Autonomous Golf Green Divot Locator Robot Project Proposal

Group 19 Team Members:
Ved Eti (vedeti2)
Michael Cherry (mcherry3)
Akhil Bonela (abonela2)

TA: Pusong LI
September 19, 2024
ECE 445

1. Introduction:

Problem

Preserving the quality of golf greens is essential to ensuring a fair and enjoyable golfing experience. However, a common challenge that undermines this objective is the failure to repair ball marks. When a golfer's ball lands on the green, it creates a small indentation, or divot, in the surface. While it is customary for players to use a repair tool to fix these marks, not all golfers adhere to this etiquette. This neglect leads to an increase in divots and uneven patches on the green, which can have detrimental consequences for both the course and the golfers.

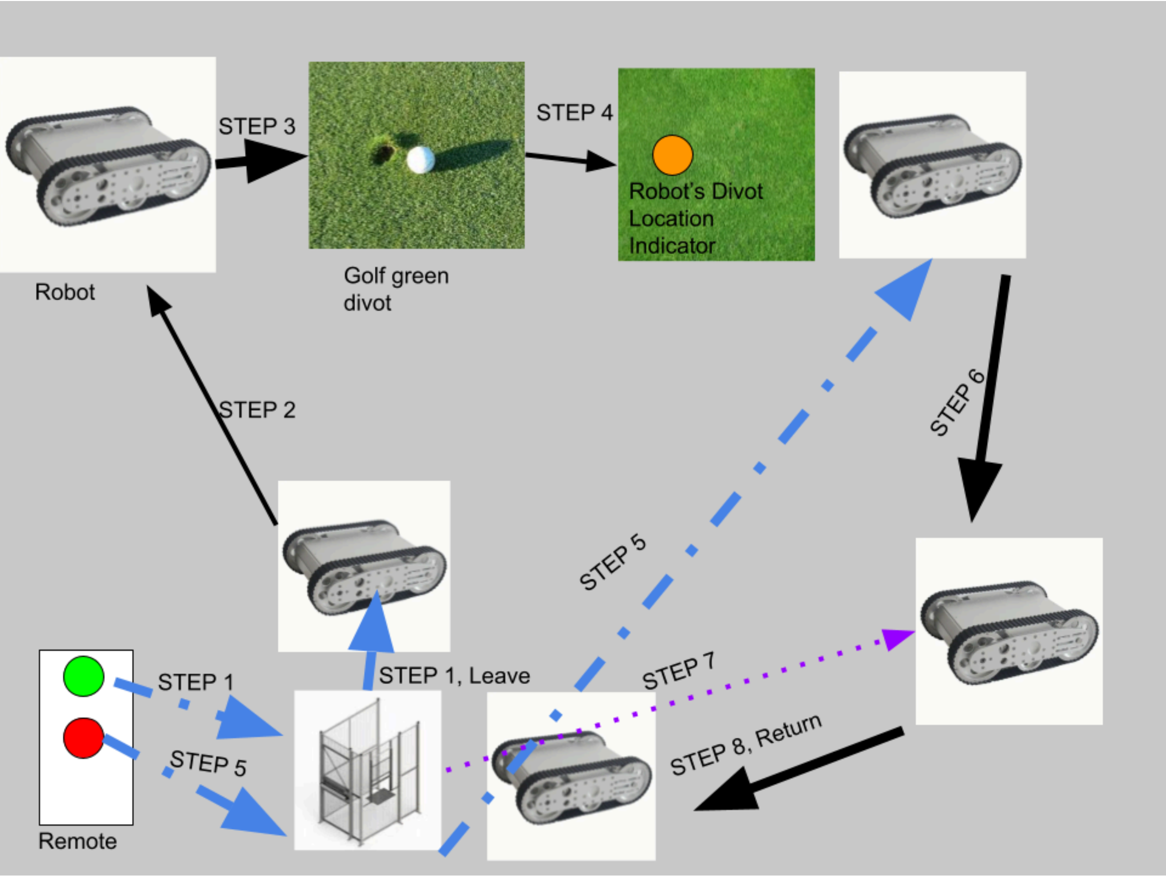
Unrepaired ball marks can significantly diminish the quality of a golf green. The divots disrupt the smooth flow of putts, making it more difficult for golfers to accurately judge distances and control their shots. Additionally, these marks can interfere with the green's drainage system, leading to localized water pooling and potential turf damage. Furthermore, the unsightly appearance of a green littered with divots can diminish the overall aesthetic appeal of the course, negatively impacting the golfing experience for all players.

