

REDBLADE: MIAMI UNIVERSITY'S MULTI-FUNCTIONAL AUTONOMOUS ROBOT

Ryan Wolfarth, Steven Taylor, Aditya Wibowo, Brandon Williams, Yu Morton, Peter Jamieson

Miami University

ABSTRACT

RedBlade is a multi-functional autonomous vehicle with two seasonal configurations which allow it to plow snow in the winter and mow grass in the summer. This vehicle participated in the Institute of Navigation's (ION) 1st Autonomous Snowplow Competition in January 2011 and in the 8th ION Robotic Lawnmower Competition in June 2011. It won 2nd place in the former. This report presents the design and implementation of the RedBlade mechanical systems, sensor components, software architecture, control algorithm, and safety systems.

1. INTRODUCTION

Autonomous vehicles capable of performing many functions with accuracy and reliability in a timely manner are highly desired in modern society. RedBlade is designed as an expandable host to perform in multiple roles. It represents the next stage evolution of an autonomous lawn mower used in the ION Robotic Lawn Mower Competition. Since its inception as an autonomous lawnmower in 2004^[1], RedBlade has been enhanced with the additional function of plowing snow. This paper describes RedBlade's mechanical, sensor electronics, control algorithms, and safety mechanisms required by the competitions.

RedBlade features a three-layer system architecture that is abstracted in Figure 1. The top layer is the navigation and obstacle avoidance sensor suite. The current generation of the RedBlade navigation sensor suite includes a Topcon Hiper Lite Plus GPS receiver, a MicroStrain 3DM-GX2 inertial sensor, and two optical wheel encoders as part of the integrated motor drive system. The current obstacle detection sensor suite uses micro-switch based touch sensors and an 180° SICK scanning laser. Additional sensors such as a stereo vision system, a second GPS unit on-board the rover for future deployment.

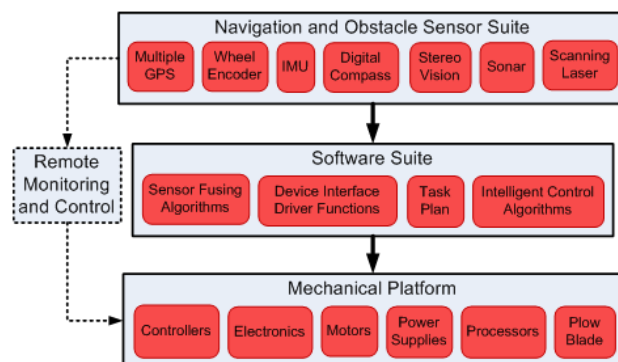


Figure 1: RedBlade three-layer system abstraction.

The middle layer is the collection of software that provides driver functions for the sensors, sensor fusion algorithms, path planning, and vehicle motion control algorithm. The bottom layer is the mechanical platform, electronics hardware, including the motor controller, safety systems, power supplies, and processors that carry out the software functions.

RedBlade's objectives are to compete in the ION Autonomous Snowplow Competition (ASC) and the ION Robotic Lawnmower Competition (ALC)^[2]. Each competition has a unique set of rules that the vehicle must follow. For the snowplow competition, RedBlade must be able to plow two path shapes: one straight line "I" shape, and one block "U" shape. The "U" shape field can be seen in Figure 2.

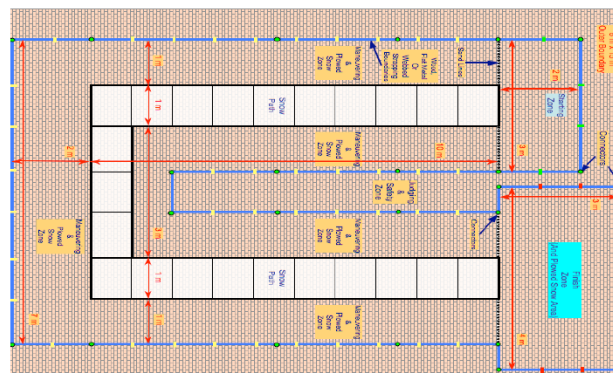


Figure 2: ION Snowplow Competition field diagrams. Each plowing path is 1 meter wide.

The next problem statement is defined by the rules of the lawn mower competition: RedBlade must be able to mow grass in two different 150 square meter fields of play^[3]. The first course is an open lawn with one randomly placed static obstacle which should be avoided. The second course contains a number of additional static obstacles and one moving obstacle that may appear at certain points on the map. This second scenario is depicted in Figure 3.

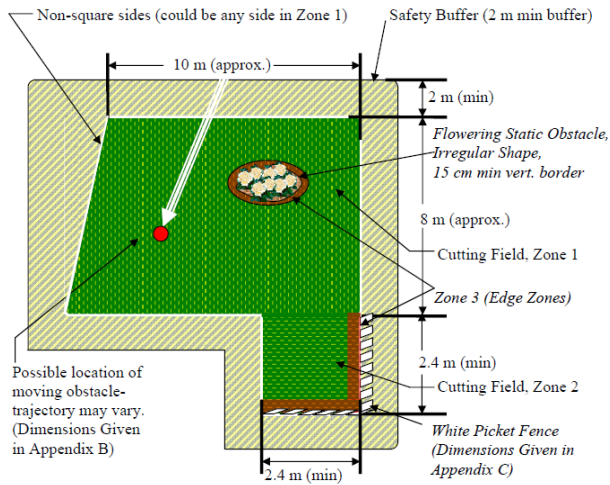


Figure 3: ION Lawn Mower Competition dynamic field diagram. Note that this is only an example of the possible field configuration.

