Moxom's Master

Nick Wahl, Trenton Tabor, Michael Pauly, Thomas Most,
James Breen, Kyle Green, Nathan Weir, Andy Milluzzi, Ander Solorzano

{wahlnj,taborts,mosttw,breenjw,greenkt,weirnc,milluzaj,solorzaa}@rose-hulman.edu

Rose-Hulman Robotics Team, CM 5000
Rose-Hulman Institute of Technology
5500 Wabash Avenue
Terre Haute, IN 47803-3999
November 27, 2011

This report and the described vehicle were designed and constructed by the Rose-Hulman Robotics Team and the work done during the 2010-2011 school year is of equivalent caliber to that which would be given credit in a Senior Design course.

David M. Mutchler

1. Introduction

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

2. Design Process

2.1 Team Organization

The president oversees the functions of the team and is responsible for setting goals and deadlines for the project, seeking sponsors, and other administrative tasks. Three sub-teams focus on the hardware, electrical, and software components of the project and hold meetings each week. The team holds a weekly administrative meeting to discuss the progress each sub-team has made as well as any administrative business. This team structure is shown below in Figure 2.1.

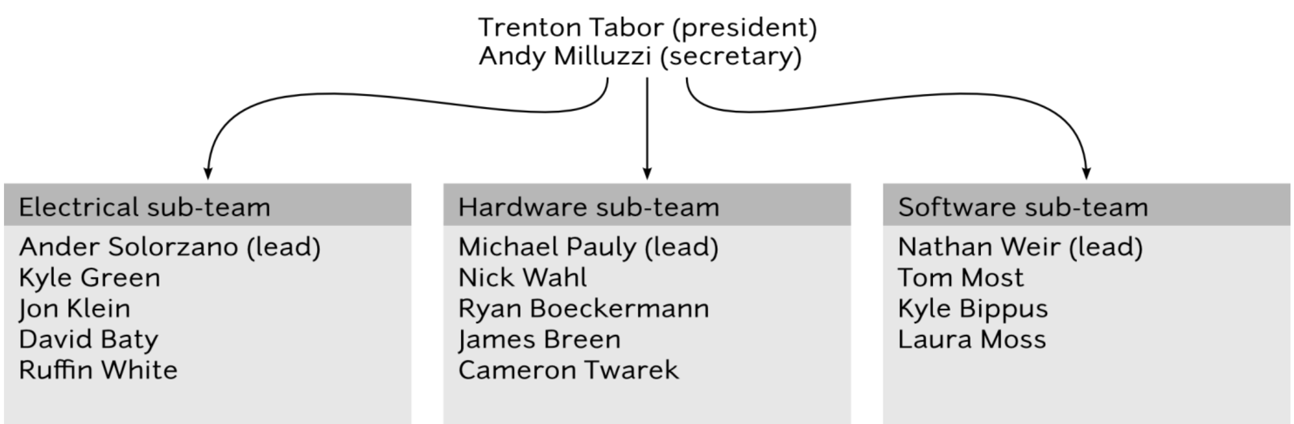

Figure 1: Team Organization

2.2 Use of Free/Open Source components

Based on our previous experience using proprietary protocols and software we made the decision to use Free and open source software and hardware wherever possible. The software we write for the robot is Free software, released under the GNU GPLv3. The electrical components we construct are also made available in the schematic form under the GPL.

2.3 Modular and expandable design

Our entry this year is intended largely as a platform for further growth, particularly with regard to hardware and electronics. Thus our design is composed of components designed to be reusable in future years.

2.4 Off-the-shelf components

We seek to balance the need for education, timely completion of the project, and budgetary constraints. Thus, we seek off-the-shelf components whenever they are appropriate. Some examples include our motors and gearboxes, which are more reliable than the custom chain drive we experimented with previously, and our motor controllers, which have proven superior to the custom units that prevented us from competing in the past. In our software we try particularly hard to leverage existing solutions, as described below.

