Bearcat Shredder

University of Cincinnati

Team Members: Joshua Casey and Mark McCrate
Advisor: Dr. Ernest Hall

Table of Contents

Table of Contents-- 2

Overview ---3

Design Specifications --- 5

Mechanical Design --- 7

Electrical Design -- 9

Sensor Selection --- 12

Software Design -- 13

Overview

Project History

The UC Robotics Team has a RL850 and a RL500 robotic lawnmower built by the Friendly Robotics Company. Both vehicles are used for student projects. The RL850 has since been used for three major projects.

The goal of the first of these projects was the creation of the Snowbot. The goal of the Snowbot was to create an automated snow removal system. As part of this project, a snow blower was mounted to the front of the RL850 chassis and a salt spreader was mounted on the rear. In order to control the vehicle, a parallel interface was created with the RL850's built-in controller that has been used with all subsequent projects. While the project was successful in putting the mower under outside control, the vehicle lacked sensing capabilities and was never made fully autonomous.

The first attempt to enter the vehicle in the ION Robotic Lawnmower Competition was made during 2008-2009 academic year. Matt Hagedon, Luke Grubbs, James Morris, and John Lammers added to the RL850 as part of their senior project. As electrical engineers, the focus of their project was in the designing and construction of a new computerized control system that made use of the interface created for the Snowbot project. Additionally, sensing capabilities were added to the vehicle in the form of vision, sonar, and GPS. The team entered the vehicle into the basic completion of the 2009 ION contest and received first place in that category. However, the team met with numerous hardware setbacks throughout the project that resulted in the vehicle being unable to cover fifty percent of the field during the completion run.

This year, the vehicle was again redesigned with the goal of entering the ION Robotic Lawnmower Competition. Major modifications were made to the vehicle based on insights of the current team members along with reflections made by the 2009 team.

Team Organization

This year's team is made up of Joshua Casey and Mark McCrate. Joshua is an undergraduate senior in UC's computer science program and is working on the Bearcat Shredder for his senior design project. Mark is a master's degree student of mechanical engineering and came into the project with a wealth of experience working with the RL850 having been part of the Snowbot project as well as lending some assistance to the 2009 team.

Both Joshua and Mark are part of UC's Robotics Team and the team advisor, Dr. Ernest Hall, is also the advisor for the 2010 ION team. Other members of UC Robotics Team and the noncompetition-oriented Robotics Club have also lent support to the project through advice, encouragement, part donation, workspace, and funding.

For the division of project tasks, Mark has taken the lead on the mechanical portions of the project. Likewise, Joshua is leading the efforts on the software aspects. For the electrical and administrative needs, both Joshua and Mark are equally contributing.

Mower Design Strategy

From the start, the intention for this year's design was to make major changes to the design of the 2009 team. The two main reasons behind this intent for change were reflections made in the final design project report of the 2009 team and the performance of the 2009 vehicle. Additionally, it is the goal of the 2010 team to enter the advanced portion of the ION Robotic Lawnmower Competition instead of the basic portion that the 2009 team participated in.

While the 2009 team took first prize in the basic course of the 2009 ION Robotic Lawnmower Competition, they were the only entrant in that category and their vehicle mowed less than half of the grass during their scoring run. The primary reason for this is that the project was plagued by hardware-related setbacks throughout the year. The custom control system built by the 2009 team was difficult to access for modifications and was vulnerable to shorting. As a result, multiple electrical components of the vehicle were destroyed over the course of the project and much repair work was needed. In their final project report, the 2009 team stated that the majority of their problems could have been avoided if a compact laptop were used to run control software instead of building their own custom computer.

Taking in the lessons learned by the 2009 team, the design for this year's vehicle uses a Dell Latitude D800 to run the control software for the mower. For communication with RL850 chassis, the same parallel interface with the built-in controller for the RL850 used by the Snowbot and 2009 vehicle is again used. However, a new electronic circuit was made in order to implement E-Stopping that serves as an intermediary between the laptop and mower. This circuit connects to the laptop using USB and outputs to the mower via the parallel interface. Together, these components serve as the physical control system for the mower.

