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

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

I. Introduction

The RHIT Robotics Team comprises of undergraduate students from Rose-Hulman Institute of Technology in Terre Haute, Indiana. Named after a chess-playing robot automaton depicted in a short story by Ambrose Bierce, this year's design entry is an improved version of Moxom's Master. The goal of Moxom's Master is to intelligently navigate through an obstacle course while avoiding obstacles, staying between white lines, and traveling to several waypoints. This report documents the design decisions and rationale for this year's vehicle design $10,698 robotic design as described in appendix I.

II. Team Structure and Design Planning Process

2.1. Describing the team's hierarchal structure and organization

The IGVC design team is structured into two sub-teams, the mechanical team and the electrical team. This year, the mechanical team is tasked with the hardware components while the electrical team installs the electrical systems and also the creation of the software algorithms. The sub-teams developed a series of Gant charts to help them meet the deadlines and objectives on time. To accomplish their objectives, each of the sub-teams held meetings two to three times a week each for three hours with an additional two-hour all-team meeting each Sunday. This resulted in roughly 2200 accumulated hours of work for the whole school year. During the all-team meetings, the sub-teams discuss the accomplished tasks, set future goals, and occasionally report minor setbacks. Aside from the president, the team also has several other officers in charge of running the team's administrative and financial tasks.

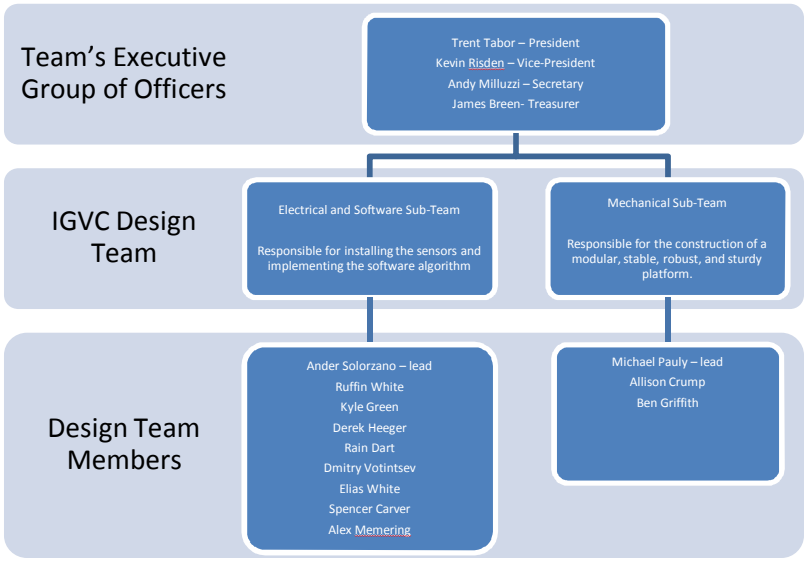

Figure 2.1: The figure above demonstrates how the team is structure and organized into its two IGVC sub-teams with a team leader assigned to each sub-team and the administrative branches of the officers.

2.2. 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 past 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. We also 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.

III. Mechanical Systems Report

3.1. Improving the mechanical systems by meet the design constraints

The primary goal of the mechanical sub-team was to build and maintain a safe, durable, reliable platform on which the electrical and software could successfully execute their tasks. 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 the other sub-teams.

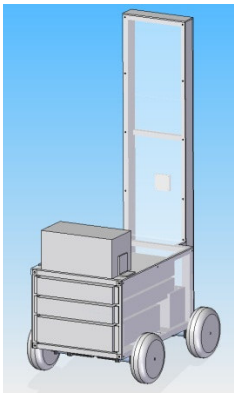

Figure 3.1: The figure on the left shows the original design frame of the robot with four wheels and three server rack cases. The figure on the right shows the actual frame of the robot with a caster wheel in place and two server racks.

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.