RedBlade utilizes the three navigation sensors (GPS, IMU, and optical wheel encoder) to determine its position, heading, and velocity (PHV). The vehicles PHV information along with its predetermined destinations are input to an on-board computer that implements a Proportional-Integral-Derivative (PID) control algorithm to adjust vehicle heading. Waypoints are adjusted as obstacles are found to ensure the most complete coverage of a designated area. The lawnmower blade and engine were taken from a Black & Decker CMM1200 electric mower. Both remote and on-board emergency kill switches allow an operator to stop all robotic motion.

The remainder of this paper is organized as follows:

- Section 2: Mechanical Platform.
- Section 3: Electrical Components.
- Section 4: Control Algorithm Design and Implementation.
- Section 5: Path Planning.
- Section 6: Obstacle Detection and Avoidance.
- Section 7: Test and Performance Evaluations.
- Section 8: Conclusions and Future Work.

2. MECHANICAL PLATFORM

RedBlade's mechanical platform consists of two drive wheels, three caster wheels for stability, and a metal chassis that houses the electrical systems. An overview of

the mechanical platform for the wheels can be seen in Figure 4. A top and side view of the vehicle in each configuration can be seen in Figures 5 and 6.

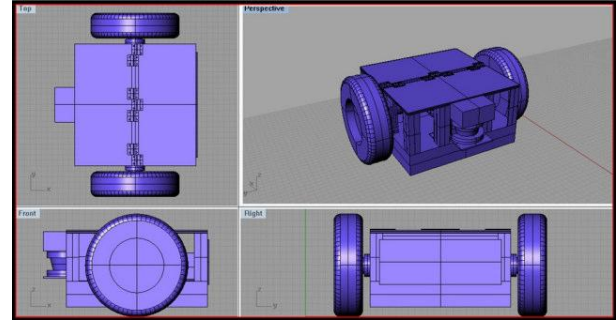


Figure 3: Drawings of the wheels and casing.

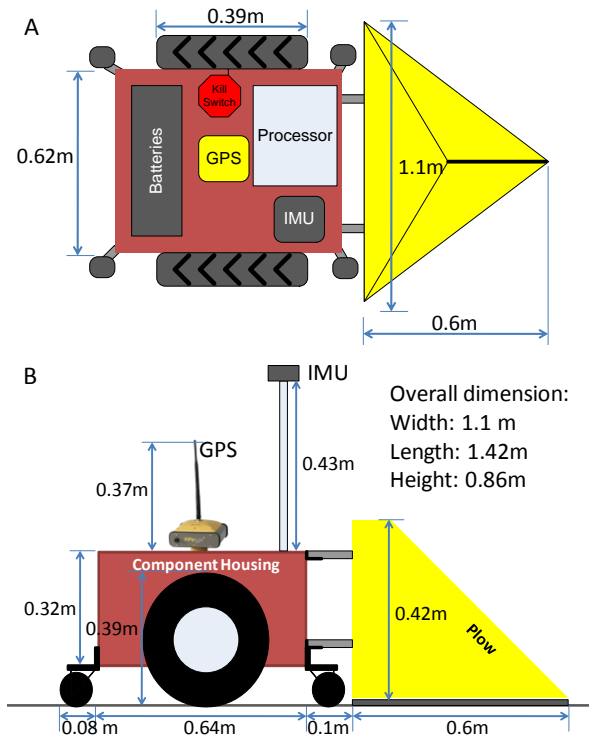


Figure 4: Top (A) and side (B) view of the robot in the snowplow configuration.

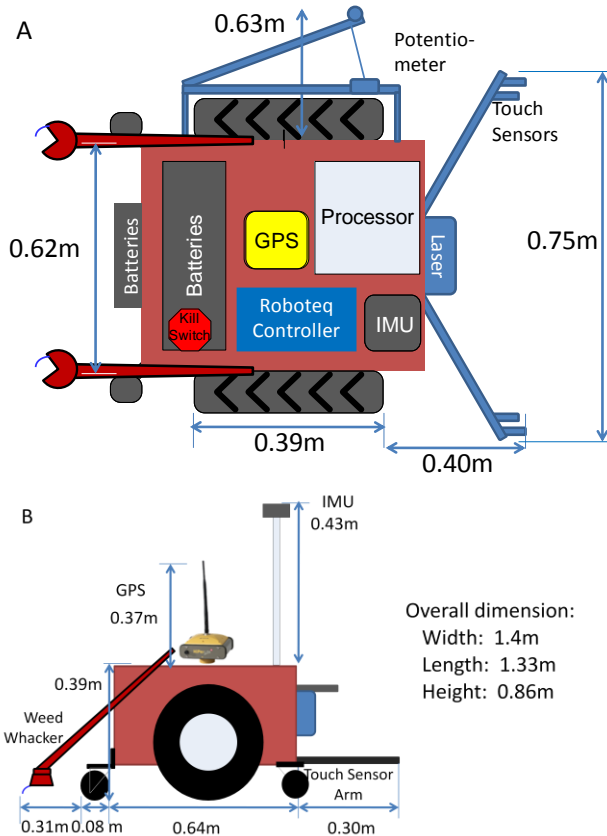


Figure 5: Top (A) and side (B) view of the robot in the lawn mower configuration.