In order to tailor the vehicle for usage in the advanced completion, the software control program for the mower was redone as well. The 2009 team used the A* searching algorithm for path planning through the course they divided into mower-sized cells similar to the approximate cell decomposition method. In an attempt to optimize for the advanced completion, the 2010 control program now uses a boustrophedon coverage pattern and exact cell decomposition. The goal of these changes was to enable the vehicle to mow closer to obstacles while producing a cleaner and more complete overall cut.

While a different set of individual sensors is being used this year, they serve much the same purpose. Vision and laser scanning are being used for obstacle detection. GPS and compass are being used for positioning data. New this year, however, is that the vehicle is making use of encoders located on the wheel motors of the RL850 as an additional form of position information

Design Specifications

Dimensions

A factory Friendly Robotics RL850 is 0.9m long, 0.66m wide, 0.32m tall, and has a mass of 39kg. After modifications to support control system and sensor mounting, the Bearcat Shredder is 1m long, 0.66m wide, 2m tall, and has a mass of 57kg.

Mower Speed

The RL850 chassis has two speed options: high speed is 0.5m/sec and low speed is 0.25m/s or 1.8km/hr and 0.9km/hr. The Bearcat Shredder currently only mows using the slow speed, but future modifications may be made in order to support the high speed.

Cutting Width

The RL850 chassis has a cutting width of 53cm.

Energy Usage

The Bearcat Shredder runs entirely off of electrical power. The RL850 is powered by 24-Volt DC battery while the laptop is independently powered by its own 11-Volt battery. The battery operating the RL850 can operate under mowing load for three hours after a full charge. Operating at the slow speed, the chassis is capable of mowing 1431 square meters per charge. However, the aging battery used by the laptop is only able to keep the control system powered for 1 hour between charges limiting the design to 477 square meters per charge. Modifications to the Bearcat Shredder to enable the RL850 chassis to sustain the charge on the laptop battery, but tests have not yet been completed to determine the resulting cutting limit.

Cost

Item	Value
Friendly Robotics DL850 RoboMower	$1500
Dell Latitude D800 Laptop	$400
PNI TCM3 Electronic Compass	$1046
SICK LMS-200 Laser Scanner	$3000 (with academic discount)
Point Grey Bumblebee2 Stereo Camera	$3,200
Garmin GPS 76	$185
Mounting Material	$75
Total	**$9406**

All parts, materials, and sensors were purchased by the UC Robotics Team except for the electronic compass which was donated by PNI to the university.

Mechanical Design

Mower Chassis

Knowing that the problem of designing a mower, even a small battery powered eco friendly green mower, has been solved, we chose to stick with a commercially available lawnmower chassis as the starting point for our design. The Friendly Robotics RL850 chassis we are using provides reliability, durability, a compact design, and zero-point turning. This greatly simplified design and enabled us to focus solely on mounting sensors and the control system. CAD was only used for planning sensor locations.

Sensor Tower

A six foot sensor tower is added to the chassis. The sensor tower is composed of a six foot aluminum tube that slip fits into a trailer hitch at the robot's rear and is held in place by gravity. This tower allows the stereo camera to be located at an optimal viewing angle and elevates the GPS antenna for better signal reception.

Device Mounting

Laptops are much heavier than our other sensors. Therefore, we mounted it low and near the vehicles center. Also because the hard drive etc is sensitive to vibrations we have a designed a shock/vibration absorbing tray on which to perch the laptop.

The wheel motor encoders are part of the stock RL850. As such, they are securely located within the water-resistant chassis of the mower.

The compass PCB is bolted inside a water resistant case and hot glued to the computer tray. Using hot glue for this low mass part reduces hole count.

The laser scanner is bolted to aluminum tracks which are zip tied to the power inverter. Zip ties provide ample strength and design modularity.

The stereo camera is mounted to a professional mini-tripod with graduated pitch, roll and yaw indicators to facilitate coordinate transformations and damp vibrations. The tripod is bolted to a short piece of square tube so it can be unbolted easily. The short tube slip fits into the top of the sensor tower.

The GPS unit is mounted beside the laptop. The antenna for the laptop is mounted at the top of the sensor tower.