3. Hardware

The primary goal of the mechanical sub-team is to build and maintain a durable, stable, modular platform for the electrical and software teams. A four-wheel skid-steer system, the chassis also includes a rear mast for mounting sensors and housing external interfaces. Moxom's Master incorporates several successful elements from previous robots, as well as improvements on previous features that proved ineffective. For example, the optical encoders are mounted in a protective housing and connected to the motors via a chain and sprocket system, a design taken from the previous robot. For improvements, the team has implemented the product development process as outlined by Ulrich and Eppinger in *Product Design and Development* to ensure a consistent, structured approach to our design. Conceptual designs are generated through a brainstorming process and evaluated based on criteria of safety, cost, ease of construction, and robustness, as well as the requirements of the other sub teams.

3.1 Durability

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. The first source observed was the loosening of bolts in the drive train assembly. To reduce the potential for shear failure lock washers are now used to secure all bolts in the drive train assembly. Secondly, slippage of inner support beams in the mast and central chassis was a hazard to the electrical system as well as the structure of the robot. These inner support beams were fixed with standard 80/20 tabs which have since been replaced by the use of set screws. The third source of mechanical failure was the shearing in the wheel hubs due the substantial weight of the robot. Mechanical failure actually occurred for this reason and the team's solution is delayed below.

3.2 Stability

To improve stability and reduce the chance of shaking sensitive electronic components, the entire drivetrain is mounted on two independent suspensions. In the front and back, the motors and gearboxes are mounted to a steel plate, which is in turn is attached by two springs and pivots to the chassis. This damping, couple with electronic encoders which limit the robot's speed to five miles per hour, smooths the robot's movement.

3.3 Modularity

A key feature in the modular design of Moxom's Master is the use of T-slotted aluminum 80/20 for the frame, which allows easy installation and adjustment of parts. For instance, the LIDAR, which was not planned for in the original design, was easily mounted in a forward position ideal for sensing. Additionally, this process required no cutting into the original frame, only the use of tabs and brackets to hold the new component in place. To provide accessibility to the electronics and computer hardware, all of the plastic side panels are removable. The electric components themselves are mounted on sliding drawers; coupled with quick-disconnect electrical connectors, this system allows the complete removal of the electronics for diagnosis, repair, or upgrade Finally, the rear mast, originally manufactured as a camera mount, now houses both the GPS and a computer interface, complete with monitor and keyboard, to allow instant access to the robot code for troubleshooting or feedback.

3.4 Improvements

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

The 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.

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.

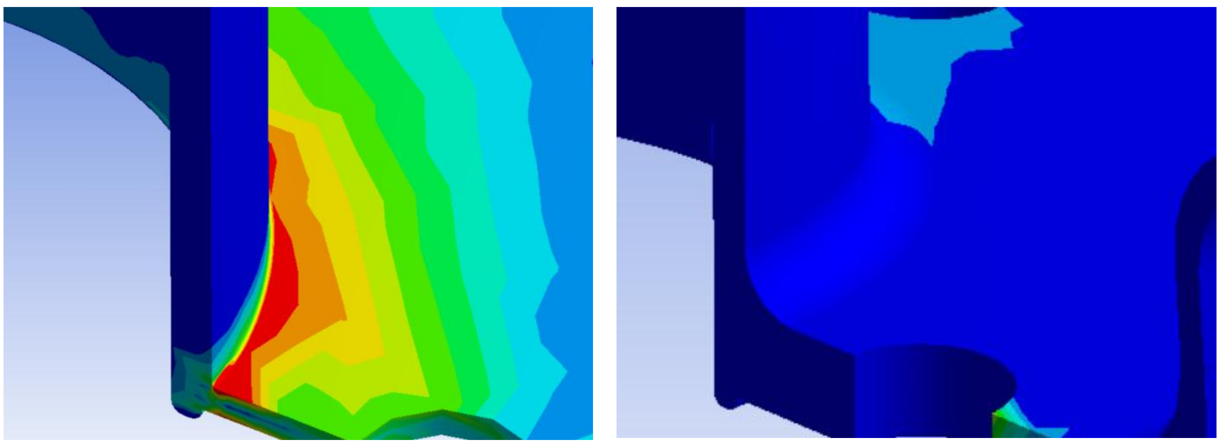

Figure 3.1: Stresses at joint for original hub (left) and redesign (right)

4. Electrical Systems

For this year's design, the team decided to review last year's design entry and make modifications to the electrical components. The team needed to come up with some solutions for improving the kill switch, monitoring the robot's state and power consumption, improving the GPS, and adding a LIDAR.

