Moogie the Robotic Lawnmower

Richard Van Hook, *Wright State University, Dayton, OH 45435*
Nicholas Baine, *Wright State University, Dayton, OH 45435*
Dr. Kuldip Rattan, *Wright State University, Dayton, OH 45435*
Dr. John Gallagher, *Wright State University, Dayton, OH 45435*

ABSTRACT

This paper presents the design and implementation of Moogie the Robotic Lawnmower. The objective is to build a lawnmower with the ability to mow grass in a specified area autonomously, while avoiding collisions with obstacles located within that area. The main task is to create a functional and efficient navigational algorithm which is able to process data from a differential GPS, IMU, and magnetometer. The output of the algorithm controls the speed and direction of the mower.

INTRODUCTION

Mowing a field of grass is considered to be tedious chore by many, so there is a demand for methods to eliminate such tasks. The customary method of dealing with unpleasant tasks is to pay someone else to perform them. There is another option, however, and that option is robotics. Robots are currently being used in industry to perform manual labor tasks, but only a few applications are available for use by a home owner. At this point, robots are being introduced for many simple tasks such as vacuuming - a task similar to lawn mowing. Like vacuuming, mowing is a simple and time-consuming task that the average person generally prefers to avoid.

The robotics industry has responded to this anticipated demand by creating a simple robotic lawnmower capable of autonomously cutting grass and navigating around obstacles. The mower available is relatively simple. It does not mow with any particular pattern; instead, it uses a copper wire laid around the perimeter of the area where it is allowed to operate. It runs in a straight line until it senses a copper wire or it runs into an obstacle. When either of these events occurs, the robotic mower changes its direction and continues its operation. Obstacle avoidance is done with only a bumper. The mower simply discovers obstacles by bumping into them. The mower has no onboard navigation other than its ability to react to collisions and identify a line it is not allowed to cross. This may work relatively well for small lawns and lawns of simple shape, but it has its shortcomings when the area to be mowed is of complex shape or substantial in size.

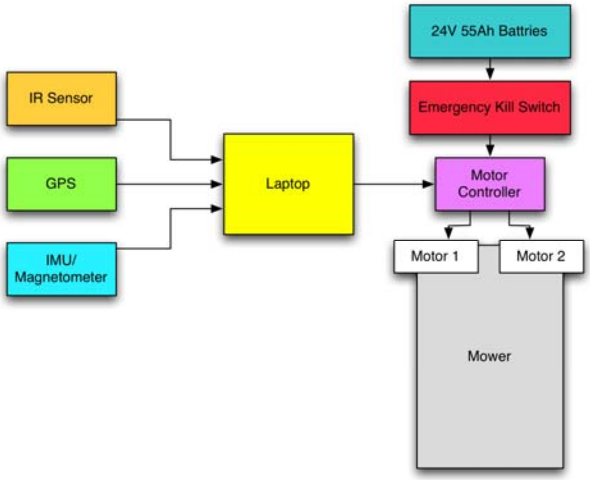


Figure 1: Block Diagram of the robotic lawnmower's systems.

A more sophisticated design for a robotic lawnmower is needed. Moogie is the answer. Moogie navigates through the use of Global Positioning System (GPS), an inertial measurement unit (IMU), and a magnetometer (see Figure 1). This equipment enables the robot to mow in specified patterns. To avoid collisions with obstacles, Moogie also includes an infrared sensor. All sensors send information to a central computer (in this case, a laptop) that runs a navigation algorithm allowing it to make decisions and direct the motors accordingly.

Although mowing grass is an important task that must be performed (which role Moogie fulfills), it also has two additional purposes. It is designed to (1) compete in the Institute of Navigation's Robotic Lawnmower Competition; and (2) to serve as a platform for navigation and sensor research.

THE TEAM

The team that developed Moogie consists of two students - Richard Van Hook and Nicholas Baine. Both are graduate students at Wright State University. Nicholas Baine is an electrical engineer and does research in the area of electronic navigation. Richard Van Hook is a computer engineer and software developer. He performs research on sensor systems. Nicholas was responsible for electrical systems and navigation algorithm development. Richard was responsible for all software. Both shared in the mechanical fabrication and construction of the robot. The pair of students is advised by Dr. Kuldip Rattan and Dr. John Gallagher.