3.2. Increasing mechanical durability for extended runs

To add increased protection against corrosion, all steel parts are now stainless or painted. Three main sources of mechanical failure were revealed during later testing. The first source 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. We resolved this issue by lessening by the installation of adjustable encoder mounts, designed to maintain chain tension. The third source of 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

3.3. Minimizing weight while maximizing mobility

The robot originally had a 4 wheel skid steer drive. Though stable, it was also heavy and restricted mobility. Replacing the 2 front wheels with a single caster wheel decreased the weight of the robot by about 30 lbs. while increasing the mobility. By replacing the mast with lighter aluminum, the team reduced weight by 15 lbs. while moving the robot's center of gravity lower to the ground. To decrease vibration for electrical components, the robot remains on a suspension system which is formed by 2 springs mounted between the wheels and the chassis. The team decided to use a pneumatic castor which would absorb some of the forward jolts.

3.4. Adding modularity to the infrastructure

The modularity of Moxom allowed 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 computer peripherals such as a keyboard, a USB hub, and a computer monitor.

3.5. 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.

A proposed redesign involves welding an additional doughnut-shaped plate to the hub. 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.

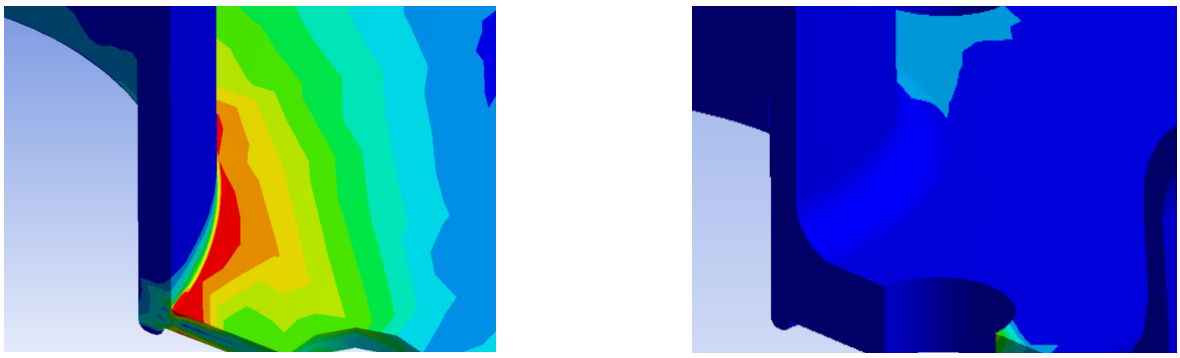

Figure 3.2: Stresses at joint for original hub (left) and redesign (right).

Increasing the base thickness and fillet radius greatly reduced shear stress thus reducing the chance of failure for the hub.

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.

Additionally, significant steps were taken to reduce the weight of the robot. As the rear sensor mast's size was defined by the by the location of the sensors, rather than high loads, it was an ideal place to start. The T-slot uprights used were excessive, given the minimal weight of the electronic components in the mast. Thus, the robot's mast was rebuilt; using lighter aluminum angle instead of T-slot, the robot shed just over 15 lbs. with no reduction in functionality. The removal of the front drive wheels, motors, gearboxes, suspension, and mountings, for reasons listed above, further reduced the weight of the robot by 30 lbs. Finally, the decision was made to remove the top server case from the robot, which contained the battery charger. Doing so necessitated a new, portable case for transporting the charger, but lightened the robot by 35 lbs. The cumulative result of these reductions is the dramatic increase in robot's acceleration, max speed, and turn speed, with no negative impacts to electrical or sensor on-field functionality

IV. Electrical Systems Report

The electrical team made significant adjustments to last year's design to improve the overall performance, reliability, and integration of the electrical systems. The team decided that to meet the objectives, Moxom'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, sturdy, and capable of handling off-road terrain. The components must be reliable, user-friendly, and rarely experience serious or minor malfunctions.

4.1. Improved requirements

To increase the safety of the robot, the team decided that the robot must contain circuit protectors to prevent unexpected damage to the components. A power supply should deliver power to the motors and electrical components of the robot separately. 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 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.

4.2. 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. One battery powers the on-board computer electronics and sensors while the other battery powers the motor.