The robot is driven by two 24 volt electric motors that each output 1.5 horsepower through a 20:1 reduction gearbox^[4]. Four spring-loaded casters mounted on the four corners of the casing were inherited from the previous RedBlade lawn mower platform, but testing showed that these caster wheels were problematic when the robot encountered small ledges, tree roots, and other terrestrial obstacles. Our research led us to the compromised, minimal modification solution of replacing the front two caster wheels with a single, more robust mount and wheel attachment in the front-center of the robot. The rear two spring-loaded casters were retained. Field tests indicate that this change afforded the robot increased mobility, but many soft terrain surfaces (mulch, depressions, etc.) continue to hinder performance.

The original drive wheels proved to provide too little traction in some dry and all wet conditions. These wheels were replaced with auger-style wheels which can be seen in Figure 6. Increased traction was proven through experimental testing.

3. ELECTRICAL COMPONENTS

RedBlade's platform houses a number of electrical and electronics components including batteries, safety switching circuits, a motor controller, and the entire

navigation sensor suite. This section presents details of the electrical system. A high level system connection diagram can be seen below in Figure 6.

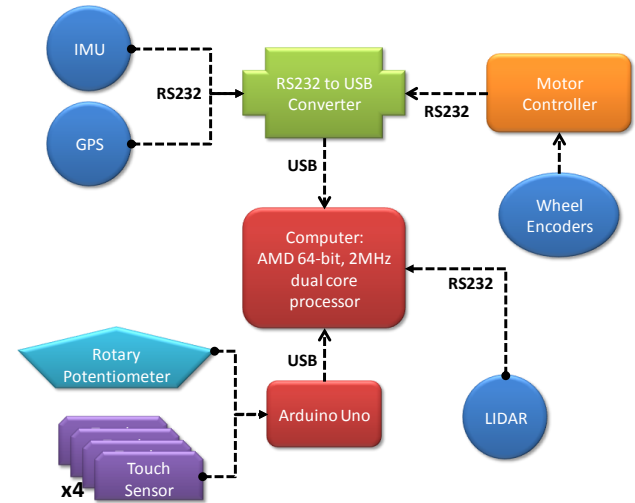


Figure 6: High level system connection diagram. Any unlabeled connections are basic signal carrying wires.

3.1. Power Supplies

Three banks of lead-acid batteries provide power to different sections of the system. Two 24V banks provide power to the drive wheels and blade mower respectively, while a third 12V bank powers the computer and other sensor components. Note that the Differential GPS system features internal batteries that are not part of the vehicles power supply.

To ensure safe operation and maintenance of the power system non-conductive plastic was placed between the battery housing and the upper equipment mount that support the computer and other control components. Early experiments showed that some systems operated erroneously when the supply voltage fell below a threshold. An additional set of batteries was purchased to facilitate prolonged testing and run times. A circuit diagram for the power supply is shown in Figure 7. Note that the safety system is integrated into the power circuit.

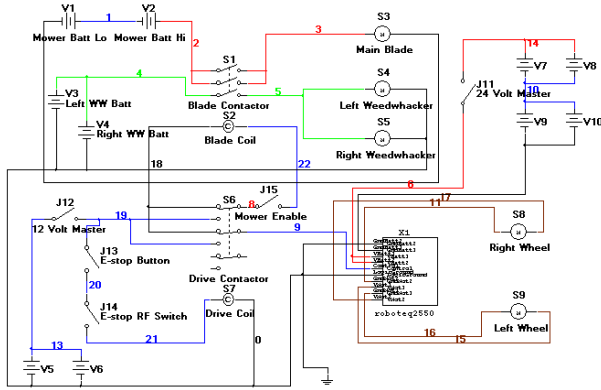


Figure 7: Circuit diagram of RedBlade's power and safety system.

3.2. Processors, Controllers, and Hard Drives

The main system control is operated by a PC running a Linux installation. We communicate with this device via direct connection or through an on-board wireless router.

All system processes are controlled by the onboard PC running a Linux installation. Communication with this device is accomplished via direct connection or through an on-board wireless router.

Because RedBlade was required to function in a vast range of environments, weather-proofing was required to ensure safe and reliable operation. A standard hard drive contains components that are likely to freeze in low temperatures. RedBlade uses a solid-state drive (SSD) to mitigate this risk. In addition to having better temperature endurance, the SSD is able to withstand much higher degrees of vibration and impact. Power consumption is reduced 85% from approximately 20 Watts to no more than 1.7 Watts.

A RoboteQ AX2550 controller was used to drive the motors. This controller is capable of directing 6 times the power that we require and has several built-in protection modes. We limit the controller's output to 20Amps in order to protect the 14 gauge wire that is used for power transfer to the motors.

An Arduino Uno microcontroller board was selected to monitor the touch sensors for obstacle detection. The board poles all five touch sensors at 10Hz and gives the resulting measurements to the main computer for obstacle detection processing. Figure 8 shows the circuit interfacing between the Arduino and the touch sensors. All inputs of prefix "Din" are digital inputs from the micro-switch based touch sensors. The analog input A2D0 reads analog voltage from the rotary potentiometer on the lateral sensing arm.