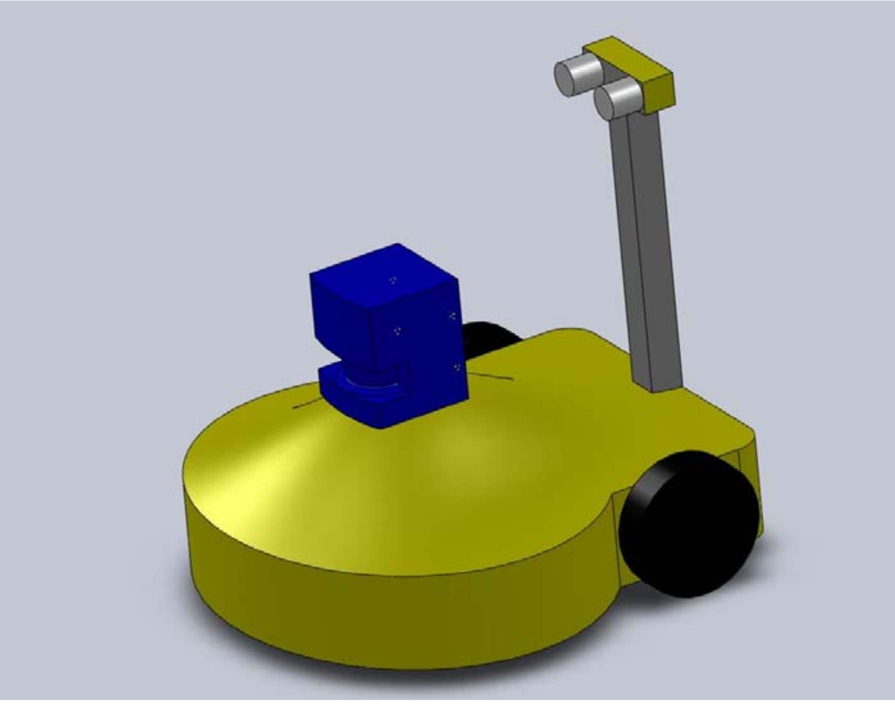


Figure 1: Mechanical Design Concept

Avoiding Actuation

Review the 2009 design reports reveals that half the teams designed/built/used actuators on some part of the robot, whether it was to move sensors or the mowing deck. We choose to honor the Keep-It-Simple rule and design all parts to operate from static positions because this reduces part count, complexity, weight and ultimately reduces points-of-failure.

Electrical Design

Simplicity

Back in 2005, the DARPA Grand Challenge winners repeatedly stated before and after the race that robotics is largely a software problem. Keeping that in mind we only designed/built one piece of custom electronics to augment the base RL850; our E-Stop/controller interface.

Controller Interface

The Snowbot controller was loosely based on IEEE 1284 parallel interface. The wires from a parallel cable were individually soldered to button leads inside the RL850's controller. By this means, a ground signal sent into the lead had the same effect as pressing that particular button on the RL850 controller. However, this design did not support E-Stopping. The second iteration included E-Stopping, but still had other drawbacks from continued use of the 1284 interface. The first is that 1284 ports produce undesired control signals during startup. This resulted in small amounts of movement by the mower during start-up and, of more concern, brief activation of the mower blades.

The current design of the control interface fixes these problems. The crux of the design is the addition of the Velleman K8055 USB based DIO/AIO board and a custom digital circuit to serve as an intermediary between the laptop and the mower. The laptop communicates with the K8055 via USB, the digital circuit filters commands based on the E-Stop state, and the digital circuit forwards those commands that pass on to the mower via the parallel interface. The K8055 board is able to handle the handshaking during computer startup without affecting its output and the digital circuit provides a means to implement the E-Stop.

E-Stop Switching Circuit

As previously mentioned, the Snowbot had no E-Stopping capability. The 2009 E-Stop worked by cutting power to the computer. This depended on the still powered mower chassis to stop on its own, took away data logging capabilities, and was harsh on the computer.

This year's E-Stop design works instead through signal blocking. Each of the output signals from the Velleman K8055 board corresponds to a particular command to the mower. A ground signal sent to the mower indicates that that particular command should be active. When the E-Stop is triggered either from the manual button or wirelessly, a high signal is combined at the gate level using Boolean logic to prevent motion or blade signals from propagating through. Additionally, the mower's own STOP command is sent through and causes the mower to stop moving and turn off the blades.