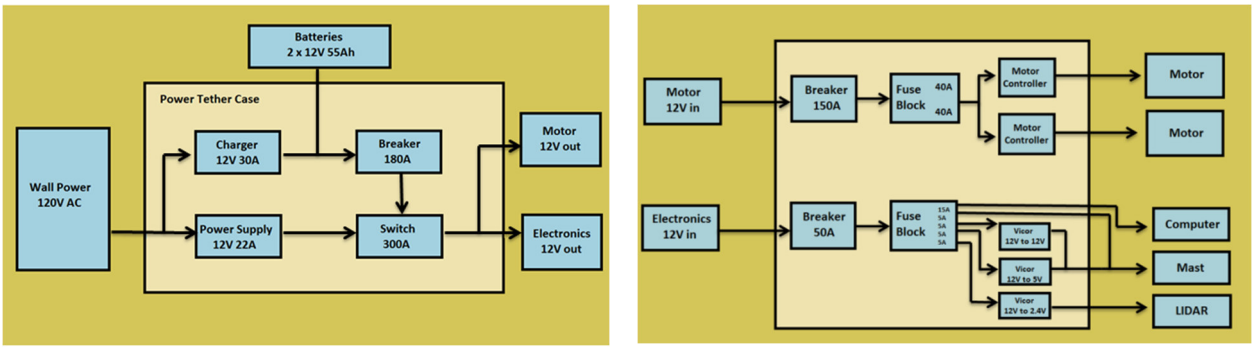

Figure 4.1: The flowchart on the left shows how the power was distributed to the motor and electronics in the original design entry. The figure on the right shows how several electrical components received power.

Last year's design consisted of three rack mount server cases. In one of the server cases, the team used an Intel i7 860S for the processor that delivers a power budget of 65W TDP. The operating system and control software were installed on a 60 GB SSD which helped reduce power consumption and eliminated the effect of moving parts. Mounted in another of the server cases included a 12V 30A battery charger, an AC-DC power supply, a main battery breaker, and a main cutoff switch as shown in Figure 4.1. The last case consisted of four Texas Instruments Black Jaguar motor controllers which provided control to the motors. The motors were controlled via a CAN bus over an integrated CAN to RS-232 bridge. Optical quadrature encoders attached to the wheels helped provide a read of the rotational speed and direction that the wheels are turning and allowed the robot to control and adjust the speed of the motors. If a problem

or malfunction was observed, the communication to the motors was interrupted via a handheld 100ms relay-controlled kill switch that transmitted packets to an Arduino microcontroller mounted on the robot.

The robot also contained an array of several sensors and a computer monitor 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.

4.3. Redesigning electrical components to meet design constraints

The team performed several tests run on the original design to observe the performance of the robot design. Critical areas of concern such as turning ability, turning speed, power consumption, average travel speed, responsiveness to the kill switch, overall performance of electrical components, and structural stability served as the basis for our tests. After careful examination, the team concluded that the overall weight, power consumption, significant motor controller unreliability, random kill switch malfunctions, inability to accurately detect its surroundings, and its relative sense of direction had to be redesigned to achieve optimal levels of performance to meet the team's final objectives.

Electrical Component Weight Restructure

The team reached a decision to remove 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 hard case. The team redesigned and reconstructed the external battery charger case in a manner that would facilitate access and ease of operation. This reduced the robot's original weight by about 40 lbs.

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 Power Distribution Restructure

The team still uses two-12 volt 60 Ah lead acid batteries connected in parallel to provide power to the motor, computer, and electronics. The team, however, increased the availability to 12-volt regulated power since all of the sensors and peripherals were changed to utilize 12 volts.

