GroundsBot: Autonomous Golf Course Maintenance

Conceptual Design Review

CMU MRSD Team A
Sponsor: Discovery Robotics
David Evans
Adam Driscoll
Henry Chen
Josh Bennett
Joe Phaneuf

September 2017

GROUNDSBOT

Contents

1 Project Description

People spend a lot of time mowing grass. This is particularly true for golf courses. Golf courses have hundreds of acres of land requiring careful maintenance. The median club in the United States spends \$1.2 million per year on maintenance [1]. A Pittsburgh area golf course superintendent [6] revealed that his staff spends 65% of their time mowing, and 60% of that time is spent just on the rough. This is an industry ripe for automation.

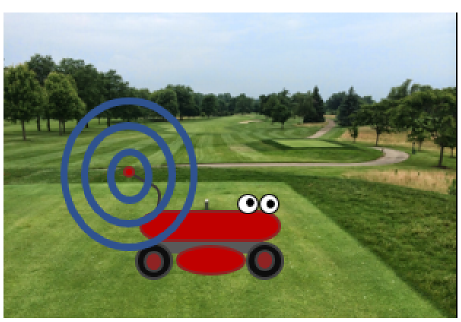
Autonomous mowers are commercially available today, however most of these units require buried wires, beacons, or extensive mapping to provide boundaries. For a golf course, the mowing area is so large that those setup methods are a barrier to deployment. The GroundsBot team aims to deliver an autonomous mower that can be deployed with minimal infrastructure, using a coarse boundary map provided by the user.

2 Use Case

Meet Steve. He's a golf course superintendent responsible for keeping the golf course looking professional. Right now, Steve and his groundskeeping team spend 65% of their time just mowing grass. To save time, Steve just bought a GroundsBot unit to autonomously mow the rough around his golf course.

To start the mowing process, Steve first has to draw an outline of the area he wants mowed. He also highlights any zones Groundsbot should avoid while mowing.

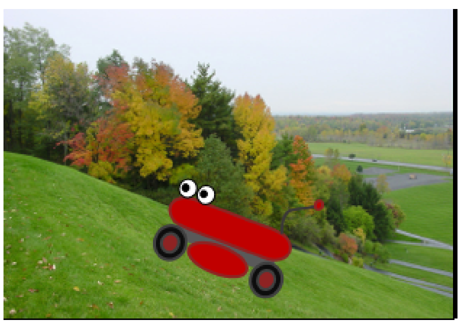
With the area to mow now outlined, the Groundsbot planning software creates a global route to cover all of the requested area. It avoids large obstacles like fairways, forests, and parking lots.



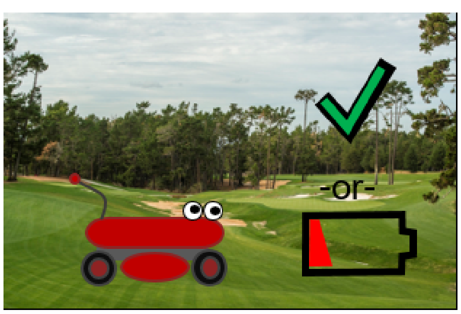
GroundsBot recieves the global route and localizes through GPS to determine where it is on the golf course.

With the unit located and a global plan uploaded, GroundsBot begins mowing the area specified by Steve. It moves back and forth cutting the grass precisely, matching the quality care provided by Steve and his groundskeeping team.

If GroundsBot encounters unplanned obstacles like sand traps or individual trees, GroundsBot avoids the obstacle and cuts around it. Once GroundsBot has avoided the obstacle in its path, it continues with the original plan.



In order to maximize coverage, GroundsBot will navigate difficult terrain on a golf course, cutting the rough on steep hills.

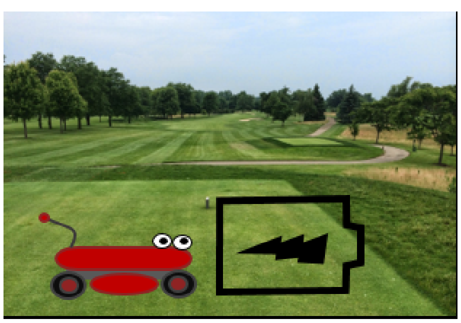


Once GroundsBot is finished or low on power, it notifies Steve and returns back to the charging dock.