4.1 Computers

Due to a power supply failure which killed the previous computer, it was necessary to purchase a new computer this year. Given the improvements in the last four years, we were able to purchase a much more powerful system given the same power budget of 65W TDP. We selected a quad-core Intel i7 860S, the lowest-TDP quad-core desktop i7, for the processor. We were constrained by the unavailability of lower-power mobile i7 procesors for desktop motherboards. The i7 is paired with 8

GiB of DDR3 1333 RAM. The operating system and control software are installed on a 32GB SSD. Solid-state storage media was chosen for two purposes: to reduce power consumption and eliminate moving parts susceptible to vibration and shock. The computer and solid-state drive are powered using an off-the-shelf 160W DC-DC converter, and all of these components placed in a 3U rackmount server case.

4.2 Motor Controllers

Moxom's Master's motors are controlled by four Texas Instruments Black Jaguar motor controllers (MDL-BDC24), which are controlled via a CAN bus over an integrated CAN to RS-232 bridge. They are configured to perform PID speed control in concert with four Grayhill 63R, 256 PPR optical encoders.

4.3 Wireless Emergency Stop

We have improved the fail-safety of our kill switch this year, making it robust against power failure. The kill switch is comprised of two devices: a handheld remote transmitter and a complementary receiver on the robot. This year's design consists of a series of RFbee relay shields using Arduino microcontrollers for both the handheld transmitter and the board mounted on the robot. The remote transmitter periodically sends a status packet to the robot, indicating whether the robot should be enabled or disabled. The normally-open relays, through which the CAN bus runs, are closed iff the following conditions are all met: the physical kill switch is enabled, a valid status packet from the remote transmitter was received, and the value of the packet indicates that the robot should be enabled. Should the relays open the motor controllers will fail to receive packets and enter an error state after 100 ms, stopping the robot.

4.4 Inertial Measurement Unit

The IMU on Moxom's Master is a MicroStrain 3DM-G which interfaces with the computer over RS-232. This accelerometer has a three axis angular accelerometer, as well as a three axis linear accelerometer and magnetometer. This device is primarily used as a source of heading and rotational velocity information.

4.5 Camera

The camera used for line detection is a 5-megapixel Elphel 353. This camera was chosen for a number of reasons: it allows for user-selectable frame-rate and resolution, it is accessible over Ethernet, and the hardware and software are licensed under the GPLv3. This last point was especially important: since the source code for the camera's onboard FPGA was available, we have the option to perform image processing directly on the camera itself, and have designed our image processing algorithm to be portable, at least in part, to the FPGA.

4.6 LIDAR

To provide for more accurate obstacle detection than our vision system can provide, the team decided to add a SICK LMS 291 LIDAR sensor this year. With an angular view of 180 degrees and angular resolution of 1 to 0.25 degrees, the LIDAR can detect objects as far away as 80 meters. We interface with the LIDAR using RS-232 serial. Powering it requires special consideration, as it requires 24V at 20W. We use a boost converter to generate this from at 12V source.

4.7 GPS

The GPS unit used on our robot is a Garmin GPS 18LVC OEM, which has an RS-232 interface. Performance wise, this unit has a 12 parallel channel GPS receiver with a sensitivity of -165 dBW minimum, providing a position accuracy of 3 meters under 0.1 knots RMS steady state. The accuracy of this sensor represents a key future area of improvement for us, as at this point we have been unable to procure a functional unit of greater accuracy.

4.8 Sensor Network

A sensor network has been added to the robot for this year's competition to allow for the monitoring of voltages, currents, and temperatures around the robot. The most important task of this network is to monitor the battery voltage and notify the operator when it must be charged.