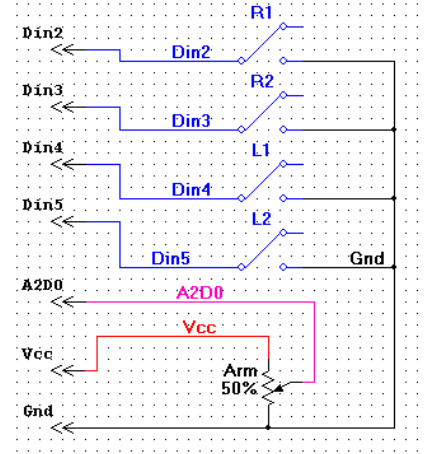


Figure 8: Arduino interface with touch sensors.

3.3. Safety System

The RedBlade platform has two emergency stop options: remote control and an on-board stop button. Our tests show that the stopping distance from a maximum speed of 2 m/s varies depending on the type of surface tested:

- Icy Surface: 0.5m
- Concrete: 0.3m
- Rough Brick: 0.2m
- Dry Grass: 0.3m
- Wet Grass: 0.4m

Each emergency stop relay can be seen in Figure 7 where manual and remote stops are labeled as J13 and J14 respectively.

3.4. Navigation Sensors

A MicroStrain 3DM-GX2 IMU is used to determine the vehicle heading ^[11]. It has an adjustable data rate to facilitate interfacing with different clients. Calibration for the device can be completed using the Hard Iron Calibration tool in MicroStrain's Data Acquisition and Display software ^[12]. While this particular guide is for the older GX1 model, the process is similar. This IMU was shown to accumulate approximately 2° error for every 3 meters of linear travel. These tests were conducted among limited ferromagnetic materials in order to glean maximum accuracy from the IMU. Urban environments or other areas with dense concentrations of ferromagnetic material would cause the device to perform less accurately. On site calibration is necessary to mitigate errors from the IMU while the robot is in the field of play.

The Hiper Lite Plus is a survey grade dual-frequency differential GPS system by Topcon. Field tests near Miami's Engineering Building with masking angle at 30° on one side shows location accuracy within 2cm as specified by the device manufacturer ^[13]. The raw geodetic coordinates given by the Hiper Lite Plus receiver

are converted to an ENU local coordinates system before being sent to the control algorithm. The origin of the local coordinate system is the beginning of the path (where the robot is to begin its job), while the robot's initial heading points to the local y-axis. Custom driver software was written to allow the GPS receiver to send its position measurements to the on-board processor.

Two US Digital E7MS quadrature optical encoders are installed on both left and right wheels of the vehicle. Each encoder sends its signal on two different channels with 90 degree offset. By using two channels it is possible to determine the direction of movement if there is no slippage. When the robot is moving forward, one channel emits a pulse before the other. By counting the pulses sent from each encoder, we are able to determine the number of revolutions, therefore allowing us to determine the distance traveled.

In order to assign a control constant to the wheel encoders we must first be able to determine how much drift the encoders have relative to each other. This is accomplished by comparing the output from each encoder after turning each wheel an equal distance. Comparison between each wheel output allows us to assign a control constant based on how much drift is observed.

Each sensor may provide inaccurate data depending on the condition of the robot. For example, if RedBlade is traveling very slowly, two successive DGPS measurements may not provide an accurate heading. Sensor fusion with the wheel encoders or IMU provides an accurate heading since they do not depend on speed of travel to yield a good measurement. Conversely, encoder or IMU error due to excessive wheel slippage or magnetic activity respectively are mitigated with a DGPS heading measurement provided the system is still traveling at an acceptable speed.

4. CONTROL ALGORITHM DESIGN & IMPLEMENTATION

A PID-based feedback algorithm is used in both RedBlade configurations. This algorithm adjusts wheel speeds based on present and past heading errors. The algorithm accepts a waypoint vector as its inputs. We designate the end of each straight path as a waypoint. For example, in this “U” shaped path (encountered in the plow competition), there are three waypoints as shown in Figure 9. We will use this “U” shaped path as an example while explaining our control algorithm.

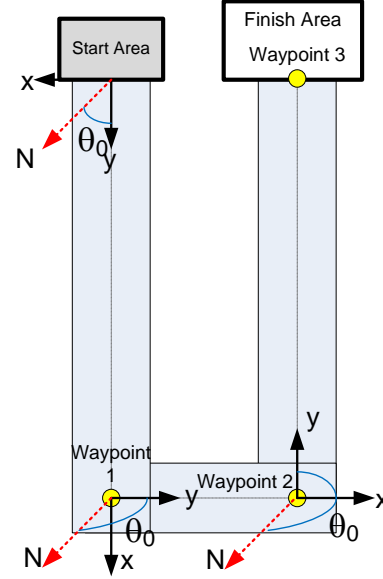


Figure 9: Shows the three waypoints and the local coordinate system associated with each path segment.

We divide the “U” shaped path into 3 segments:

- Segment 1: starts at the start, ends at waypoint 1.
- Segment 2: starts at waypoint 1, ends at waypoint 2.
- Segment 3: starts at waypoint 2, ends at waypoint 3.