GroundsBot collected a lot of data while it was out cutting. It transfers diagnostics, coverage reports, and cutting data to Steve's mobile device.

From here, Steve can analyze what areas of the course GroundsBot missed and what areas he has to send his team out to trim or tidy up.

Steve sends out his groundskeeping team to quickly address missed spots or edging. This is a much faster experience for the team because they only have to touch up a few areas on the course.



With the course looking professional, Steve and his groundskeeping team are happy they don't have to worry about mowing. GroundsBot recharges to prepare for another day of mowing.

3 System Level Requirements

The system requirements were decided upon to demonstrate a proof of concept autonomous lawn mower that can be set up with minimal added infrastructure. The only added infrastructure will be a docking station for GroundsBot. To accomplish this, GroundsBot must be able to localize with extreme accuracy in an unknown environment. GroundsBot must also be able to navigate consistently along a mowing path plan. Finally, GroundsBot must be able to detect and avoid any obstacles along the way.

3.1 Mandatory Requirements

Table 1: Mandatory Performance Requirements

ID	Requirement	Description
M.P.1	First time user inputs map within 15 minutes	GroundsBot should be easy to use
M.P.2	System returns proposed route/coverage map within 5 minutes	Related to M.P.1, GroundsBot should start up quickly to ease the mind of the user
M.P.3	Cut 0-25% overlap for 95% of grass	When mowing, GroundsBot should cut in a way that accomplishes full grass coverage while maintaining efficiency by reducing overlap
M.P.4	Mow $50ft^2$ of 30 degree sloped grass	GroundsBot should be able to handle steep slopes to eliminate safety hazards
M.P.5	Detect 80% of objects greater than 27 cubic inches	GroundsBot should be able to recognize obstacles in order to prevent collisions and mowing accidents
M.P.6	Mow to within 1 foot of detected obstacles	GroundsBot should be able to detect an obstacle (M.P.5) and navigate around it to continue its mowing path
M.P.7	Mow 90% of a $\frac{1}{4}$ acre area	GroundsBot will encounter obstacles along the way preventing it from 100% coverage

3.2 Desirable Performance Requirements

Table 2: Desirable Performance Requirements

ID	Requirement	Description
D.P.1	Mow to within 3 inches of a detected obstacles	A stretch goal for when M.P.6 (mow within 1 ft. of obstacles) is achieved
D.P.2	Visually report mowing coverage and obstacles encountered	GroundsBot should report areas it missed to the user so the user knows where to manually mow to achieve full coverage

3.3 Mandatory Non-Functional Requirements

Table 3: Mandatory Non-Functional Requirements

ID	Requirement	Description
M.N.1	Return home to within 5 feet of dock	GroundsBot should return to its starting position to remove as much hassle from the user as possible
M.N.2	Have a functional and easily accessible emergency stop	GroundsBot should be safe to use and easy to shut down in case of emergency
M.N.3	Be clearly visible	GroundsBot should indicate its presence and status to everyone nearby
M.N.4	Do not tear up grass	GroundsBot should not ruin any area it travels over

3.4 Desirable Non-Functional Requirements

Table 4: Desirable Non-Functional Requirements

ID	Requirement	Description
D.N.1	Operates in variable lighting conditions	GroundsBot should operate at night to avoid interrupting golfers
D.N.2	Deck adjustable 0.5” to 2”	GroundsBot should be able to meet the grass height standards of different golf courses
D.N.3	Bot is resistant to impacts	GroundsBot should be unaffected if hit by a golf ball while mowing
D.N.4	Return home to and mate with charging dock	A stretch goal if M.N.1 (return to within 5 ft. of dock) is achieved

4 Functional Architecture

There are three subsystems that combine to accomplish the functionality needed for GroundsBot (Figure 1). The User Interface accepts inputs from the groundskeeper, generating detailed mowing regions. The Charging Dock acts as a home point for charging the robot, while also providing a reference signal for correcting localization issues. The Mowing Robot includes internal subsystems which enable the robot to navigate the golf course, cut the rough, and return to the charging dock when needed.