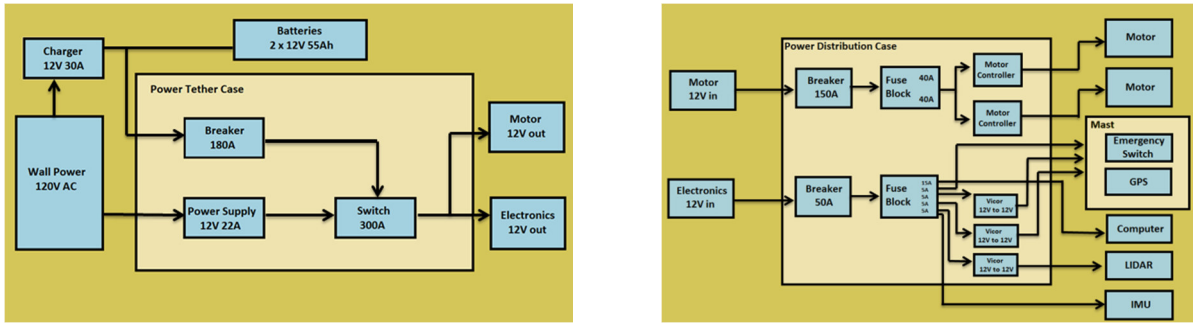

Figure 4.3: The figure on the left shows the improved power case with the charger removed from power tether case. The figure on the right shows the new power distribution to various sensors mounted on the robot.

Motor Controller Restructure

The team also reached an agreement to replace the original motor controllers. After the replacement of the two front wheels with a single caster wheel, the electrical team removed the TI motor controllers and replaced them with the RoboteQ Motor Controller MDC 2250 that only controls two motors and proved to be more reliable than the previous design.

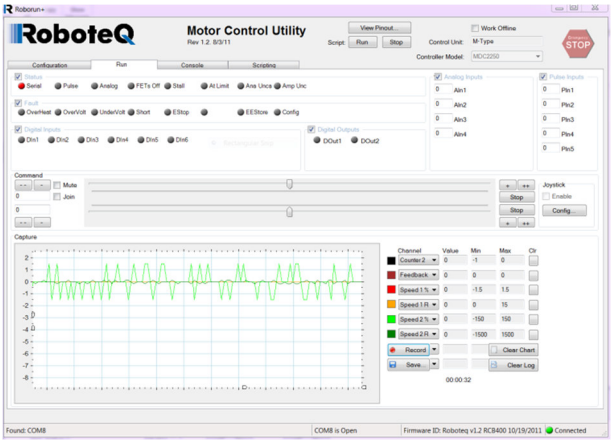

Figure 4.4: Motor controller tests showing wheels changing direction due to the direction of current flow.

The objective of this change was to add reliability, accuracy, and performance to our motor controllers. The team managed to achieve improved communication between the motor controller and the computer via RS232 to USB communication by using well-documented motor controllers.

Emergency Shut-Off Switch Restructure

After numerous tests, the team observed that the Arduino kill switch lost reliability due to signal interruption or unaccounted power fluctuations. As a result, the team replaced the entire emergency shut-off switch by replacing the Arduino boards 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 rather than setting the motor speed to zero. The handheld emergency switch device was also redesigned to make a more robust and reliable device than the previous design.

The team managed to augment the safety measures and the emergency capabilities of the robot in case of an unexpected problem. After continued testing of the new safety features, the robot constantly responded well and fast enough.

Improving the Vision and Object Detection Capabilities

The electrical team also agreed 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 power. The objective of this redesign feature was to decrease power consumption while still allowing the robot to accurately detect its surroundings.

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 from 180^{o} to 270^{o} with a 30 meter range. We decided that more scan angle rather than range was crucial since it would allow the robot to detect obstacles that are 90^{o} left or right and on the blind spots of the robot.

Added Sensors to 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, in degrees, 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 10-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, integrity, and availability by measuring small variations due to the geometry of earth, the ionospheric conditions, and electromagnetic disturbances.

After individual testing for the IMU and the GPS, the team managed to read accurate data from the IMU for extended periods of time. However, continued testing showed that the GPS first needs time to acquire sufficient data and it sometimes loses connection after it is able to acquire data.