For each segment, we define a local coordinate system whose origin is located at the starting point of the segment and the local y-axis is along the path direction, while the x-axis is perpendicular to the y-axis as shown in Figure 7. The waypoint vector for this “U” shaped path is defined as:

$$W = \{[\theta_0(1), L(1)]; [\theta_0(2), L(2)]; [\theta_0(3), L(3)]\} \quad (1)$$

θ_0 is the path direction relative to true north, and L is the path length. At each segment of the path, the vehicle's objective is to reach the point $[0, L]$ while trying to maintain a heading direction of θ_0 . Clearly, this W vector is expandable if a more complicated path is required.

Figure 10 explains how this objective is achieved. While traveling along a segment of the path at time t , the GPS input shows that the vehicle is located at some intermediate point (x, y) . We can compute the desired vehicle heading based on the current location (x, y) and the end point $(0, L)$:

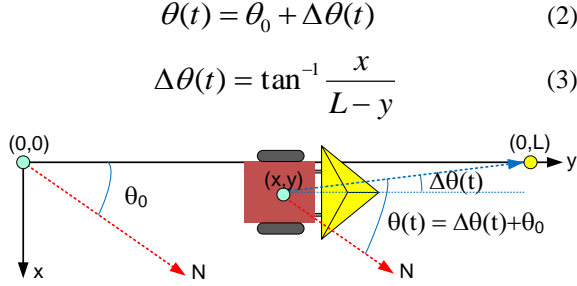


Figure 10: Schematics explaining the relative vehicle heading dependence on vehicle location and target location.

The $\theta(t)$ value is the desired set point value of the vehicle heading in order for the vehicle to reach the desired destination. The IMU measures the actual vehicle heading. The difference between the computed set point and the IMU measurement is the error input to the PID loop. By properly selecting the K_p , K_i , and K_d coefficients, the PID loop generates a signal that drives the two wheel motors to minimize the error, thereby forcing the vehicle stay on the path. Figure 11 shows the block diagram of the PID feedback loop.

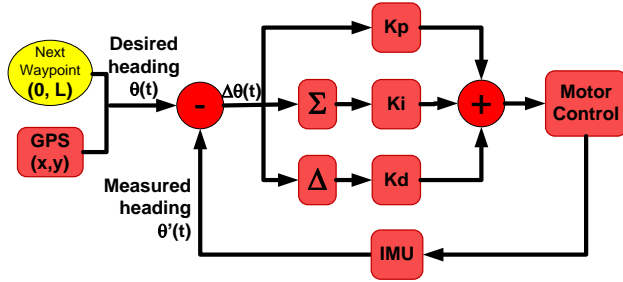


Figure 11: PID feedback control block diagram.

The above algorithm applies whenever RedBlade is moving between two points. Therefore, the entire procedure is repeated for each waypoint vector component. Figure 12 shows the top level block diagram that cycles through each of the waypoints and executes the PID for each path segment. Any number of path segments can exist in the field of play, so the path planning can be abstracted to allow navigation around more complicated paths.

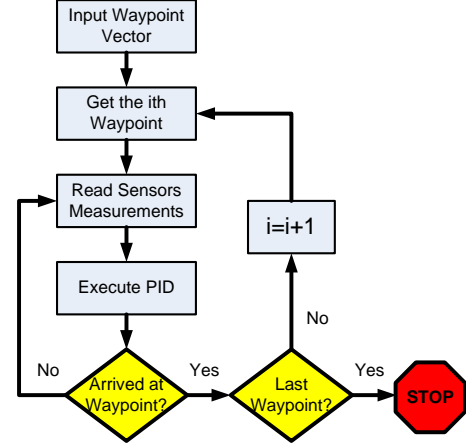


Figure 12: Block diagram of vehicle operation over the entire path.

System stability and response is dependent on the selection of the aforementioned control constants. We experimentally derived these constants through the use of the Ziegler-Nichols tuning method^[14]. This method requires that the K_p constant be set such that the system is put into sustained oscillation without becoming unstable. Herein, this maximum K_p will be referred to as K_U . The period of these sustained oscillations is defined as T_U . The three control constants are derived from the equations listed below:

$$K_p = 0.6 \cdot K_U \quad (4)$$

$$K_i = \frac{2 \cdot K_p}{T_U} \quad (5)$$

$$K_d = \frac{K_p T_U}{8} \quad (6)$$

Figure 13 shows the data plot of IMU heading versus time that was used in deriving our control constants. The Fourier transform of this data set gave the data shown in Figure 14. The same plot is visible in Figure 15 with only the area of interest shown.

It should be noted that there are two peaks in Figure 13 that suggest two possible values of T_U . The value for T_U derived from each peak and the resulting control constants were empirically tested. Our final control constants were derived from $T_U = 2.14$ seconds. The criterion for these tests was that the system should travel as straight as possible over a given distance while still suppressing errors caused by lack of traction, unbalanced terrain, etc. It is necessary to retune these constants whenever the terrain changes. The aforementioned constants were calculated for use in the lawn mower competition.