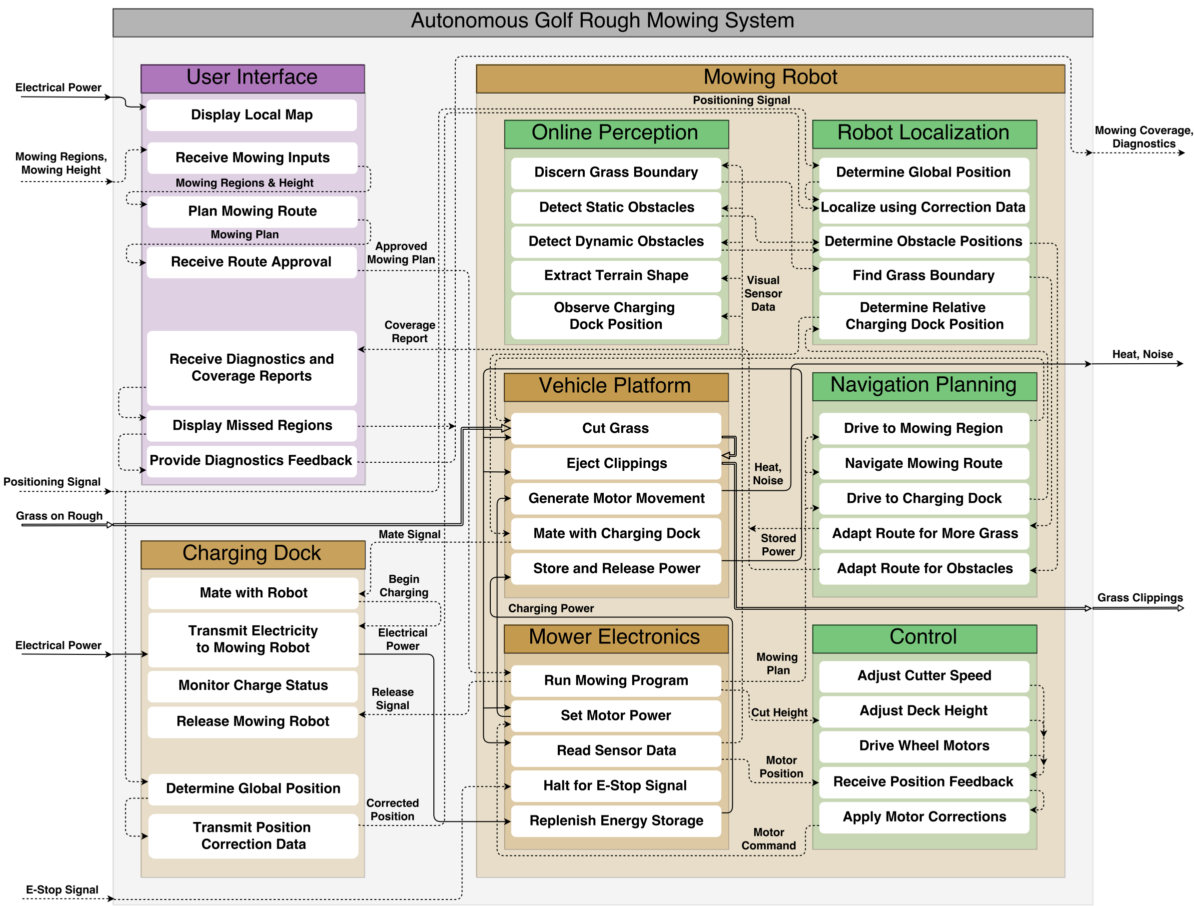


Figure 1: Functional Architecture

The Mowing Robot subsystem is responsible for many critical tasks in the Autonomous Golf Rough Mowing system. The Vehicle Platform is one internal subsystem that ensures effective mowing operations by storing energy, providing mobility, and docking when charging is needed. The Mower Electronics subsystem provides electrical controls for the motors and hosts the computing resources needed to run the algorithms for mowing. The sensor data provided by this subsystem is crucial for the remaining systems, enabling full mowing autonomy.

