

GroundsBot:Autonomous Golf Course Maintenance

Team A
David Evans
Adam Driscoll
Henry Chen
Josh Bennett
Joe Phaneuf

September 2017

Contents

1	Project Description	3
1.1	Functional Requirements	3
2	Use Case	3
3	System Level Requirements	3
3.1	Mandatory Requirements	3
3.2	Desirable Performance Requirements	4
3.3	Mandatory Non-Functional Requirements	4
3.4	Desirable Non-Functional Requirements	4
4	Functional Architecture	5
5	System Level Trade Studies	6
5.1	Platform Trade Study	6
6	Cyberphysical Architecture	8
7	Subsystem Descriptions	9
7.1	UI	9
7.2	Hardware	9
7.2.1	Platform	9
7.2.2	Mower	9
7.3	Software	9
7.3.1	Planner	9
7.3.2	Localization and Perception	9
7.3.3	Mobility	9
8	Project Management	10
8.1	Work Plan	10
8.1.1	Tasks	10
8.1.2	Schedule	10
8.1.3	Progress Reviews	11
8.2	System Validation Experiments	11
8.2.1	Fall Validation Experiment	11
8.2.2	Spring Validation Experiment	12
8.3	Team Member Responsibilities	12
8.4	Provisional BOM	12
8.5	Risk Management	12
9	References	12

1 Project Description

GroundsBot

Humans 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 [reference]. A Pittsburgh area golf course manager 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 a prototype autonomous mower that can be deployed with a simple rough user sketch on a map.

1.1 Functional Requirements

Groundsbot shall: Receive mowing regions
Cut grass at the correct height
Create a mowing plan
Cut the rough
Avoid obstacles
Provide feedback to the groundskeeper

2 Use Case

3 System Level Requirements

3.1 Mandatory Requirements

Table 1: Mandatory Performance Requirements

ID	Requirement	Description
M.P.1	First time user inputs map within 15 minutes	GroundsBot shall 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	GroundsBot should be able to efficiently and effectively mow the grass
M.P.4	Mow $50ft^2$ of 30 degree sloped grass	GroundsBot should be able to handle these slopes to eliminate the safety hazard
M.P.5	Detect 80% of objects greater than 27 cubic inches	GroundsBot should not crash into anything
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	This error should account for obstacle avoidance, and any unforeseen circumstances such as localization issues

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 is achieved
D.P.2	Return home to within 5 feet of start	Groundsbot should return to its starting position to remove as much hassle from the user as possible
D.P.3	Return home to and mate with charging dock	A stretch goal if D.P.2 is achieved
D.P.4	Visually report mowing coverage and obstacles encountered	GroundsBot should report areas it missed to the user so they can go and manually cover those areas

3.3 Mandatory Non-Functional Requirements

Table 3: Mandatory Non-Functional Requirements

ID	Requirement	Description
M.N.1	Bot has emergency stop	Ensure Groundsbot is safe to use
M.N.2	Bot is clearly visible/noticeable	Nearby pedestrians need to be alerted that there is an autonomous robot nearby. Users need an easy way to tell the status of the robot
M.N.3	Bot does not destroy grass	

3.4 Desirable Non-Functional Requirements

Table 4: Desirable Non-Functional Requirements

ID	Requirement	Description
D.N.1	Operates in variable lighting conditions	Operating at night would be optimal to avoid interrupting golfers as much as possible
D.N.2	Deck adjustable 0.5" to 2"	Rough height at golf courses can vary
D.N.3	Survives deluge of golf balls	Groundsbot is quiet enough to run during golfing hours it is likely to get hit with golfballs

4 Functional Architecture

There are three sub-systems 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 sub-systems which enable the robot to navigate the golf course, cut the rough, and return to the charging dock when needed.