Small circuit boards are designed to be configured as voltage, current, or temperature nodes. Voltage nodes can measure between 0 and 30 V, using the built in ADC with multiple voltage dividers. Current nodes can measure up to 100 A, using FHS 40-P/SP600 non-contact current sensors. Temperature nodes use LM75A sensors and are accurate up to 0.125 degrees Celsius.

A monitoring board is used to facilitate communication between the sensor nodes and the computer, as well as report exceptional conditions using an on-board LCD display. The monitoring board communicates with the sensors over a daisy chained RS-485 network, serving as bus master, and relays sensor readings to the computer via USB.

4.9 Power Distribution

Two 12V 60AH Power-Sonic sealed lead acid batteries are wired in parallel to power the entire system. This parallel setup increases overall runtime and balances the use of the cells to maximize battery life.

The power system is divided into two 2U rack mount cases to ease maintenance as well as accommodate several large components. One case houses the power tethering circuitry, including the AC-DC power supply, battery charger, main battery breaker, and main-cutoff switch. Figure 4.1 shows the connections internal to the case, as well as the external hookups. The tether case takes in AC wall power and the 12V DC parallel battery connection and outputs separated 12V lines for the motors and electronics. The tie point for connecting earth, motor, and electronics ground is also in this case.

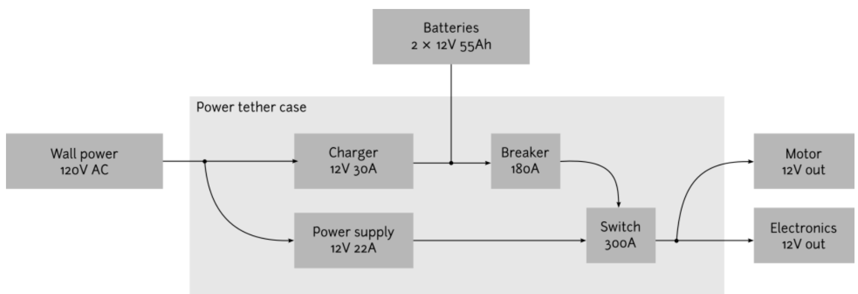

Figure 4.1: Power Tether Case Diagram

Inside the power tether case, AC power is routed to a 12V, 30A battery charger and a 13.5V, 22A power supply. The battery charger is connected directly to the battery inside the case and automatically charges as soon as AC power is applied. This hard-wired automatic charger will ensure that batteries are not over- or under-charged due to improper settings, a problem that has come up often with cheap external chargers. The 13.5V supply is used to power the on-board electronics while the batteries are charging and is enabled by switching the master switch to the secondary position. This ease of swap-over simplifies software testing and is designed to enable automated switchover with the addition of a single circuit board for next year's design.