LAWNMOWER DESIGN

The lawnmower frame is composed of 80/20 T-slotted aluminum and was chosen for ease of fabrication. Inside the robot, a manual reel mower deck occupies the base of this frame. It is important to choose the correct type of mower deck. Vibrations are of concern in this application, as well as electrostatic interference generated by the ignition system of a gas mower. In previous competitions, gas mowers have stalled and proven undesirable due to the vibrations they generate. The reel style mower deck was chosen based on its ability to cut grass at lower revolutions per minute (RPM) with less power than a rotary mower. The downside of using the reel mower is that it needs to be routinely adjusted and sharpened to maintain performance.

The reel style mower used is normally powered by a person pushing it; consequently, it would be difficult to maintain a cut as it was designed by simply attaching it to the robot. Therefore, the drive wheels that powered the blades were removed and an electric motor and belt drive system were put in its place. The motor and belt pulleys were chosen to spin at 500RPM, which is ideal for this style of mower. Plating was used to separate the mower deck area from the rest of the robot to prevent grass clippings from blowing into the upper compartment of the robot.

In addition to the mower deck on the bottom, there are two separate 24V/16A drive motors to provide mobility to the robot. In the back, two 12V batteries, connected in series, supply 24V to the electric motor powering the mower and the motor controller, which controls both drive motors. The power supplied to the mower deck and the drive motors can be terminated instantaneously by the use of safety kill switches. The robot has two safety switches. One is a large red mushroom switch that is located on the top of the robot near the rear; the other is a remote keychain transmitter similar to those used on remote keyless entry systems for automobiles. The kill switches are integrated into the system in such a way that, if a wire came loose or the switch broke, the system would break in the off position.

To prevent the electronic noise produced by the motors from interfering with the operation of susceptible electronic devices, a third battery is used for powering electronics on board, leaving the first two batteries supplying the 24V for powering motors only. These batteries have 55Amp Hours of capacity. With the two drive motors and the mower deck operating at full speed, the mower can operate for 1.3 hours. The battery providing power to the electronics lasts over 3 hours on a single charge to allow time for setup and programming when the mower and motors are off.

While the motors and batteries are located toward the rear of the robot, the electronics and computers are located in the middle of the robot above the mower deck.

THE FRAME

The frame material selected is T-slotted extruded aluminum, pictured in Figure 2, because it is very durable, light weight, affordable, and easy to assemble. The material was cut to a desired length and then a vertical mill was used to drill anchor holes. As pictured in Figure 2, the anchor fastener fits into the anchor hole and the bolt in the anchor was screwed into a T-slotted nut in the T-slot of the outer piece of frame and tightened to connect the frame. The material and anchors were purchased from a company called 80/20 Inc. Using the T-slotted frame and specially made bolts that attach to any surface of the frame have allowed for simple connections of the motor and wheel mounts. This eliminated the need for an expert in machining and/or welding, saving time and cost of labor.

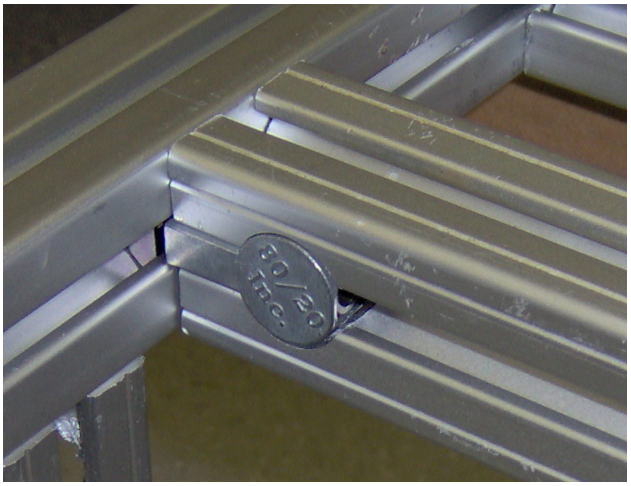


Figure 2: T-slotted extruded aluminum.

Figure 3: Scott’s Classic Reel Lawnmower.

MOTORS AND REAR WHEELS

To choose the correct motors, the rear wheels need to be taken into consideration because the radius of the rear wheels affects the torque and RPM requirements of the motor. The wheels used are 14 inches in diameter, 4 inches in width, and weigh 10.2 pounds per wheel, as shown in Figure 4. The burly lug on the tire is made for outdoor traction, and the rim is made with stamped two-piece steel. The accompanying hub, pictured in Figure 5, from NPC was selected to connect the motor to the wheel.