4.4. Component performance evaluations following redesign to set software design goals

We concluded that after various tests of the components, the entire system should behave as expected with minor problems. After observing the specs of the components, we noted that the GPS is our slowest sensor. The team took this observation into consideration when integrating all the sensors and electrical components in the software. Following installation and configuration, the team then proceeded to build an intelligent, efficient, and reliable software program which would allow the robot to read data from the environment, process the data, and then perform the necessary actions and decisions.

V. Software 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 proved more advantageous since 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 instead of sequentially.

As a team, we reached a consensus that the overall software design must contain a top-level module that instantiates shared-variables between Virtual Instrument called VI's, initializes all the sensors at once, reads and manipulates the data from the sensors, performs the necessary operator loops, and correctly closes all sensors when needed. This increased the system integration between all the electrical sensors and peripherals.

5.1. Top-level module to improve system integration

Using a top level module to encapsulate the subsequent sub-processes, we are 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 to improve the system's integration. This allows for different acquisition rates that are dependent on each sensor type. For instance, the GPS updates once every second, while the LIDAR data updates at 500 Hz. This enables each sensor to operate at its maximum frequency, allowing the environmental sensor database to update as periodically possible. All sensor interfaces 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.

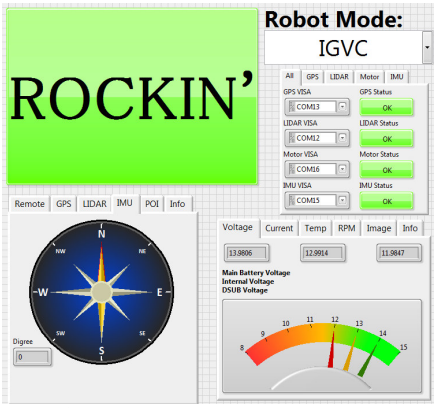

Figure 5.1: The front panel view of the top level module showing various tabs that the user can select to switch between sensor outputs and operator modes.

5.2. Describing the "sense" loop logic

The LIDAR continuously broadcasts polar coordinates that contain the angle 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 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 it the current sensor data along with the current set mode that defines the robot's behavior, which include a "tele-operated" mode for remote operation and drive control with external Wii mote or any other handheld remote and an autonomous mode for intelligent course navigation.

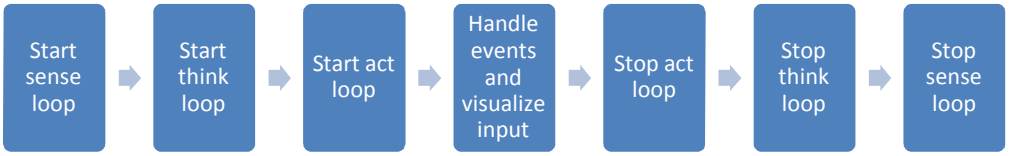

Figure 5.2: This flowchart shows the start-up and shut-down of the sense-think-act method. The order is reversed so that the motors are shut down first, followed by the computer, and the sensors. It is important that all sensors close down correctly at the same time before rerunning the code once again.