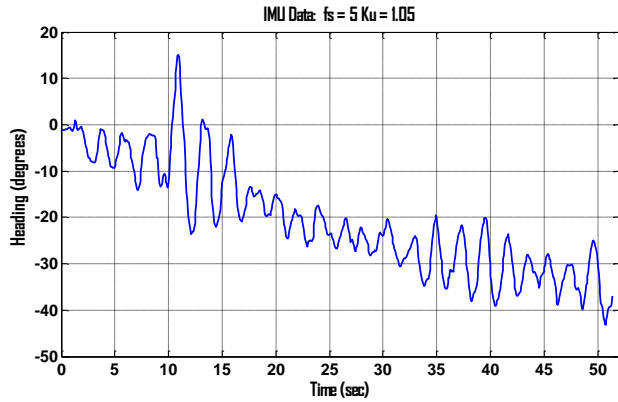


Figure 11: IMU heading versus time. This data was used to empirically derive the control constants.

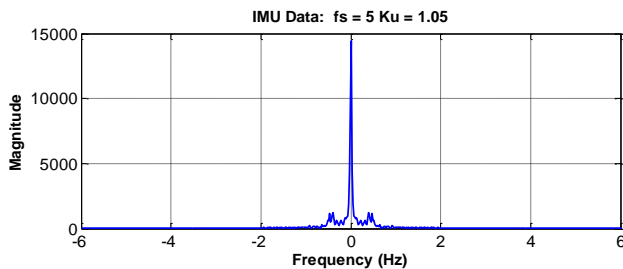


Figure 12: Fourier transform plot of Figure 11 data set.

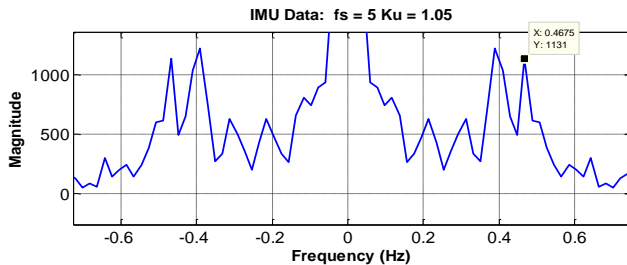


Figure 13: Peaks of interest from the Figure 12 plot. Notice the two peaks that suggest two possible values of T_U .

5. PATH PLANNING

RedBlade incorporates an adaptive path plan that modifies its waypoints as static obstacles are detected. It is important to note that this is only pertinent in the lawn mower configuration: the snowplow configuration does not have the obstacle detection capability as there is no obstacle placed in the plow path. The path plan for the snowplow configuration need only follow the surveyed waypoints sequentially: no intermediate waypoints need to be calculated. Each path plan is described in more detail below.

5.1. Path Plan: Snowplow Configuration

When configuring RedBlade for operation in the snowplow configuration the user only need survey the corners of the field. The width of the plow is 10cm wider than the path width, so only one pass is required. This

fact allows RedBlade to simply travel to the corners of the surveyed area in order to complete its task. The flow chart in Figure 14 depicts the method by which RedBlade follows a set of given waypoints. This example traverses waypoints 1 through 3 as seen in Figure 9.

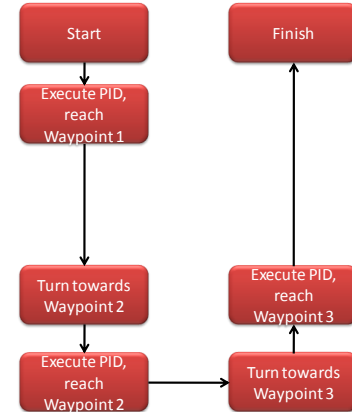


Figure 14: Example flow diagram of the path progress used to traverse the snowplow field seen in Figure 9.

5.2. Path Plan: Lawn Mower Configuration

In the lawn mower configuration, the user must survey all corners of the field that should be mowed. RedBlade computes and plots intermediate waypoints which ensure the entire field is mowed. This is done assuming a conservative cutting surface width of 0.25 meters.

RedBlade uses a heuristically derived approach to mow the competition field. By pointing it directly at the flower bed obstacle, RedBlade is able to detect and mow the circumference of the bed first. This gives the location and shape of the obstacle. The intermediate waypoints are calculated to avoid the flower bed in the remainder of the path. An example of a plotted path after flower bed obstacle was detected and mapped is shown in Figure 15.

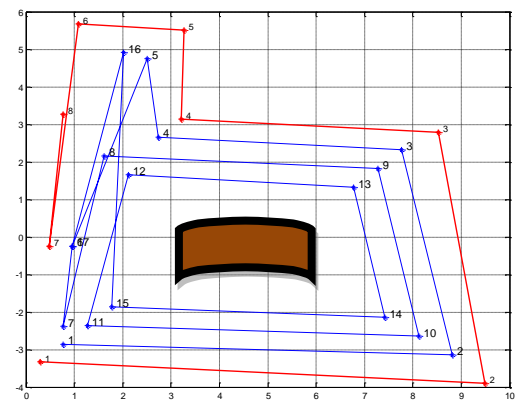


Figure 15: An example of how RedBlade modifies its path plan to avoid static obstacles.

The example above assumes that the area which is uncovered by the intermediate waypoints has been sufficiently mowed during the mapping of the flower bed. This allows the path planner to simply “box out” the obstacle with a safety zone to ensure the vehicle will not come within close proximity.

6. OBSTACLE DETECTION & AVOIDANCE

As stated in the rules for the ION Autonomous Lawnmower Competition^[3], the advanced category had several obstacles to avoid: a fence, a flower bed, and a stuffed poodle mounted on an R/C car. The first two fall into the category of static obstacles, and had the goal of mowing around them, but not moving/damaging them. The last obstacle fell into the category of dynamic obstacle, with the goal being simply to avoid it.