Figure 4: Rear Wheel.

Figure 5: Hub.

With a rear wheel selected, analysis was performed to determine the specifications for the motors. Two of the most important specifications for motor selection were the output torque and the output RPM. The output RPM is important because it insured that the mower would run at the highest speed allowed in the competition. For this particular lawnmower, the maximum weight is set at 100 kg, the maximum speed is set at 10 km/hr (required for the competition), and the wheel diameter is set at 14 in. (based on the selected wheels). The acceleration is estimated so that the lawnmower reaches its max speed of 10km/hr in 5 seconds. The required torque, based on the estimated acceleration, was found to be 87.43Lb-in. The motors run at approximately half of the maximum torque while cruising; however, for a safe and competitive design, the motor needed to have at least 87.43Lb-in of torque. In addition to torque, the output RPM required was determined to be approximately 150 RPM.

Based on these specifications, the motor selected is model # NPC-PT5306 from NPC Robotics, shown in Figure 6. In addition to having built in encoders, which are used for feedback control by a motor controller, it also meets the torque and RPM requirements.

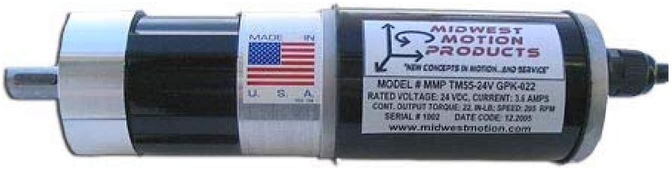


Figure 6: Midwest Motion Motor.

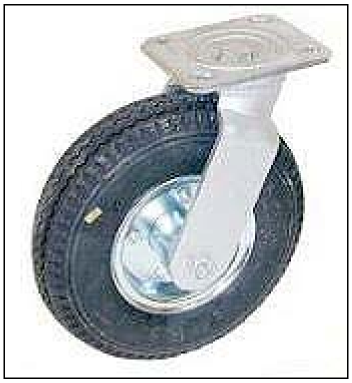


Figure 7: Pneumatic Caster Wheel.

CASTER WHEELS

The front two wheels on the lawnmower are caster wheels as illustrated in Figure 7. Caster wheels are used in the front because they are designed to swivel and move in the direction the back wheels are driving. The pneumatic swivel wheels have an 8 inch diameter and have a 2 inch width. The pneumatic wheels help soften the impact of the lawnmower on the ground. This helps decrease vibrations and movements that could be harmful to the functioning of some of the electrical components.

MOTOR CONTROLLER

The motor controller chosen for this design is a RoboteQ AX2850. This controller was chosen because it has two independent power stages required for control of two independent drive motors, and these power stages can handle the delivery of 80A each for an extended period of time. The motors consume a maximum of 16A each, providing a safety margin of 5A for the motor controller. Requiring the motor controller to run no higher than 20% of its capacity makes the heat sink temperatures easy to manage. The controller also has a built in proportional plus integral plus derivative (PID) controller and works off the input from the encoders on both motors. This feedback control is programmable and is set up using the Ziegler-Nichols rules for tuning. This makes programming the controller possible without having to model the response of the mower as a system.

The controller also has two modes of operation built into it: radio controlled (R/C) and PC serial communications (RS-232). The R/C mode is used in conjunction with a receiver and controller to allow for a human interface. This interface can be used during testing and to move the mower from location to location. The RS-232 mode is used when mowing, giving control of the motors to the computer. The computer then determines the speed required for each of the motors and sends commands via RS-232 to the motor controller. In addition, the computer makes requests for information from the controller. These include encoder counts (to detect wheel slippage), heat sink temperatures, and analog input state (to monitor the bumper). The computer also controls the state of a digital output on the controller, which turns the mower blade on and off. To switch between the two modes, a simple R/C relay switch has been utilized. When the R/C transmitter is turned on, it sends out a signal that activates the switch, changing the mode of the controller to R/C mode. When the transmitter is turned off, computer control is restored.