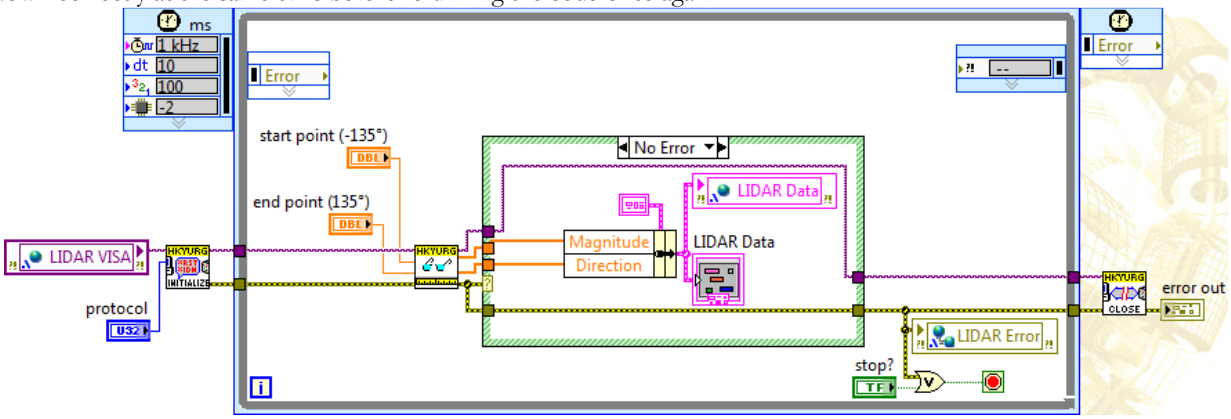

Figure 5.3: This figure shows the LIDAR acquisition loop used for mapping out and detecting the obstacles along the path.

Figure 5.4: 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.

5.3. Describing the "think" loop logic

While the LIDAR data is operating, 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 distance 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 find alternative paths that might provide a more direct path.

5.4. 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.

