm that attempts to align the robot towards a suitable opening within the histogram field. An opening is determined “suitable” when the distance and angle of the opening produce an absolute opening wider than the width of the robot’s wheelbase plus tolerances. While searching for suitable openings the robot will also implement a cost function based on its current difference in sub-goal heading and current heading along with a current distance. If the robot veers too far from the sub-goal heading or the current distance exceeds the specified maximum tolerance, the robot will proceed to orbit around looking for alternative paths that might provide a more direct path.

## Describing the “act” loop logic

Lastly, the motor control loops are initialized, thus implementing the "act" portion of the code. From the command bearing and command velocity specified by the logic module, the control loop attempts to drive the robot in the set direction. We calculate the specific motor velocities or RPM values by defining the drivetrain model matching our robot's own differential drivetrain. All relevant physical dimensions are specified within the model such as wheel radius, wheelbase width, and gear ratios along with clockwise and counterclockwise motor orientation. By using the system controller and implementing an integrator feedback loop, we take our command heading that serves as a set point and our current heading from the IMU that serves as the control system output. Using a basic integrator method we adjust the robots drivetrain set angular velocity to orientate the robot towards its command heading.

|  |
| --- |
|  |
|  |
|  |
| Figure 5.4: This figure shows the PID controller in charge for adjusting the turning speed as the robot turns to a given heading (top). The motor loop showing the telemetry data that user can use to monitor the state of the robot (medium). The path planning VI that allows the robot to intelligently navigate the course by transposing the IMU data, the LIDAR data, and the GPS data. |

## Incorporating the vision challenges

LabVIEW’s vision module has a VI which can collect images from the camera, process them, and focus on certain patterns. Even though the device is capable of higher resolutions, we collect the images at 640x480 pixels to decrease the time needed to process the images via the Logitech webcam installed on the mast of the robot.

The team decided to run a perspective transform to allow the camera to detect the distances of the white lines. Using a grid with known distances, we generated the transformation using a tool built into LabVIEW. Once we acquire an image, we processed the image so that we can read an array of multiple lines that can be easily transposed to our mapping algorithm.

|  |  |
| --- | --- |
| C:\Users\greenkt\Dropbox\RHRT LabVIEW\vision stuff\Images\img46.png | C:\Users\solorzaa\Dropbox\RHRT LabVIEW\vision stuff\Images\img13538.png |
| Figure 5.5: The figure on the left shows the method used for calibrating the camera. The figure on the right shows a snapshot of the webcam during the test runs. | |
| PLACEHOLDER FIG (get me the black and white pic with points Trent) | |
| Figure 5.6: This figure shows the field of lines that the robot sees after it processes the picture. The lines are broken down into several lines containing several start points and end points. | |

When this transformation is applied to an image on the course, the image appears to be viewed from straight above, and distances on the ground can be determined. A color threshold is placed on the image to find the white lines. These areas are then dilated and eroded to remove specks while also filling up gaps in the line. A line detection algorithm then finds the lines. These lines are returned in real world coordinates, due to the perspective transformation.

# Final Remarks and Setting Future Goals

# 