THE COMPUTER

The computer that interfaces with the RoboteQ controller is an HP 8710W Mobile Workstation. It contains no RS-232 ports. Since RS-232 is needed to communicate with sensors and motor controller, a USB to RS-232 converter is used to provide 8 RS-232 ports. The computer runs custom software that sends commands to the RoboteQ controller based on information gathered from several sensors via RS-232 connections. These sensors include a differential GPS, IMU, magnetometer, and infrared (IR) sensors. The first three of these sensors (the differential GPS, IMU, and magnetometer) are used to navigate the mowing field. The IR sensors are used for collision avoidance.

GLOBAL POSITIONING SYSTEM (GPS)

GPS is the main source of position information used by the robot. To mow grass in an efficient manner and stay within a set of boundaries, the position of the robot needs to be precise. The average GPS device is accurate to approximately ten meters. This error is unacceptable for use in a robotic mower used to cut grass in a straight line (Refer to Table 1 for a breakdown of the sources of this error).

GPS works by using a receiver to listen to satellites, using the time of flight of the signal in calculating the distance between the position of the antenna and the satellites. Given the distance to several satellites, a position is calculated by triangulation. Several non-ideal conditions make the determining of an accurate position a complicated process. The first error is in the timing. The timing between the satellites has to be extremely accurate. Satellites have highly accurate atomic clocks onboard, but the receiver does not have anything approaching that level of sophistication. The error in the receiver clock is minimized by a least squares algorithm that determines the best time fit given the time of flight information of more than three satellites (the minimum number required to triangulate a location).

Other errors are due to the atmosphere, satellite position errors, and multipath signal reception. To minimize atmospheric and satellite position errors, there is a satellite-based augmentation system available in the U.S. called Wide-Area Augmentation System (WAAS). It transmits updates and corrections for ionospheric errors and satellite orbit errors. This improves the known position to around two meters, but that is still not acceptable for this application.

Table 1: Typical Errors for Standard GPS.

Error Source	**Typical Error in Meters**
Satellite Clocks	1.5
Orbital Errors	2.5
Ionosphere	5
Troposphere	0.5
Receiver Noise	0.3
Multipath	0.6
Total	10.4

Another method of improving the accuracy of the measurement is through the use of differential GPS. Differential GPS works through the interaction between the satellites, a base station, and a rover unit. In this case, the rover unit is located on the robotic lawnmower. The base station, at a known location, calculates necessary corrections to be made to the signals received from the satellites and sends these corrections to the rover unit by radio. The rover unit uses the corrections and signals received to calculate a position that is accurate to within two centimeters, which is accurate enough for this application. With regard to differential GPS, there is a caveat: a latency error is created by the calculation time required to produce the error corrections. The magnitude of the latency error is proportional to the velocity of the rover unit. Since the top speed of the robot is less than six miles per hour, the latency error produced is not significant.

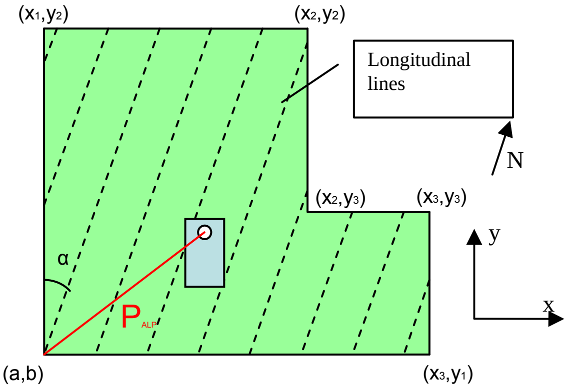


Figure 8: Field Navigation Coordinate System.

The differential GPS chosen is an OEM4 by NovAtel Inc. The GPS unit sends readouts of coordinates in a variety of coordinate systems. For this application, UTM coordinates are used. This is reported to the laptop, via RS-232 in binary format, along with status information regarding signal integrity and solution accuracy.

INERTIAL MEASUREMENT UNIT

The IMU chosen for this robot is a low-cost consumer IMU from Sparkfun. It has 3-axis MEMS gyros and accelerometers on board, as well as two 2-axis magnetometers. This unit is coupled with the GPS and is used to control turns and determine the heading of the lawnmower. The IMU reports information at a rate of 100Hz in the form of a simple string. This information is used to aid the GPS, allowing for a higher rate within the control loop, rather than being limited by the output rate of the GPS. The control loop is not limited to position control. Rate-based control is also being used on board the robot. Information from the gyros allows for constant rate turns and reduces the risk of getting stuck.