The remaining subsystems within the Mowing Robot include Online Perception, Robot Localization, Navigation Planning, and Control. These software systems provide GroundsBot with awareness of surroundings, knowledge of position, and in-depth details about the grass. This enables successful navigation through complicated environments, and provides the information necessary to control GroundsBot effectively.

5 System Level Trade Studies

5.1 Platform Configuration Trade Study

The first step was to decide a platform configuration that will meet the system requirements. A handful of common wheeled platforms [2] variants were considered (Table 5). Each was evaluated on speed, wheel compaction, stability, complexity, ease of odometry, turning radius, and performance on uneven terrain. A two wheeled chassis with casters was selected (Table 6).

Table 5: Platform Configuration Diagrams

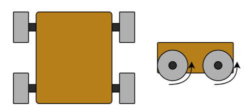
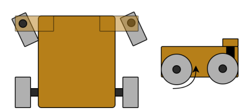
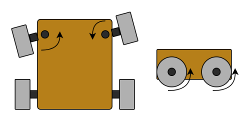
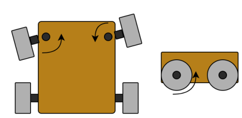
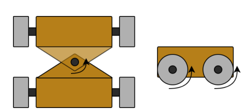
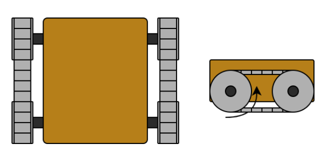
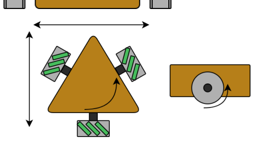
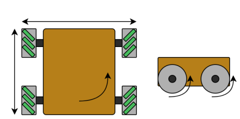
Configuration	Description	Diagram
4 Wheel Skid	Four driven wheels with skid steering.	
2 Wheel Differential with Casters	Two driven wheels with casters for free rotation.	
AWD Standard Steering	Four driven wheels with standard differential steering.	
RWD Standard Steering	Two driven wheels with standard differential steering.	
Articulated	Four driven wheels with center pivot articulated steering.	
Tracked	Two driven wheels with tracks and skid steering.	
3 Wheel Delta	Three driven wheels with immediate x, y, and rotation control.	
4 Wheel Omniwheel	Four driven wheels with immediate x, y, and rotation control.	

Table 6: Platform Configuration Trade Study

	Speed	Wheel Compaction	Stability	Platform Complexity	Odometry Accuracy	Turning Radius	Performance on Uneven Terrain	Score
Weights (1-5)	3	2	1	3	4	5	5	
4 Wheel Skid	5	4	4	4	1	1	5	73
2 Wheel Differential with Casters	4	3	4	5	5	5	5	107
AWD Standard Steering	5	4	4	2	4	2	5	84
RWD Standard Steering	5	4	4	3	5	2	5	91
Articulated	3	4	2	1	3	2	5	69
Tracked	2	5	5	3	1	5	5	84
3 Wheel Delta	1	3	1	1	1	5	1	47
4 Wheel Omniwheel	2	4	2	1	5	5	1	69

Once a base configuration was determined, a specific implementation of the platform needed to be chosen. Several off-the-shelf as well as DIY platforms were considered [3] [5]. A DIY platform was chosen (Table 7).

Table 7: Platform Base Trade Study

	Ease of Integration	Ease of Repair	Ease of Construction	Flexibility for Modifications	Cost	Lead-time	Traction	Score
Weights (1-5)	3	2	3	5	3	2	4	
Complete DIY	4	5	2	5	4	4	5	93
RC Lawnmower	5	3	5	4	1	2	5	83
Modify Robot Lawnmower	3	3	4	1	1	4	2	51
Modify Electric Push Mower	2	2	3	1	2	5	1	44
Modify Electric Ride on Mower	2	1	3	1	3	5	5	61
Stock Platform with Mower Attached	5	3	5	4	1	3	3	77

5.2 Sensor Trade Study

Once the platform was decided upon, analysis on different sensors and sensor combinations began. Sensors were evaluated individually first to establish baseline capability and relevance to the project. Then sensors were evaluated in different combinations, taking into account the results of the individual sensor study. The results showed that a stereo vision camera was the sensor that best fit the needs of GroundsBot (Table 8).