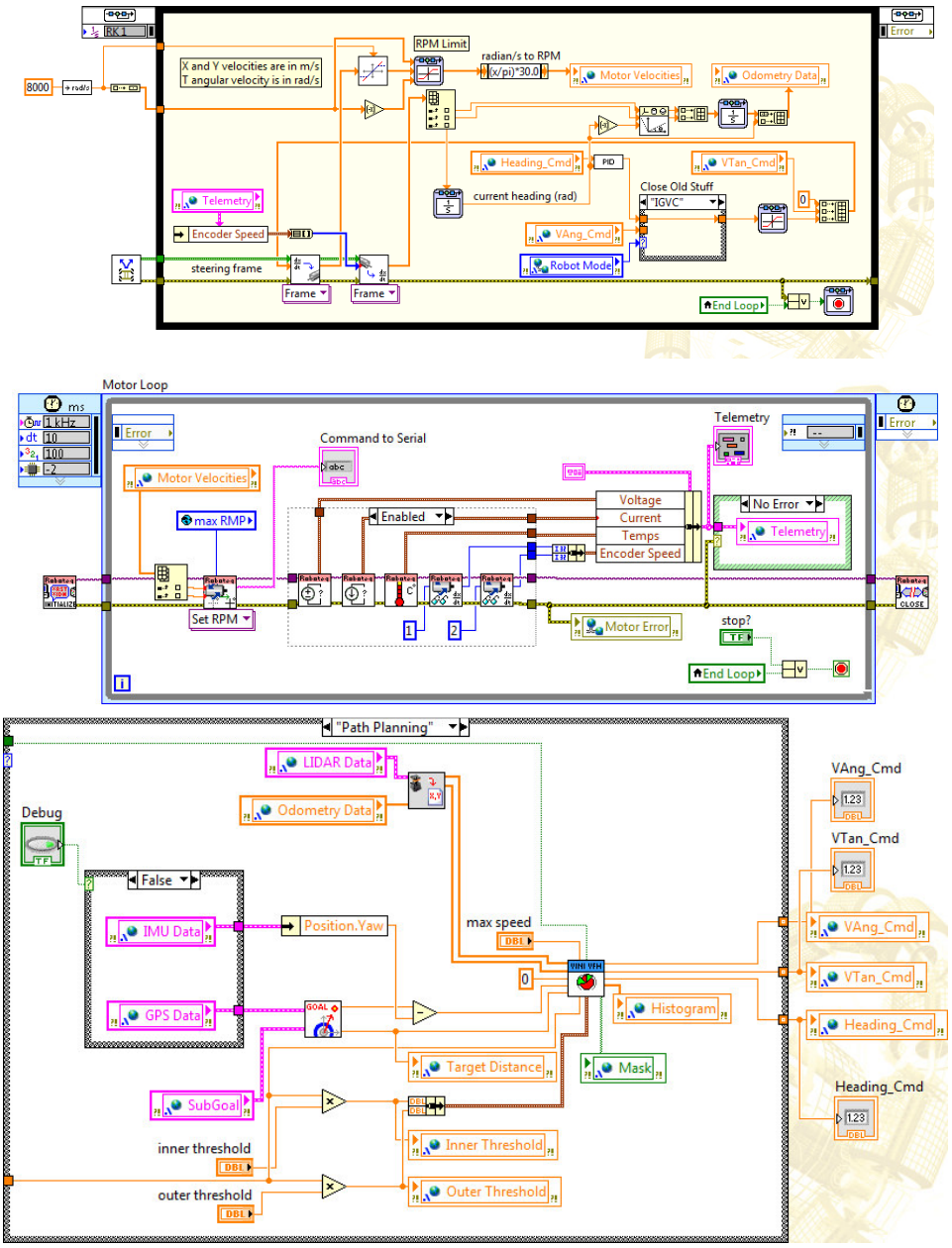

Figure 5.5: This figure shows the PID controller in charge for adjusting the turning speed as the robot turns to a given heading (top). The motor loop showing the telemetry data that user can use to monitor the state of the robot (middle). The path following VI that allows the robot to intelligently navigate the course by transposing the IMU data, the LIDAR data, and the GPS data.

5.5. Integrating the vision module with the webcam

LabVIEW's vision module has a feature that can collect images from the Logitech webcam located on the mast, allow the computer to process them, and create an algorithm that focuses on finding white lines. 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.

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.

Figure 5.6: 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.

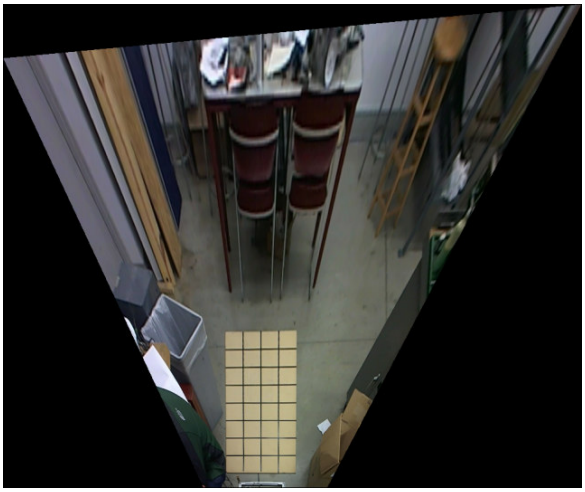

Figure 5.7: This figure shows the calibrated image from the camera. From this we can take direct measurements of objects.

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.

VI. Final Remarks and Future Perspectives

The most significant aspect of this competition is that the team managed to learn and experience the design process involved in the making of an intelligent autonomous vehicle. We learned to communicate our ideas among the team and to meet deadlines by establishing proper organizational methods such as the use of Gant charts. Furthermore, we learned that even after meeting several tasks and objectives, changes can be further made to improve the durability, intelligence, mobility, and safety of the robot.

After switching to LabVIEW, we noticed that the team accomplished more in the past 10 weeks compared to the collective work of the team from the past 3 years. It is certain that with continued calibration and fine tuning of the obstacle avoidance, path following, line detection, and waypoint algorithms, we can create a more efficient, intelligent, and complete robotics platform for future competitions.

Appendix I: Cost Estimate

Component	List	Cost to Team
Hardware		
Cases	$255	$85
Drivetrain	$400	$400
Acrylic Panels	$119	$119
Lubrication	$50	$50
80 / 20	$400	$200
Frame Hardware	$200	$200
Electronics		
RoboteQ Motor Controllers	$385	$385
Optical Encoders	$228	$228
Wire and Connectors	$380	$330
Breakers, fuses	$170	$170
Batteries	$330	$330
Tools	$56	$56
Battery Charger	$200	$200
Power Supply	$80	$80
MicroStrain 3 DM -G IMU	$1,300	$0
Hokuyo LIDAR	$7,000	$7,000
Logitech Webcam	$80	$0
Miscellaneous	$100	$100
Computer		
CPU	$360	$360
Motherboard	$150	$150
Video Card	$50	$50
RAM	$120	$120
60 GB SSD	$85	$85
TOTAL	**$14,247**	**$10,698**