The magnetometers are used in conjunction with the gyros to provide stable heading information along with the heading vector from the GPS. The magnetic readings are in terms of field strength in x, y, and z axis. These values, along with the known magnetic deviation and dip in the local area, are used to convert the measurements into orientation information. This is using the Earth's magnetic field as a reference. Consequently, electromagnetic noise and disturbances can result in erroneous readings. To avoid this problem, the IMU is mounted away from any devices (such as the motors) capable of creating interference. All devices capable of creating a magnetic field disturbance are in the rear of the robot and the IMU is located in the front. The presence of ferrous materials can also cause a deviation in the reading; therefore, the IMU is mounted to the aluminum frame of the robot with non-ferrous fasteners.

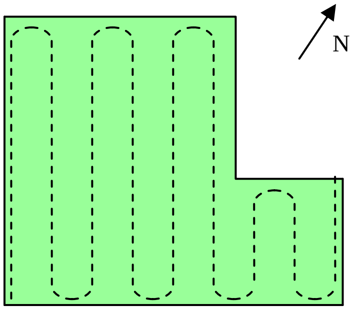


Figure 9: Line Following.

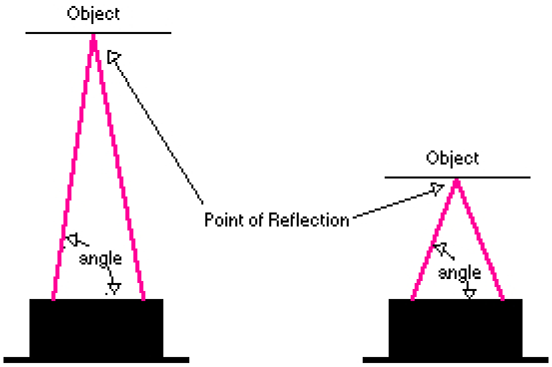
NAVIGATION ALGORITHM

For the navigation algorithm, the heading information gathered is normalized relative to the field so that zero degrees lines up with the lines the robot must follow as shown in Figure 8. A linear function is used to determine the direction that the mower needs to point based on the difference between the robot's location and the line it is following. Then, a proportional-plus-derivative (PD) control algorithm determines the speed of each motor to attain the correct heading referencing the difference between the desired heading and the heading indicated by the compass. For example, if the robot is to the left of the line it should follow, it is instructed to turn to a new heading to the right. The linear relationship determines how far right the new heading should be, given the deviation from the line it is following. The PD algorithm is then used to determine the speed difference between motors required to obtain the new heading. In this case, the left motor is instructed to go faster than the right motor, causing the robot to turn right. Line following works great for an empty field, but, in the event that there is an obstacle in the way, the robot needs the ability to identify it.

COLLISION AVOIDANCE

To detect and identify and obstacle, an additional sensor is needed. On this robot, a MWIR sensor is used to detect an object within a certain range. For this competition, the goal is to detect the trash can that has been placed in a non-deterministic location within the mowing area. The sensor is placed upon a 2-axis gimbal and continuously panned until an object has been detected.

This MWIR sensor is a Sharp's Gp2D12J0000F. It emits a pulse of IR light that is very slightly skewed towards the center of the sensor. That pulse continues until it reaches a surface where the light then scatters. Some of these photons will be reflected back to the IR sensor. When the nearest surface is significantly far away, there will be insufficient light returned to the sensor, and it will not detect any object. However, for an object that is within range of the emitter, that light will be reflected back in sufficient strength to the receiver. Since the original pulse was skewed towards the center of the sensor, the returned light will be reflected back onto the receiver portion of the detector. As illustrated in Figure 10, the path of the pulse creates a triangle with the detector.



The receiver consists of a linear CCD array. The receiver contains a lens that transmits the reflected light onto portions of the CCD array. The detector then uses information from each CCD to calculate the angle that the emission pulse formed with the front surface of the detector. Once that angle has been determined, the linear distance to the object is then calculated.

This type of sensor has several advantages over traditional Sharp's IR sensors. The first is that it is extremely tolerant of ambient noise (i.e. sunlight). For this application, sunlight is a definite concern, and this particular sensor is nearly immune to its effects. The second advantage this sensor boasts over previous Sharp's sensors is that, in addition to object detection, it also provides range estimation. This additional information provides our software with a much better mapping of where the object exists within the cutting field.