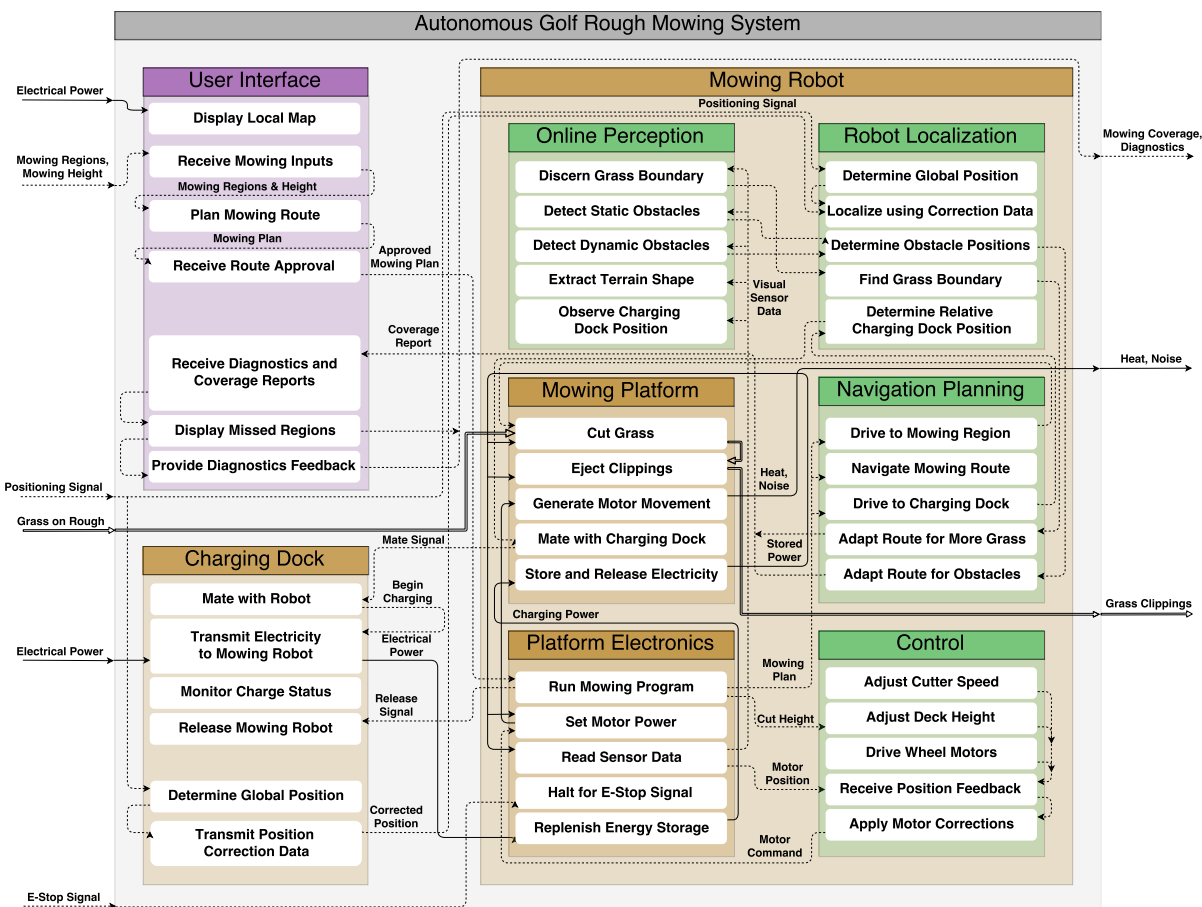


Figure 1: Functional Architecture

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

The remaining sub-systems 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 Trade Study

Table 5: 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

Table 6: 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 Pushmower	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

Table 7: 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	5	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 8: Perception Sensor Trade Study

			Hardware Integration Complexity	Software Integration Complexity	Computational Intensity	Score
Weights	Ability	Cost	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

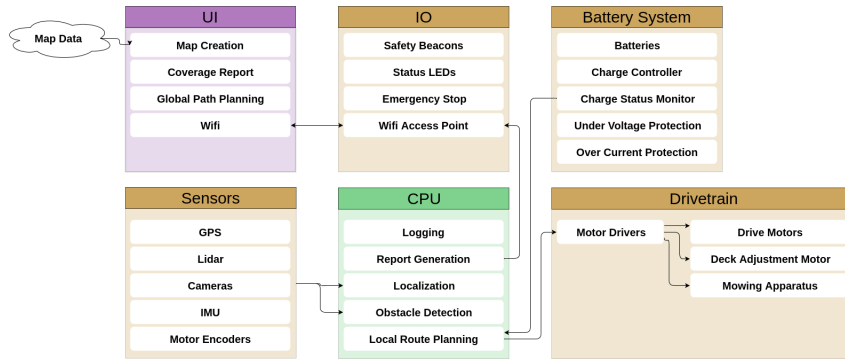


Figure 2: Cyberphysical Architecture

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, IMU, and Motor Encoders are used for coarse localization while the cameras are used to ensure optimal grass overlap while cutting. Cameras and lidar also enable GroundsBot to detect static and dynamic obstacles.

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

An important aspect of the project is a way for the user to communicate his 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 one or more caster wheels will be used. Another trade study was used to narrow down the platform to DIY platform instead of a preexisting platform. This custom platform will utilize aluminum extrusions, allowing for easy resizing of the platform as well as easy 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 is also 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 that 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 maximal 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 take care of the driving to and from the dock to the actual cutting area.

7.3.2 Localization and Perception

An accurate description of the robot's 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 gets closer to the boundary. For obstacle detection, a 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 the control of the robot's motors. Part of this involves the robot going from point A to point B at any given time. Using current position information from localization subsystem, this subsystem will find the next waypoint that will allow for the robot to follow the predetermined plan and then emit 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.

1. Chassis
 - (a) Complete BOM
 - (b) Design Power Distribution and Sensor Integration PCBAs
 - (c) Design Chassis
 - (d) Design GPS RTK Base Station
 - (e) Acquire Parts
 - (f) Integrate Subsystems
2. Simulation
 - (a) Design Simulation Environment
 - (b) Complete Cursory Simulations
3. Localization
 - (a) Integrate GPS + RTK Localization
4. Perception
 - (a) Develop Static Obstacle Detection
 - (b) Develop Dynamic Obstacle Detection
5. Planning
 - (a) Develop Global Route Planner
 - (b) Develop Obstacle Rerouting
6. Control
 - (a) Develop Motor Control
7. UI
 - (a) Create Web Application for Mapping

8.1.2 Schedule

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

The full list of milestones is laid out below in Table 9

Table 9: 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 Milestone	Global Planning Algorithm Complete Static and Dynamic Obstacle Detection Complete
February 2018 Milestone	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

8.2 System Validation Experiments

8.2.1 Fall Validation Experiment

Fall Validation Location: Field by Doherty Apartments

8.2.2 Spring Validation Experiment

8.3 Team Member Responsibilities

8.4 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

9 References