One problem encountered with this design is that holding the STOP signal caused the RL850 to continually power off and back on. This was solved by the addition of capacitor and resister. When initially receiving the STOP signal, the capacitor allows a pulse of the STOP signal to propagate through before charging and no longer propagating the signal. The resister, which is connected parallel to the capacitor, to discharge once the STOP signal has ceased. The resistance on the resister is high enough that the RL850 controller does not interpret the small amount current making it past the capacitor as being a command signal.

When initially powered, the circuit starts out in a stopped state. When the "all clear" button is pressed, a signal relay holds the circuit in the clear state until the E-Stop is activated. The momentary circuit connection made by the "all clear" button causes the relay to pull its switches to the secondary state. From there, the signal relay holds itself open until the circuit is interrupted either the manual or wireless E-Stop.

The manual E-Stop button is configured such that when the button is pulled up, a ground signal is propagates through one portion of the button and a high signal through the other. The ground signal is required by the AND gates in the gate portion of the circuit to propagate control signals. The high signal enabled the signal relay to be held open once the enabling button is momentarily pressed. When the E-Stop button is depressed, both signals are cut off. Cutting off the high signal results in the signal relay falling back into the stopped state. Since the RL850 controller interprets ground signals as the active state, the AND gates in the E-Stop switching portion of the circuit require a ground signal to propagate any such signals through. So when the ground signal coming through the E-Stop button is cut off, high signals are sent to the controller for the motion and blades. Additionally, the STOP line grounded such that the RL850's internal stopping is activated.

The wireless E-Stop is implemented using parts from a garage door opening assembly. When the wireless E-Stop button on the remote is pressed, a momentary ground signal is sent into the circuit. This causes the signal relay to drop into the stopped state.

To bring the mower out of the stopped state, the manual E-Stop button must be in the pulled out position and the enable button pressed. This will cause the signal relay to once again be held open.

Encoder Buffering

A problem was encountered when first using the wheel encoders. Sending the signal directly to the digital inputs of the Velleman K8055 board caused interference with the RL850s internal control system. Resisters were initially tried as a solution, but a middle ground could not be found in which the RL850 functioned normally and the K8055 was able to sense the encoders. The final solution was to send the encoder signals first through inverts as a buffer. The impedance on the input side of the inverts is high enough that the RL850 is not affected by usage of the inverts and full signal strength can be sent from the output side of the gates to the K8055. Since the encoders only alternate between high and low signals without regard to direction, inverting the signal has does not hamper usage of the encoders.