This sensor requires several other pieces of hardware to function. The first is an IR Distance Adapter which interfaces with an analog input on a Phidget Interface Board. Both of these are contained within sealed project boxes. Lastly, the gimbal is controlled by a PhidgetServo 4-motor board. This board is also contained within a sealed project box.

The entire sensor package must also be able to deal with the temperature it will be exposed to. The MWIR sensor's operating temperature is 14˚F to 140˚F. Since this component is not enclosed, it is not expected to be exposed to temperatures this high. The rest of the components do not have operating temperatures specified in their technical data sheet. However, given past experience with these components, they should easily withstand the reasonable range of temperatures that can be expected during the competition.

The distance sensor package must also be able to cope with rainfall during the course of the competition. The only exposed components are the MWIR sensor itself and the gimbal. The MWIR sensor has a single exposed surface; that surface is bolted to an aluminum block with the edges sealed. The 3-wire connection is, itself, air-tight. The gimbal consists of two servos, both of which are water-tight. The rest of the components are in sealed project boxes and are thus not expected to have any issues relating to precipitation.

SOFTWARE

Due to the software team consisting of a single person, most collaborative software development models did not apply to this project. However, many common practices have been observed (functional unit testing, code reviews, etc).

Despite having code from previous years, the code base was built entirely from scratch for several reasons. The first is that documentation of previous years' code was extremely sparse. There were very few user's guides and no developer's guides. In addition, there were several unidentified code bugs that had significant impact on the last two years' competition performances. Lastly, many of the low-level classes were not fully developed. This led to high-level classes that were implementing low-level logic to accomplish their task.

The end result is that low-level components were developed first and thoroughly tested. Only when the testing was completed was a higher-level class built using the lower-level class. In addition, whenever additional functionality was needed, it was checked on whether the same-level classes needed identical functionality. If so, that functionality was built into the lower-level class and was, again, retested before development of the higher-level classes resumed. This has led to a hierarchy of classes where common functionality has been pushed to the lowest common level.

Initially, modules for individual components were created. This allowed us to become familiar with each individual component and verify that they were functioning properly before they were integrated into system. This approach led to the discovery of several components that were intermittently failing. These components were subsequently replaced before development continued.

Since nearly all the sensors, as well as the motor controller, require serial communication via an RS-232 link, the first class to be created was the Serial class. Its job is to handle low-level communication for the serial port. This class implements serial communication that is compatible with all POSIX operating systems. It is based upon the documentation described in the publication "Serial Programming Guide for POSIX Operating Systems."

This class controls the communication with the IMU. It allows both two connection methods. The first is via Bluetooth. When connecting with this method, this class scans for Bluetooth devices and then checks the MAC addresses of potential connections. This ensures that it connects to the appropriate device (since *all* Bluetooth-enabled devices, not just IMUs, will respond). The second mode, which is an RS-232 connection, does not have any authentication checking as RS-232 is a physical connection.

The next step in the process is that the IMU class requests samples at 100Hz from the IMU. The incoming data is then parsed and the mower's status is updated as described in the Navigation section.

The Roboteq class's main purpose is to provide signals to the motor controller, which in turn will control the motors that will drive the rear wheels. Periodically (i.e. several times a second), the AI will have the single Roboteq object send desired speed commands for each motor to the motor controller.

The feedback from the motors is read by the motor controller and power adjustments are made directly by the motor controller itself. Therefore, there is no need for the Roboteq class to handle encoder feedback and attempt to readjust power settings to compensate for terrain.

This class is a wrapper around Phidget's development library. The Phidget developmental library presents an API that is meant for general use for its components, but is somewhat clumsy to utilize. So, the RangeFinder class is aimed at hiding several of the ambiguous and/or irrelevant details of the underlying architecture and tailors it for the MWIR sensor. In this manner, the programmer is presented with a much more user-friendly interface.

The Servo class for the gimbal is analogous to the RangeFinder class for the MWIR sensor. Again, by creating wrapping functions that hide the general use of the servo controller, the user has a much more user-friendly interface that enables quicker development of the higher-level modules.

This class's purpose is to get data from the Novatel GPS unit(s). This class is capable of sending a single request for continuous data at around 15Hz, as well as sending a request to get a single position reading. However, our navigation techniques have GPS correcting the IMU readings, so the GPS class requests single readings periodically as needed.