Solution

Our proposed solution involves developing an autonomous robot equipped with advanced sensing and marking capabilities. This robot will be designed to traverse the golf green at the end of the day, when the golf club closes. Using stereo cameras, the robot will accurately locate divots by analyzing the differences in depth between the surrounding turf and the indented areas.

Once divots are identified, the robot will use a custom-designed marking tool to clearly indicate their locations. This tool will leave a visible mark on the green, guiding golfers to repair the divots before their next shot. By automating the process of divot identification and marking, our robot will significantly reduce the manual effort required to maintain the quality of golf greens while ensuring that all divots are promptly addressed.

Visual Aid



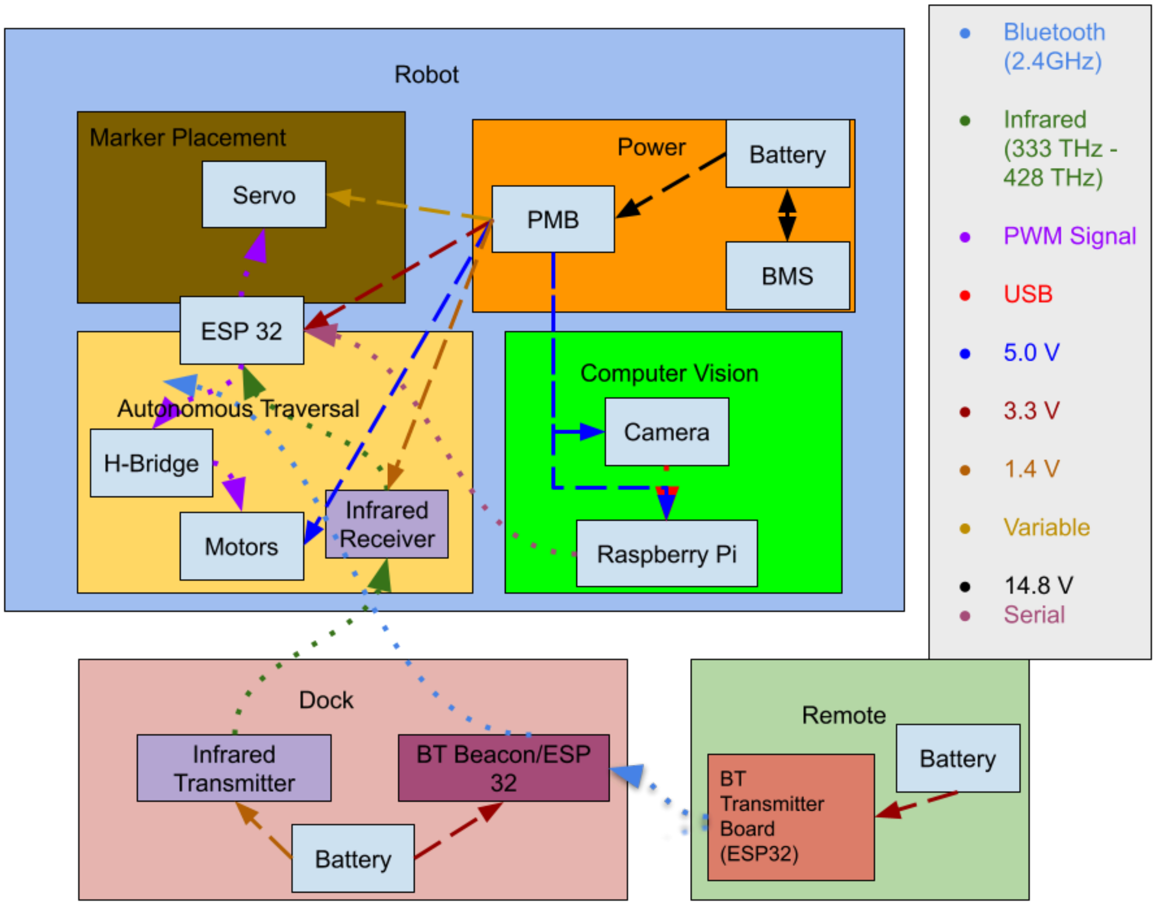
Blue arrows indicate Bluetooth Signals, Purple indicates infrared signals, black is overall traversal of the robot