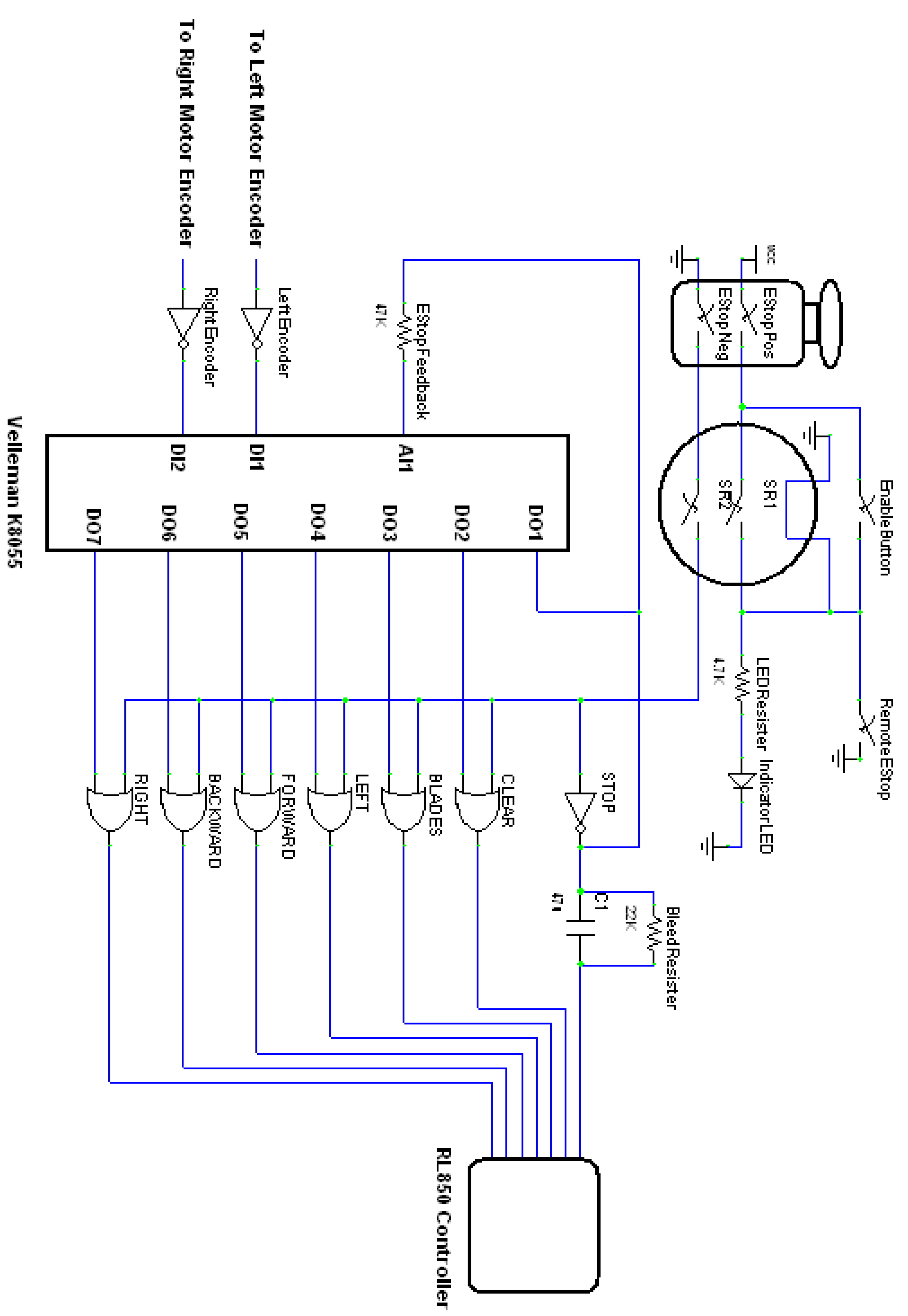


Figure 2: Circuit Diagram of Bearcat Shredder Electronics

Sensor Selection

Encoders

Encoders come standard on RL850s. We need encoders for odometry. An encoder-to-computer interface is not standard. We needed to tap into the encoders and run these splices through inverter buffers before reading them into the K8055 board therefore long leads are run and protrude from middle of the robot then plug into our Velleman/E-Stop circuitry.

Compass

A PNI TCM3 electronic compass aids turning because there is significant wheel slip on grass. It was best to mount the compass to the computer tray since it is a large, flat and well oriented to the robots axes. RS-232 is used for communication and we inventively use a standard USB port for power rather than some AC/DC converter.

Laser Scanning

Laser LIDAR is made possible by the SICK LMS-200, a staple in the robotics community for detecting objects/obstacles for years. This light barrier/noncontact bumper is mounted to the top of the inverter, where it can get a good view of the approaching terrain but is low and centrally located so as to not compromise or increase turning inertia. The laser can range objects out to 25m and can do high resolution at 8m. Mounting it at a downward facing angle will reduce these numbers. In either case, it can see obstacles at a good distance. This will be the first year the UC Robotics team attempts high speed RS-422 communication in-situ with the LMS laser scanner.

Vision

A Point Grey Bumblebee2 will be used for vision. This camera captures stereo images suitable for obstacle classification and is well documented. Unlike the laser scanner, cameras need to be mounted high to get a good survey of the surrounding area. Therefore the Bumblebee2 is mounted on a six foot pole on the robot’s rear where it can counterbalance the laser and inverter. Power comes straight from the mowers 24VDC bus and data streams through an IEEE1394 port.

GPS

GPS is utilized as well. Garmin’s trail rated GPS 76 is suitable for us, since we only need relative accuracy, and they come with a convenient cradle that mounts to the laptop. 76’s can run on 2 AA batteries and interface via RS-232. A perfect SLAM would provide just as much information but is more computationally expensive.

Software Design

Coverage Algorithm

The 2009 team a method similar to approximate cell decomposition to achieve coverage and A* searching for path planning. While this approach would theoretically work well for simple situations, it had some drawbacks that we wanted to avoid for this year's design. To achieve coverage this year, exact cell decomposition and boustrophedon coverage are used.