INTEGRATION

Before integration of individual components could begin, there needed to exist a framework that they would integrate into. The overarching idea is that there should be a thread for every sensor. This, therefore, includes input from GPS, input from IMU, and input from the MWIR sensor. In addition, there should be a single controlling AI that looks at the current available navigation information and makes a decision about both how to drive the two motors and where to point the MWIR sensor. This AI inherently needs its own thread as it needs to operate independently from all other processing.

Communication between the sensor threads and the AI thread would be through a class called 'TankStatus'. This class contains the latest navigation and MWIR information that is available. This class is, thus, a critical resource whose data members should be accessed/modified by only a single thread at any given time. To ensure the mutual exclusion safety property, each object of this class contains a mutex.

COST

The cost of all the equipment that makes up the lawnmower places the price tag over $6,000. The IMU, Novatel OEM4 GPS receivers, RoboteQ motor controller, and laptop are all high-end electronics. All of these systems could be replaced with less expensive equipment in the future. The cost of the laptop can be removed by replacing it with a field-programmable gate array (FPGA) or board computer. The Novatel OEM4 is a commercial/surveying GPS unit and is one of the largest costs for the production of this prototype lawnmower, especially since two receivers are needed for differential mode. Different systems have been proposed to replace the GPS in the design. However, the Novatel OEM4 GPS system was already owned by the team and is a very good system to use due to its accuracy. The last of the expensive components is the RoboteQ motor controller. It is capable of delivering five times the current the motors can handle. It could be replaced with a lower cost motor controller that is designed to fit our needs, and the feedback can controlled with a programmed microcontroller rather than having it built into the motor controller.

CONCLUSION

Mowing a field of grass is a tedious task that the average person prefers to avoid. There are two options to eliminate this task from a person's schedule. One is to pay someone else to perform it; the other is robotics. There is a simple robotic lawnmower on the market, but it is unable to perform well in fields that are of complex shape or substantial size. The robotic mower in this paper is a more sophisticated approach using GPS navigation and is capable of mowing fields of large size and complexity. It is equipped with a motor controller that drives two motors based on commands sent by the computer. The computer sends the commands to the motor controller based on data received from an array of sensors, including a differential GPS, IMU, magnetometer, and IR sensors. With all these systems, the robot is an autonomous lawnmower capable of navigating a field, avoiding obstacles, and cutting the grass.

REFERENCES

1. Arlinghaus, Mark. The Raider Rover. Wright State University. 2006.
2. Banner Engineering Co., "Product Line Specifications forVALU-BEAM Sensors," 10 Jan. 2007. Internet: <http://info.bannersalesforce.com/xpedio/groups/public/documents/ datasheet/32894.pdf>.
3. Craig, John J. "Introduction to Robotics, Mechanics and Control 2nd ed.," Reading, MA, Addison Wesley Publishing Company, 1997.
4. "DR Power Equipment." Neuton. 27 September 2006. <www.Neutonmowers.com>.
5. "Introduction to T-Slotted Framing." 80/20 Inc. 2005. 29 September 2006. <www.8020.net>.
6. Kuo, Benjamin C. and Golnaraghi, Farid, "Automatic Control Systems," 8th Edition, John Wiley and Sons, Inc., 2003.
7. "Midwest Motion Products." Midwest Motion Products. 2 October 2006. <www.midwestmotion.com>.
8. NovAtel Inc., "Product Specification for OEM4-G2," 6 Jan. 2007. Internet: <http://www.novatel.com/Documents/Papers/oem4g2.pdf>.
9. "NPC Robotics." NPC Robotics. 29 September 2006. <www.NPCroboitcs.com>.
10. OceanServer Technology Inc., "Digital Compass User Guide," 10 Jan. 2007. Internet: < http://www.ocean-server.com/download/Compass_Manual.pdf>.
11. RoboteQ Inc., "User's Manual for RoboteQ AX2550/2850 Dual Channel High Power Digital Motor Controller," 12 Jan. 2007. Internet: <http://www.roboteq.com/documentation .html>.
12. Shigley, Mischke and Budynas. "Mechanical Engineering Design." 7th ed.
13. Trimble Navigation Ltd., "All About GPS," 10 Mar. 2007 Internet:<http://www.trimble.com/gps/>.