The two means of obstacle detection use a SICK Light Detection And Ranging (LIDAR) sensor and several forward mounted micro switch based touch sensors. A third touch sensor is used when RedBlade is in obstacle “tracking” mode. All three sensors can be seen in Figure 16. The LIDAR was the primary means of dynamic obstacle detection, while the touch sensors were used to detect static obstacles.

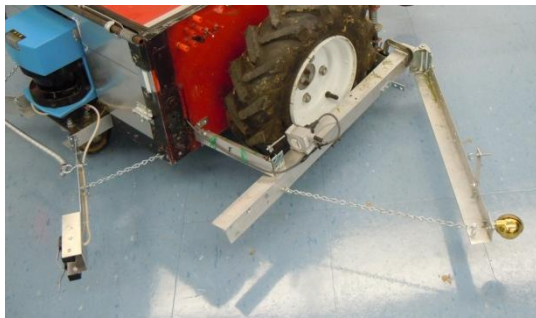


Figure 16: Laser and touch sensors used for obstacle detection.

6.1 Detecting Static Obstacles

The only two static obstacles in the lawn mower competition were the picket fence and the flower bed. The picket fence was placed along the boundary line mapped by the survey points (see Figure 3). Our approach to avoiding this obstacle was to program the path planner to avoid tightly spaced corners (as found in the fenced in area) by only mowing a portion of the encompassing area. This was necessary because the guide arm used to track and follow static obstacles would get caught in the picket fencing.

Mapping and tracking the flower bed was accomplished exclusively with touch sensors. The forward sensors were activated when the flower bed material made contact with the switches. This triggers an obstacle tracking sequence that uses the lateral guide arm to keep a predetermined distance from the surface of the flower bed.

This tracking mode first orients RedBlade such that the guide arm is displaced a small amount against the flower bed. Next, the local ENU coordinate of the vehicle is stored for later reference. The mode then engages a proportional feedback loop to move the robot forward while keeping it a predetermined distance from the obstacle. When the vehicle returns to its original ENU location, the tracking mode disengages and RedBlade proceeds to an outlying waypoint. This is done to allow clear access to the newly calculated intermediate waypoints which will avoid the newly mapped obstacles location.

6.2 Detecting dynamic obstacles

The dynamic obstacle (stuffed dog) in the competition would enter the field perpendicular to the mower's velocity vector while the mower is moving straight, and at least two meters away from any static obstacles. This solves the problem of needing to detect the dynamic obstacle while also detecting a static obstacle, and gives a narrow window to need to detect the dynamic obstacle.

The LIDAR is able to see 180° in front of the vehicle and processing data at 3Hz. It works by sending and receiving a laser pulse once every 0.5° around the range of sight. Distance of objects in the field of view is determined by multiplying the time between transmission and reception of a pulse by the speed of light. The successive pulses around the field of view can build a two dimensional picture of the surrounding environment. A rudimentary example is shown in Figure 17. This results in a large amount of data, since each complete laser scan requires 361 values, each of which is a 64-bit floating point integer. Testing and optimization of this data rate is discussed later in Section 7.

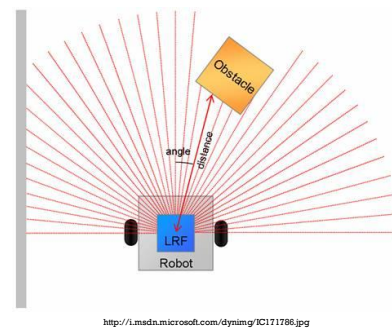


Figure 17: An example of the LIDAR as an obstacle detection system.

The LIDAR on RedBlade is configured to ignore anything farther than 2 meters away. This was done to prevent false detections of spectators since they are allowed to be as close as 2 meters to the competition field. When a dynamic obstacle is detected, the robot stops until the obstacle leaves. Dynamic obstacle detection is only

active when the vehicle is not near the fence since it would be misconstrued as a dynamic obstacle.

7. TESTING & PERFORMANCE EVALUATIONS

The completed vehicle in both configurations can be viewed in Figures 18 and 19. These configurations were tested for performance and the results are discussed below. Testing was done in parallel and in multiple stages because components were often added simultaneously.



Figure 15: RedBlade in the snowplow configuration.



Figure 15: RedBlade in the lawn mower configuration.

7.1 DGPS Characterization & Performance

The manufacturer of the differential GPS system claims it to have no more than 2cm error in a Real Time Kinematic (RTK) configuration. This was confirmed by operating the system in RTK mode with the base station and rover stationary. This was also confirmed while mounted to the robot and moving in a straight line. There were points lying outside of the 2cm error margin, but these were accounted for due to the vibration induced by the rest of the system.

A high masking angle and poor satellite geometry sometimes hindered system performance. The Topcon DGPS system can track L1 and L2 GPS and GLONASS signals for use in its dual frequency position tracking and position solution. However, conditions in the urban environment at the snowplow competition sometimes caused loss of lock for multiple satellites which resulted in a bad position solution. When this would happen, the Topcon system would provide an error signal to RedBlade, which would then stop and wait for a valid position solution.