The first issue we wanted to avoid was the heavy dependence on having accurate GPS readings. Last year's algorithm called for the mowing area to be decomposed into essentially mower-sized squares and tracking which squares had been covered. If GPS readings drifted, coverage would be hurt as the mower lost alignment with its initial cell placement.

This issue is resolved by using a boustrophedon coverage pattern. The boustrophedon coverage pattern involves covering the length of the area, then doubling back on a parallel course. By combining wheel encoder and compass readings, a relatively precise u-turn can be calibrated. By making the u-turn and paralleling the previous course back, tight coverage can be achieved that is not dependant on GPS. This shifts dependence to the compass and wheel encoders, but the u-turn itself can be calibrated to leave more or less overlap over the previous course to leave a margin of error.

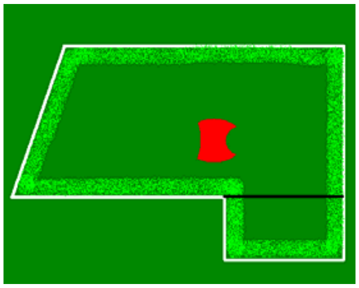
The second issue dealt with obstacles. In approximate cell decomposition, a cell is deemed either totally free or containing a hazard. Hazard squares are not entered. In the advanced competition scoring, grass near obstacles is very point dense. So when a square is only partially filled with an obstacle, coverage would not be attempted for the rest of that square and result in significant point loss.

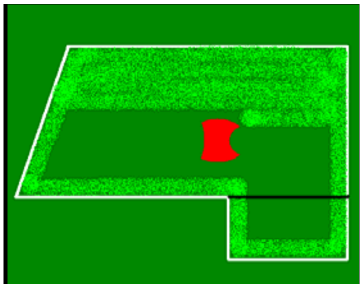
Exact cell decomposition does not have this problem. Cell boundaries are placed right against obstacles. So in covering, the mower would leave less of that point-dense grass unmowed.

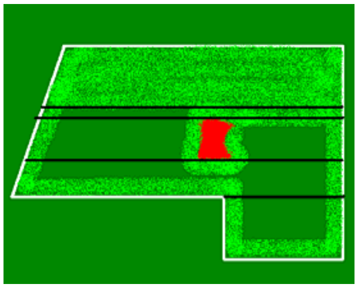
Both methods require some previous knowledge of the environment in order to define the location of cells. The same approach is used this year as last year to gather this starting information. At the start of the run, the mower will find an outer boundary line and trace that line until it returns to starting location. GPS seeds are taken along the way and used to define the boundaries of the course. This data is then used as to initially define the placement of cells and coverage can begin.

This approach, however, needs special handling for obstacles encountered later during the run. The A* path planning algorithm used last year would deal with this by finding a clear path through via the Manhattan method. This year, the mower will trace the boundaries of newly found hazards in a similar fashion to what was done with the outer boundaries and use that data to recalculate cells before resuming coverage.

To handle dynamic obstacles, the control software will attempt to calculate a velocity vector for a detected hazard. If the hazard's velocity vector indicates the obstacle is moving, the mower will wait for the obstacle to leave before resuming coverage. Waiting for the obstacle to leave is meant to avoid leaving uncut grass or having to return to the location later. If the obstacle does not leave before the mower has waited 45 seconds, the mower will attempt to move around the obstacle before resuming coverage as a contingency plan.


Define Initial Cells


Perform Boustrophedon Coverage


Redefine Cells as Necessary

Sensor Usage

The three sensors being used for positioning data are GPS, compass, and wheel encoders. To offset the error rates of any individual device, Kalman filtering is used. Compass and wheel encoder data is filtered together for calculating orientation. GPS and wheel encoder data is filtered together to achieve dead reckoning for positioning.

Hazard detection is dually handled by the stereo vision and the laser scanner. At this time, the two are mutually exclusive in that vision is only being used to detect lines and the laser scanner is solely used to detect solid obstacles. The two readings are jointly used to determine if it is safe for the mower to enter a location. No filtering is required.

Software Architecture

The control program itself is written in C++ using an object oriented approach. Each sensor has a wrapper class defined. The main routine follows the coverage algorithm by manipulating a Mower class to follow simple, high level behaviors. Wrapper classes have been created for each sensor used and for the Velleman K8055 output. The Mower class utilizes the sensor wrapper classes to extract formatted data and sends commands to the mower itself through the K8055 wrapper class.