Table 8: Sensor Capabilities

	Identify Static Obstacles	Identify Dynamic Obstacles	Detect Grass / No Grass	Functions Outdoors In Daylight	Detect Grass Transitions	Sense Driveable Land
LiDAR	5	5	0	5	0	5
Camera	3	3	5	5	5	1
RGBD Camera	5	5	5	5	0	5
Omnidirectional Camera (upwards)	3	3	0	5	0	0
Stereo Camera	4	3	5	5	5	3
Thermal Camera	1	5	0	5	0	1
Sonar	3	3	0	5	0	2

Table 9: Perception Sensor Trade Study

	Ability	Cost	Hardware Integration Complexity	Software Integration Complexity	Computational Intensity	Score
Weights	5	3	4	5	3	
LiDAR + 1x Camera	4	2	3	4	3	67
LiDAR + Stereo Camera	5	1	3	3	2	61
Omnidirectional Camera + Down Camera	2	5	4	2	3	60
Stereo Camera Only	3	4	5	4	4	79
Thermal Camera + LiDAR + Camera	4.5	1	3	2	2	53.5
Many Cameras	3	4	2	2	2	51

6 Cyberphysical Architecture

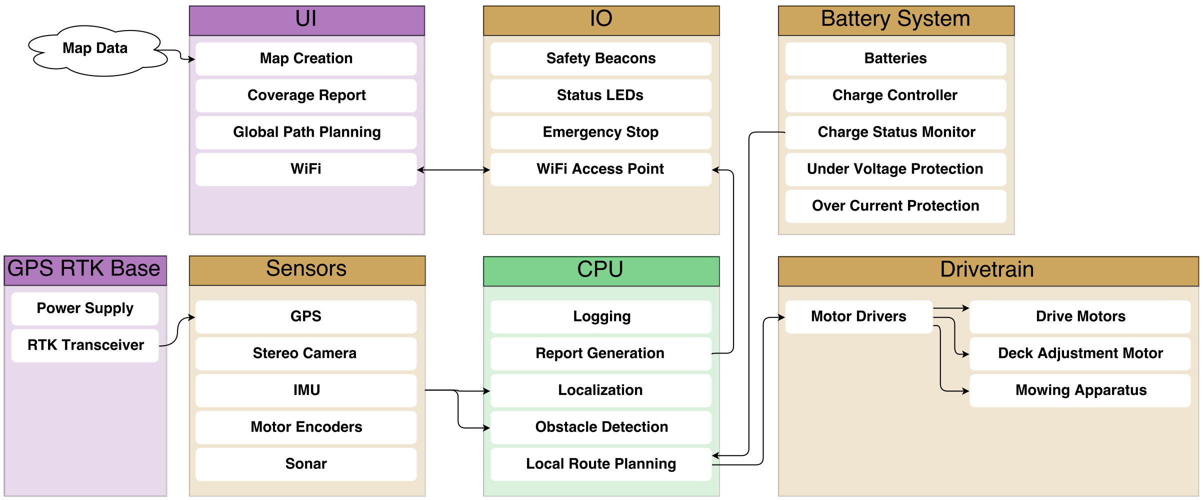


Figure 2: Cyberphysical Architecture

The user interface and IO subsystems are the contact points for the operator. The WiFi access point allows the user to load mowing boundaries from their laptop or mobile device. This also enables the user to download coverage reports and diagnostics. Status LEDs allow the operator to understand the system status at a glance, while safety beacons alert humans to GroundsBot's presence at a distance.