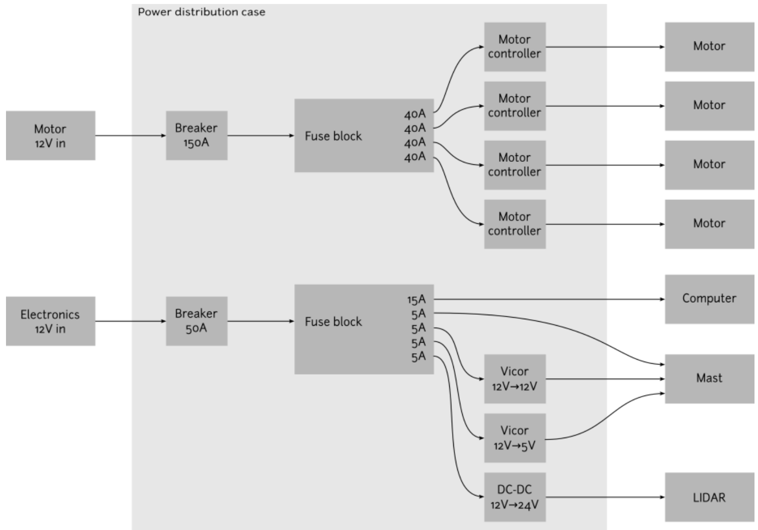

Figure 4.2: Power Distribution Case Diagram

The power distribution case takes in separate raw 12V DC connections for the motors and electronics in order to minimize noise. The motor power line is passed through a 150A manually-resettable breaker and into a quad fuse block with a 40A auto-resettable fuse for each motor controller. The motor controllers individually regulate the power passed to each motor and their outputs are passed directly out of the case to the motors themselves. The electronics input is passed through a 50A manually-resettable breaker into a fuse block that then supplies two Vicor DC-DC modules (12V to 12V and 12V to 5V), the LIDAR boost converter (12V to 24V), the computer, and the mast with raw battery power. The Vicor modules' regulated outputs are also passed to electronics mounted on the mast. Figure 4.2 outlines the robot's power distribution system.

5. Software Systems

A major change was made this year to migrate our software platform to the Robot Operating System (ROS). ROS is an open source suite of robotics tools that are intended to be used as the primary code base for a robotic software project. ROS provides bult in tools for image processing, robotic navigation, sensor manipulation, and various others robotics topics. In the spirit of ROS and the open nature of our project, we continue to use free and open source software throughout and release our own work under the GNU General Public License, version 3.

5.1 Operating System

Moxom's Master is running the desktop version of Ubuntu 10.10. Ubuntu is ideal for anyone using ROS because there are precompiled packages available.

5.3 Architecture

Adopting ROS has meant adopting its architecture, which implements several communication methods, including a publish/subscribe "topic" system and RPC function calls ("services"). The vast majority of communication occurs via topics, which are published by ROS "nodes." The node system encourages software reuse because nodes are implemented as independent processes, which eases library integration issues, particularly as compared with our previous multithreaded approach.

5.2 Programming Languages

The transition to ROS is natural to us in part because it is linguistically compatible with our previous software. ROS itself is written in C++, with bindings to Python, while Moxom's old software was written in C and Python. We do all of our development in Python for the same reasons as ever: Python facilitates rapid development and experimentation while being familiar to many Rose students and easy for others to pick up.

Because of the shared language we have been able to reuse software components from previous years, simply by adapting it to present the appropriate ROS node interface. This has been done with drivers for the Black Jaguar motor controllers, our MicroStrain IMU, and our image processing code.

5.4 Vision Processing

One of our key innovations is our line detection algorithm, which uses color and a derived metric for "texture" to classify pixels in the image. The texture and color of a presumed clear area in front of the robot are used to generate a histogram which is then used to classify the pixels in the remainder of the image. Classified pixels are processed into a set of sparse points representing the 2D locations of the nearest obstacles, mapped to the ground plane via a projective homography. This algorithm is both fast to execute and simple to implement.

Figure 5.1: Grass classification based on a “safe zone”.

This algorithm is based on the work of Ulrich and Nourbakhsh (1), who explored a purely color-based model. Here there work is extended by including texture as a metric. This is made possible by the high resolution of the Elphel 353 camera. Note that most of the following calculations are done on downsized images (the scaling is elided for clarity), but that the Sobel filter is applied to the full five megapixel image.

1. Generate an HSV histogram of a “safe region” at the bottom of the image. This safe region is assumed to be grass.
2. Compare each pixel to this histogram, marking everything that does not fit the definition of grass marked as an obstacle (see 5.1, above).
3. Apply a Sobel edge-detection filter to the value channel of the HSV image. Blur this, producing a grayscale indicator of the textural “roughness” of a pixel.
4. Mark each obstacle pixel where the roughness indicator exceeds a threshold value as a line. This is the line mask.
5. Apply a Hough transform to the line mask.
6. Projectively transform the resulting lines such that the ground plane is parallel to the view plane, also converting to physical units. The resulting lines are the line outputs.

The key observation behind the method for distinguishing obstacles and lines is that the lines, painted on grass, are texturally “rough”. In comparison, the plastic barrels and trash cans used as obstacles are smooth. Both of these classes differ significantly in color from the grass, so the three classes can be separated reasonably well. Note that it is more important to limit false positives on line pixels than it is to fully identify all lines, as small segments of false line are likely to cause problems when joining dashed lines later.