High-Level Requirements

1. Be able to create a robot that is compact and light enough to traverse the golf course green without damaging the golf green
    a. Want size to be with in 18”x18”x18” and have it weigh under 15lbs
2. Have the robot be able to effectively indicate to the user, either visually or programmatically, where it “thinks” a hole/divot in the golf green is, and place a marker, accurate to within +/- 3 inches
    a. The diameter and depth we hope to successfully locate and indicate is about 1-1.5 inches in radius and 0.5-0.75 inches in depth
3. Have the dock/cage successfully transmit signal to the robot to return on press of user’s remote, and be accurate to within +/- 6 inches of the center of the cage/dock

2. Design

Block Diagram:



Subsystem Overview

Autonomous Traversal

This subsystem controls the robot's movement as it traverses the golf green. This subsystem is mostly going to be interfacing with our microcontroller (ESP 32), our motors, an H-Bridge chip, and the Raspberry Pi. All of the computer vision tasks will be performed on the Raspberry Pi, as our microcontroller will not have enough computational power. The output of the computer vision to either detect the edges of the golf green or a divot will then be sent to the ESP32 microcontroller through general GPIO pins. The microcontroller will then give out instructions for the robot if it detects a golf green, interacting with an H-Bridge motor driver, allowing us to turn around and change the direction of the robot, and continue sweeping the area. This will act very similar to a common Roomba, and other robotic vacuum cleaners. We repeat this process until we traverse and check the entire green.

We plan on using a pre-produced chassis listed above in our component list so that we don't have to spend time making and manufacturing our own chassis with motors. We will add our own microcontroller, PCB for power distribution, and battery to the chassis, and use the chassis mostly for the mechanical build.

The main bridge between this subsystem and the rest of the subsystems is the ESP32 microcontroller and the Power distribution board. The ESP32 microcontroller will receive signals from the Raspberry Pi that detail signals of finding either a divot or detecting the edge of the golf green. We plan to use two general-purpose input-output (GPIO) pins from the Pi and connect it to the microcontroller. These signals will be a simple binary signal (On/Off) to specify if a divot is found or if the edge of the grass is detected. The microcontroller will be programmed to send respective signals to the H-bridge motor controller; either stop if a divot is detected or turn the robot around if the edge of the green is detected. Please see the computer vision subsystem module description for more information on how the raspberry pi and sensors will be used as detection devices. This subsystem is vital to completing the high-level requirements, including the return to the dock capability, traversing the green without causing damage, and stopping to place a marker at the location of the divot. Removing any of the components (Microcontroller, motors, H-Bridge) will cause failure of the system, and will not meet our high-level requirements. Removal of motors will cause our robot to be stationary, not allowing us to traverse the green and mark the divots. Removal of the H-bridge motor driver will not allow us to control the motors effectively using the microcontroller and will require using either a Raspberry Pi or multiple microcontrollers. The microcontroller is extremely important, as it is the component that takes the detection signals and tells the motors and h-bridge how to maneuver the robot. The Raspberry Pi can also be used in place of the microcontroller and would have been our preferred choice but was not allowed for this project.

The last main part of the autonomous traversal is an infrared receiver. This fulfills the last high-level requirement, being able to return to the dock with a level of accuracy. This part is connected to the microcontroller and will direct the robot to move towards the dock. More information on how exactly this will work is given in the dock / remote control subsystem module. Please see power subsystem requirements for details on specific power requirements of components in the autonomous traversal and control subsystem.

Image Processing

.The image processing module will mostly have two tasks, identify divots, and identify edges of golf greens. The two components of the computer vision/image processing submodule are the Raspberry Pi and a stereo camera. The Raspberry Pi will get live image feed from our camera and will perform some computer vision algorithms for detection and classification. It will pass along information about what it detects to the ESP 32 microcontroller so that we can either use the traversal module to move the robot, or the marker placement module to place markers down. For the divot sensing, we plan on using the stereo camera as a method to estimate the changes in depth. As per our research, we plan on using semi-global block matching for our depth perception, which seems like can tell the difference between uneven surfaces on the ground. As another option, we will also take a look at using a yolo v8 model in order to increase the speed at which we create our inferences. We will also use the camera feed and algorithms for the green detection to make sure the robot does not go outside the green. For this we plan on using some thresholding and segmentation to differentiate between the current surface it is on and the other surface it sees. We plan on using various Python libraries such as PyTorch and OpenCV, which are both too computationally intensive for a microcontroller.