GroundsBot uses its sensor suite to ensure a clean cut. GPS [4], IMU, and Motor Encoders are used for coarse localization while the cameras are used to ensure optimal grass overlap while cutting. Cameras and also enable GroundsBot to detect static and dynamic obstacles. A GPS RTK base station allows for tighter localization.

GroundsBot's battery subsystem reports charge status to the CPU to prevent GroundsBot from becoming stranded. The battery subsystem also contains standard protection protocols including charge control, under-voltage lockout, and over-current protection.

The CPU utilizes sensor information for localization, obstacle detection, and planning. After a local path is devised drive signals are sent to the motors. The CPU also logs coverage data and generates reports.

The Drive Train contains motors and motor drivers to propel GroundsBot across a golf course. This also encompasses the mowing apparatus and mowing deck height control.

7 Subsystem Descriptions

7.1 UI

It is critical for the user to communicate a desired mowing path to the robot. This input method must be separate from the robot itself, allowing the user to be somewhere else as the robot mows autonomously.

This will be achieved through a mobile device, either a dedicated tablet or the user's own smartphone. An app or website will be loaded onto this device and allow for the user to input a map outline that will be communicated to the robot.

7.2 Hardware

7.2.1 Platform

Based on the trade study, a differential drive system with two drive wheels and caster wheels will be used. Another trade study was used to narrow down the platform to a DIY platform instead of a preexisting platform. This custom platform will utilize aluminum extrusions, allowing for easy re-sizing of the platform as well as simple adjustment of the sensor mounts. Currently, motors and batteries from Discovery Robotics will be integrated, but alternate products may be also be considered.

7.2.2 Mower

In addition to navigating a plot of grass, the system also needs to mow it. Given that a custom platform will be made, this will be done by attaching an existing electric push mower to the robot platform. Alternatively, it will also be possible to attach the blades from a manual reel mower to the platform.

7.3 Software

7.3.1 Planner

The planner subsystem consists of two parts: a global planner and a local planner. The global planner will take the user input from the UI and translate the input into a path that the robot will follow. This involves refining the outline that was provided by the user and then finding a path that will allow for maximum coverage of the selected area.

The local planner will be responsible for adapting the route on-the-fly to avoid any obstacles that the robot may encounter. This system will also drive to and from the dock and cutting area.

7.3.2 Localization and Perception

An accurate description of the robot global position and orientation will be obtained from this subsystem. This information may be obtained from a high-accuracy GPS system, such as RTK-GPS, but visual SLAM based methods will also be investigated.

The vision system will also be used to detect boundaries of the grass as the robot approaches the boundary. For obstacle detection, our trade study determined that a stereo camera system would best fit the detection requirements. However, if needed, a LiDAR may be added to improve the detection accuracy.

7.3.3 Mobility

The mobility subsystem will be responsible for controlling the robot motors. Part of this involves the robot going from point A to point B at any given time. Using current position information from the localization subsystem, this subsystem will find the next waypoint that will allow for the robot to follow the predetermined plan and then transmit the necessary motor control signals to move towards this waypoint.

The cutter motor speeds will also be controlled by this subsystem. This subsystem must both detect the speed of the motor, as well as apply the correct motor signals to maintain a constant speed when the cutter encounters an obstacle.

8 Project Management

8.1 Work Plan

8.1.1 Tasks

The GroundsBot work plan is shown in the list below. The plan is separated into high level categories and underlying subtasks.

Chassis

- Complete BOM
- Design Power Distribution and Sensor Integration PCBs
- Design Chassis
- Design GPS RTK Base Station
- Acquire Parts
- Integrate Subsystems