The reliance on a clear sample area of grass is both a plus and a minus here. Most importantly, it means that the algorithm is robust to changes in lighting. Testing has revealed that as the robot rotates the color and saturation of the captured images changes significantly, particularly in the early morning and evening. These lighting changes make exemplar-based methods that rely on color difficult to train, as large numbers of example images must be captured and classified to train the system. However, as the sample area is essentially an exemplar itself, if it is not in fact completely grass the entire output of the processing step is rendered invalid. Thus, a large buffer of space must be maintained in front of the robot to avoid this.

5.6 Autonomous Navigation: ROS Navigation Stack

Moxom's Master takes advantage of the ROS navigation tools, known as the navigation stack, to handle sensor-body coordinate transforms, perform SLAM mapping, and make sensory input based navigation decisions. The ROS navigation stack serves a hub that connects our sensory inputs (LIDAR, camera, odometry) to the global and local path planners. The navigation stack also maintains the local and global costmaps used by the planners.

The navigation stack makes use of both a local and a global planner, each with a costmap representing the time costs of traversing to various locations on each map. The local planner determines the local path trajectory to follow, with an emphasis on obstacle avoidance. The global planner maintains a higher-level view of the course, and focuses on generating turn trajectories. This combination of both local and global planners allows Moxom's Master to treat obstacle avoidance separately from general pathing, and achieve both simultaneously.

6. Integration and Cost Summary

6.1 Electronics Assembly

Assembly of the robot's electronics was a significant undertaking this year, as there were major changes to the robot's chassis and power systems. Electrical assembly tasks were focused on mounting and wiring the new power supplies and switching systems, as well as proper electrical conduction of the several sliding chassis to the frame for proper grounding. This required interaction with the mechanical team to properly accomplish, as minor modifications were required to ensure proper contact.

Another major integration hurdle was the addition of the large mast. The mast contains support electronics for the camera, LCD, keyboard, and killswitch, as well as mounting locations for them. Significant inter-team communication was necessary to correctly place all of the components on this mast correctly, coöperating on issues like camera angle and keyboard size and height.

6.2 Vehicle Purchases Summary

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		
Motor Controllers	$436	$436
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
LIDAR	$6,000	$0
Garmin GPS 18	$65	$0
Elphel 373 Camera	$978	$0
Miscellaneous	$100	$100
Computer		
CPU	$360	$360
Motherboard	$150	$150
Video Card	$50	$50
RAM	$120	$120
32 GB SSD	$85	$85
TOTAL	**$12,512**	**$3,749**

7. Conclusion

Moxom's Master represents not only the culmination of work from 2010-2011 school year, but also the knowledge gained through participation in the IGVC the previous three years. Though unsuccessful, these three previous entries provided a vital foundation of experience upon which to build this year's robot. This paper details the design process which the team undertook in the development of all aspects of the robot: mechanical, electronic, and software. Through participation, the IGVC offers great educational benefit, including a great deal of applied engineering experience that cannot be gained in even the best of classrooms. For this reason, the Rose-Hulman Robotics Team awaits eagerly this year's competition, and competitions in years to come.

8. Acknowledgements

We would like to acknowledge support from our sponsors, including our title sponsor Rockwell Collins, and our other sponsors Advanced Circuits, Garmin, Elphel, Precise Path, and Jason Zielke. We also received help from our club advisors Dr. Mutchler and Dr. Fisher, and from Rose-Hulman technicians Gary Burgess and Mike Fulk.

References

Efunda.com. *Stress Concentration Factor Calculator*. http://www.efunda.com/formulae/solid_mechanics/stress_concentration/calc_swss_m.cfm, 17 February, 2011.

Juvinall, Robert c. and Kurt M. Marshek. *Fundamentals of Machine Component Design*. Hoboken, NJ: John Wiley & Sons, Inc., 2000

Ulrich, Karl T. and Steven d. Eppinger. Product Design and Development Fourth Edition. New York: McGraw-Hill. 2008.

Ulrich, I. and Nourbakhsh, I. *Appearance-based obstacle detection with monocular color vision.* Cambridge, MA : MIT Press, 2000, Proceedings of the National Conference on Artificial Intelligence, pp. 866-871.