7.2 LIDAR Characterization & Performance

As mentioned previously, the LIDAR outputs 361 samples every image with an imaging frequency of 3Hz at 64-bit floating point resolution per sample. This was the maximum imaging frequency we could obtain due to hardware constraints in the computer. The system runs a multi-threaded architecture where each sensor requires its own thread to be processed. Most sensors like the guide arm do not take much time to read and can be sampled at a high data rate without any loss of overall system performance.

Due to the volume of data required by the LIDAR, sampling any higher than 3Hz resulted in missed detections. During the ALC, 3Hz proved to be too slow for detecting the dynamic obstacle, which became caught in the forward touch sensors. RedBlade did stop, but not in time. Having a dedicated image processor for the LIDAR to give report a binary detection state to the main computer would free up system resources and allow a higher imaging rate.

7.3 PID Considerations

Whenever the robot is to be operated on a new type of terrain, or the condition of the terrain changes, the PID constants must be returned to ensure motion system stability. In test we saw a pronounced instability in the motion control algorithm when the grassy surface became wet with dew.

7.4 Power System Constraints

Run time in each configuration is limited by different conditions. In normal temperatures with no plow attachment or blades turning, RedBlade can operate for approximately 90 minutes before the computer batteries require a new charge. This time was reduced to approximately 40 minutes in the cold environment of the snowplow competition.

In the lawn mower competition, the limiting factor was the battery life of the blade and string trimmer batteries. This was observed to be approximately 30 minutes depending on the density of the grass.

7.5 Safety System Performance

The safety system was required to cause a complete stoppage in movement in less than 3 seconds and in less than 2 meters. RedBlades system stops in approximately 0.5 seconds and in less than 0.5 meter in worst case testing.

7.6 Mechanical Limitations

The mechanical limitations of RedBlades platform prevent it from coming into close contact with surfaces that have sharp corners or other features that may become caught on the touch sensors or guide arm. The design compromise for this situation was to intentionally stay away from obstacles such as the picket fence in favor of mowing smooth faced obstacles like the flower bed with 100% accuracy.

8. CONCLUSIONS

This iteration of RedBlade was an eight month project undertaken by four undergraduate students at Miami University. They designed, built, implemented, and tested an autonomous vehicle which can plow snow in the winter and mow grass in the summer. RedBlade has demonstrated its ability to function as an autonomous vehicle in both of these configurations. This ability has been achieved through a navigation sensor suite, including a DGPS receiver, IMU, and wheel encoders, a PID-based control algorithm, and an in-house mechanical platform. Several failure modes have been taken into consideration and recovery actions have been implemented after more extensive testing.

The more long-term impact of this project is the valuable learning experience gained by the students working on the team. Students learned trouble shooting, managing deadlines under a tight schedule, and interfacing with parts and supply sources. They also learned specialized technical skills through this complicated project that required interfacing multiple components.

ACKNOWLEDGEMENTS

The RedBlade team would like to thank the Institute of Navigation Satellite Division for sponsoring the ASC and ALC and for the ION North Star and Dayton Section for organizing the competitions. The team received the funding support from Miami University Office for Advancement of Research and Scholarship, School of Engineering and Applied Science Dean's Office, the

Department of Electrical and Computer Engineering, and the Department of Computer Science and Systems Analysis. The Heritage Equipment Company is also thanked for the manufacture and donation of the plow shovel. Additionally, the team appreciates the technical guidance and support from Drs. Wouter Pelgrum and Frank van Graas of Ohio University and Mr. Jeff Peterson from Miami University.

REFERENCES

- [1] McNally, B., M. Stutzman, C. Korando, J. Macasek, C. Mantz, S. Miller, Y. Morton, S. Campbell, J. Leonard, "The Miami Red Blade: An Autonomous Lawn Mower," *Proc. 2004 ION Annual Meeting*, P538-542, Dayton, OH, Jun. 2004.
- [2] <http://www.automow.com>
- [3] <http://www.ion.org/satdiv/alc/rules2011.pdf>
- [4] Newstadt, G., K. Green, D. Anderson, M. Lang, Y. Morton, and J. McCollum, "Miami RedBlade III: A GPS-aided autonomous lawnmower," *J. Global Positioning Systems*, 7, No.2, P115-124, 2008.
- [5] http://www.engineeringtoolbox.com/friction-coefficients-d_778.html
- [6] Casassa, G., Narita, H., & Maeno, N., "Shear cell experiments of snow and ice friction. *Journal of Applied Physics*," 69(6), p3745, 1991.
- [7] Kuroiwa, D., "The kinetic friction on snow and ice," *Journal of Glaciology*, 19(81), p141-152, 1977.
- [8] http://www.kingston.com/ukroot/ssd/v_series.asp
- [9] http://www.pcpower.com/technology/power_usage
- [10] Roboteq, Inc., "AX2550 AX2850 Dial Channel High Power Digital Motor Controller User's Manual". RoboteQ. 2007.
- [11] <http://www.microstrain.com/3dm-gx2.aspx>
- [12] <http://www.microstrain.com/pdf/3DM-GX1%20Hard%20Iron%20Calibration.pdf>
- [13] <http://www.topconpositioning.com/products/gps/geodetic-receivers/integrated/hiper-lite-plus.html>
- [14] <http://www.mstarlabs.com/control/znrule.html>