Simulation

- Design Simulation Environment
- Complete Cursory Simulations

Localization

- Integrate GPS + RTK Localization

Perception

- Develop Static Obstacle Detection
- Develop Dynamic Obstacle Detection

Planning

- Develop Global Route Planner
- Develop Obstacle Rerouting

Control

- Develop Motor Control

UI

- Create Web Application for Mapping

8.1.2 Schedule

The full list of milestones is laid out below in Table 10

Table 10: Project Milestones

Deadline	Milestone
Progress Review 1 [OCT 17 2017]	Mechanical CAD Complete Electrical CAD Complete BOM Complete Parts Ordered ROS Environment Initialized
Progress Review 2 [OCT 26 2017]	Chassis Assembled with Mowing Apparatus ROS Simulation Environment Complete GPS RTK Base Station Complete
Progress Review 3 [NOV 7 2017]	GroundsBot System Integration Complete GPS RTK Integration Complete Control Systems Demo Complete
Progress Review 4 [NOV 21 2017]	GroundsBot Accepts GPS Waypoints GroundsBot Follows GPS Waypoints Teleoperation Test Complete
Fall Validation Experiment [NOV 28 2017]	GroundsBot Follows Route From Web App GroundsBot Differentiates Grass From Other Objects
January 2018	Global Planning Algorithm Complete Static and Dynamic Obstacle Detection Complete
February 2018	Read User Map Input GroundsBot Reroutes Around Obstacles
March 2018	Full Mapping UI Complete GroundsBot Autonomously Cuts Lot
April 2018	Final System Tests Complete

8.1.3 Progress Reviews

Progress Review 1: The goal for Progress Review 1 is to have all hardware design complete with parts on order.
Progress Review 2: The goal for Progress Review 2 is to have an assembled frame/chassis, a full ROS simulation environment, and a completed GPS + RTK base station.

8.2 System Validation Experiments

8.2.1 Fall Validation Experiment

The aim of the fall validation experiment is to test individual subsystems. As such many of the systems requirements set for GroundsBot will not be fully met. All tests to be performed have been designed to indicate significant progress towards reaching the system requirements. More specifically the team plans to test the base functionality of the mobility, localization, planning, and perception subsystems of GroundsBot. The details of the test are laid out below.

Test 1:

Location: Field by Doherty Apartments [7]
Equipment: GroundsBot, GroundsBot dock/RTK base station, laptop

1. Power on GroundsBot next to its docking station
2. Establish connection between GroundsBot and mobile device
3. Input GPS waypoints following a typical zigzag pattern a groundskeeper might make when mowing a lawn
4. Send waypoints to GroundsBot
5. GroundsBot will navigate to each waypoint entered, in the order they were entered
6. Once the last waypoint is reached, GroundsBot will navigate back to the docking station (Note: the docking station will not be one of the entered waypoints)

Test 2:

Test 2 has been designed to demonstrate base functionality of the perception subsystem. The team will present a perception algorithm capable of differentiating between grass (i.e. a mowable surface) and non-grass (i.e. a non-mowable surface.) This test will be performed outside of the fall validation experiment and a replay of the test will be displayed during the fall validation experiment.

8.2.2 Spring Validation Experiment

The spring validation experiment will be when the team will test the full GroundsBot system to demonstrate that all system requirements have been met. The details of the test are laid out below.

Test 1:

Location: Field by Doherty Apartments
Equipment: GroundsBot, GroundsBot dock/RTK base station, mobile device

1. Power on GroundsBot next to its docking station
2. Open UI on mobile interface and establish a connection with GroundsBot
3. Have a new user (someone not on Team A) use the UI to draw an outline of the area to be mowed on a map including both static obstacles and a non-mowable surface (i.e. concrete)
4. Use the UI to submit the mowing area to GroundsBot
5. GroundsBot will develop a coverage plan of the area input by the user
6. GroundsBot will navigate to the area to be mowed
7. Once it has reached the edge of the mowing area it will begin mowing
8. GroundsBot will detect and avoid any obstacles it comes across while mowing and will also only mow where there is grass
9. Once GroundsBot has mowed the whole area it will return to the docking station
10. GroundsBot will generate and transmit a coverage report to the UI indicating areas it could not mow

8.3 Team Member Responsibilities

Joe Phaneuf

Team Lead

Systems Engineer

Project Management

Henry Chen

Mechanical Design

Prototyping

Computer Vision

Josh Bennett

Electrical Design

Mechanical Design

Instrumentation and Sensors

Adam Driscoll

Control Systems

Perception

David Evans

Power Distribution

Motion Planning

Localization and Mapping

8.4 Provisional BOM

A bill of materials is included in Table 11 with potential components and rough pricing information. Batteries and motors will be loaned to us from Discovery Robotics, reducing overall project cost. The batteries and motors shown in the BOM are backup components to be used if the loaned components do not fulfill project requirements.

Table 11: Provisional BOM

Manufacturer	Part No.	QTY	Cost	Description
SuperDroid Robots	TD-111-135	1	$900	Wheelchair Motors & Encoders
SuperDroid Robots	TE-240-030	1	$330	Motor Controller
SuperDroid Robots	TD-178-000	1	$150	13" Tiller Tires and Mounting
Caster Connection	S-5210-PRB	1	$70	10" Pneumatic Swivel Caster
Smart Battery	SB2425	1	$700	24V 25aH Battery
Smart Battery	DP-RS2	1	$165	Battery Charger
NVIDIA	Jetson TX2 Dev Kit	1	$560	Embedded GPU
ITEM	Extrusion	1	$400	Extrusions and Joints
McMaster-Carr	Mechanical Components	1	$500	Misc Fasteners, Bearings
DigiKey	Electrical Components	1	$600	Misc Connectors, Components
Total:			$4,375	

8.5 Risk Management

Risk	Management Strategy	Risk Category
Too Few Features for Perception: If we cannot consistently detect grass from non-grass features, perception subsystem will not be able to direct platform accurately	Install infrastructure such as fiducials to assist system in recognizing objects.	Technical
Team Falling Behind Due to Work: Team may overestimate capability and overwhelm itself with assignments.	Prepare for risk by managing tasks and using buffer weeks. Mitigate effect by delaying milestones or cutting compromising targets.	Schedule
Localization Poor for Following Edges: If our localization algorithms cannot accurately follow edge of lawn, system will not mow lawn effectively.	Prepare by understanding criticality of localization. Mitigate effect by changing scope of mowing problem or adding infrastructure to assist.	Technical
Injury from Spinning Blade: Our system has a dangerous cutting instrument. It can injure a team member or a bystander.	Prepare by making sure team understands the risk of spinning blade. Put warning lights and sounds to alert bystanders when testing.	Technical
Poor Weather for Validation Experiments: Poor weather on demonstration days would prevent us from system from performing.	Team monitors upcoming weather before validation experiments. Records the validation experiment before hand and replays in the case of inclement weather conditions.	Schedule
Team Member Has Family Constraints: Any team member could experience a family emergency or responsibility that occupies their time and focus. One specific concern is Josh's wife is pregnant and expecting in December.	Mitigate the effect on entire team by spreading work across several team members and communicating consistently with absent team member to keep them in the loop.	Personal
Platform Devastating Event: Platform could have electrical, thermal, or kinetic event that leads to its destruction.	Team designs platform with risk in mind, and manages parts reserve to quickly rebuild platform if necessary.	Technical
Subsystem Harder Than Expected: Any subsystem could require more work than expected, taking away focus from other subsystems.	Team can prepare by focusing on critical subsystems first, and may revise scope related to subsystem if necessary.	Technical
Not Enough Capital for Development: There is a risk our system requires more money than initially allocated.	To mitigate the effect of this event, team will maintain strong relationship with sponsor by providing consistent communication of development progress. Team may also reach out to other sponsors or donors.	Schedule

9 References

[1] http://www.clubbenchmarking.com/blog/golf-course-maintenance-how-much-should-you-spend

[2] https://www.clearpathrobotics.com/husky-unmanned-ground-vehicle-robot/

[3] http://www.husqvarna.com/us/products/robotic-lawn-mowers/

[4] https://www.swiftnav.com/store?category=GNSS+Modules

[5] https://www.superdroidrobots.com/shop/item.aspx/prebuilt-2wd-66in-lawn-mower-wc-sold/1983/

[6] Duxbury, Jeff [Bob O'Connor Golf Course Superintendent]. (2017, September 16). Personal interview.

[7] Guenther, Steve [Carnegie Mellon University Director of Facilities Operations]. (2017, September 22). Personal interview.