As mentioned before, the two components of image processing are the stereo camera and a Raspberry Pi. The stereo camera that we plan on using will be connected to the Raspberry Pi through a USB port. Depending on whether the algorithm detects a divot or if the robot is veering off of the green the Pi will have two GPIO pins that will connect directly to the microcontroller via a simple binary on/off signal. Please see power subsystem for extended details on the power requirements of computer vision subsystem.

Power

The robot utilizes a Zeee 14.8V 4S Lipo Battery (50C 3300mAh) with an XT60 plug, paired with a 14.8V 4S 30A 18650 Lithium Battery BMS PCB Integrated Circuits Protection Board. This combination provides reliable power management and safety features for the robot. LiPo batteries are chosen for their high energy density, which allows for a compact and lightweight battery pack, ideal for mobile robots. The BMS safeguards against overcharging, over-discharging, short circuits, and overcurrent, ensuring the battery's longevity and the robot's safe operation. There will also be a custom PCB Power Management Board (PMB). The PMB is responsible for distributing regulated power to all sub-systems. The dock and remote will both use a 3.7 Volt 3000mAh Rechargeable Lithium-ion (Li-ion) Battery as a power source.

Marker Placement

The purpose of the marker placement subsystem is to facilitate the communication between our vertical marker dispenser and the robot's esp32 microcontroller. This is the subsystem that will make it possible for our robot to indicate where it detected a divot and leave a visible marker on the golf green we can use to directly compare its predictions with actual divots. When the computer vision subsystem detects a divot using its segmentation algorithm, the ESP32 will receive a HIGH signal from the Raspberry Pi once the robot is moved into position. Upon receiving the Pi's signal, the ESP32 will send a serial Pulse Width Modulation (PWM) signal from one of its General Purpose Input/Output (GPIO) pins to indicate to the robot that it is time to dispense a marker from the vertical dispenser. The servo will be rotating a small fin that will move out from under the vertical dispenser containing our markers. This will make a marker fall down, and upon returning to under the vertical dispenser, it will push the marker down a second vertical tube to drop the marker in a static position relative to the body of the robot and ensure a flat landing of the marker.

Remote Begin & Return to Location / Dock

The role of the remote control and dock subsystem is to allow users to remotely communicate with the robot and to give users the power to send the robot out onto the green or back to its home location on their command. This is the driving subsystem for the robot to be able to accurately return to the center of the cage, within an error of 6 inches. The remote will contain its own ESP32 that will function as a Bluetooth client, transmitting 2.4GHz signals to the dock to indicate the user wants the robot to begin traversal. The dock itself will also contain its own ESP32 that will function as both a Bluetooth transmitter and receiver. The dock's ESP32 will receive a Bluetooth signal from the remote and proceed to turn send a Bluetooth signal to the robot to instruct it to run the code to begin traversing the green and detecting divots. We made the dock have its own ESP32 subsystem because it will also include an infrared transmitter to interface with the robot. We have used both Bluetooth and infrared signals for the dock and robot communication to make use of the reliable distance of Bluetooth and the accuracy of infrared to best meet our high-level requirement for dock returning accuracy.

Subsystem Requirements

Autonomous Traversal:

Requirements:

1. Limit a top speed of 0.61 meters per second to keep in line with tolerance analysis calculation.
2. Stop within 3 inches of where a signal to stop is sent to the microcontroller.
3. Be able to complete a 180-degree turn when asked to do so.

Image Processing:

Requirements:

1. Be able to identify divots with 90 percent accuracy.
2. Create a green edge detection algorithm with a run time of less than 150 ms per inference.

3. Create a divot detection algorithm with a run time of less than 75 ms per inference.

Marker Placement

Requirements:

1. When the robot detects a divot and moves into position, the robot dispenses a location indicator marker within 10 second
2. When the servo dispenses a marker, the marker lands flat and within an inch of directly under the dispenser

Remote Begin & Return to Location / Dock

Requirements:

1. The remote can successfully command the robot to begin traversal within 10 seconds of user press
2. The remote can successfully command the robot to begin returning to dock within 30 seconds
3. When the user wants the robot wants to stop traversing, the press of the emergency stop button on the remote makes the robot stationary, within 5 seconds

Power:

1. Minimum Current Output: The power subsystem must continuously supply a minimum current that meets the combined demands of all the robot's components.
    - Minimum Current = Σ (Typical Operating Current) + 20% buffer
    - Minimum Current = (250 + 80 + 500 + 100 + 4*250 + 30) mA * 1.2 = 2352 mA
2. Regulated Voltages: The power subsystem must provide well-regulated voltages to each component according to their specifications.
3. The PMB will need to include voltage regulators to convert the battery voltage (14.8V nominal) to the required voltages for each component:
    - 5V regulated output for Stereo Camera, Raspberry Pi, and the 4 TT DC Motors
    - 3.3V regulated output for the Micro-Controller
4. Runtime Analysis:
    - Idle State:
        1. Robot: Microcontroller (idle), Ultrasonic Sensor, Servo Motor (holding)
        2. Total Power: ≈ 325 mA
    - Low Activity State:
        1. Robot: Microcontroller (active), Ultrasonic Sensor, Servo Motor (holding), TT DC Motors (low speed)
        2. Total Power: ≈ 1200 mA

- High Activity State:
    1. Robot: Microcontroller (active), Ultrasonic Sensor, Servo Motor (active), TT DC Motors (high speed), Raspberry Pi (active)
    2. Total Power: ≈ 4200 mA
- Docking/Charging State:
    1. Robot: Idle
    2. Dock: Microcontroller (active), Docking Mechanism
    3. Total Power: ≈ 250 mA
- Remote Control State:
    1. Remote: Microcontroller (active), Wireless Communication Module
    2. Total Power: ≈ 230 mA
- Weighted Average: Baised on an assumption for how long the robot will last in each state
    1. Low Activity: 60%
    2. High Activity: 20%
    3. Docking/Charging: 10%
    4. Remote Control: 10%
- Average Power Consumption ≈ 1500 mA
- Runtime = Battery Capacity / Average Current Draw = 3.3 Ah / 1.5 A ≈ 2.2 hours

Components	**Typical Operating Voltage (V)**	**Typical Operating Current (mA)**	**Required Voltage (V)**
Stereo Camera	5.0	250	5.0
Micro-Controller (ESP32-S3)	3.3	80 (active)/20 (idle)	3.3
Raspberry Pi (Model 4)	5.0	3000 (peak)/500 (idle)	5.0
Servo Motor	Variable	500 (peak)/100 (holding)	Variable
TT DC Motor	5.0	250	5.0
Infrared Emitter and Receiver Diode	1.4	30	1.4

Tolerance Analysis

Through discussions with our TA Pusong, we identified that the main subsystem that will affect the success of the project is the vision subsystem, specifically with the speed, complexity, and accuracy of the computer vision algorithms. The efficiency of the algorithm to process camera feed and classify divots will then affect how close a marker is placed to the actual divot. While it is hard to give a complete estimate on the speed of our algorithm on the Raspberry Pi, through prior coursework and projects working with the Pi, a reasonable inference time will be about 100 ms. Using a 25 percent buffer, we can assume that the maximum amount of time that will be used is 125 ms, this includes any other time delay to get camera feed to the vision algorithm, and raising signal flags to the microcontroller. Using the example inference time of 125 ms, we can now calculate the speed at which the robot should travel so that we can meet our high-level requirement of placing the marker within 3 inches of the divot. The speed formula that we use is given as following:

$$S = D / T$$

$$S = 3 \text{ inches} / 125 \text{ ms} = 0.024 \text{ inches/ms}$$

The speed at which we can travel to be reasonably safe with our marker placement is 3/125 inches/ms otherwise 0.024 inches per millisecond.

$$S = 0.024 \text{ inches/ms} * 25.4 = 0.61 \text{ m/sec}$$

Converting this to metric units will give us 0.61 m/sec as a top speed at which our robot can go to account for any delay in our processing pipeline. We might opt to go slower than this speed for any safety concerns, but will not go over this speed at all.

In addition, we have also provided a tolerance analysis for our power subsystem to ensure all components are sufficiently powered to work, including robot, docking station, and remote control. If any of these systems fail, and do not have enough power / or too much power to continue working as intended,o our project will not be able to work as intended. To combat this, we plan on using a printed circuit board that acts as a power distribution board to regulate the voltage. Since we have components that operate in different voltages (3.3 V, 5 V,6 V), we need to add in some voltage regulators. For example to calculate the power dissipation of such a regulator can be found by using the following formula:

$$Pd = Iout*(Vin - Vout).$$

In addition to the voltage, it is also necessary to regulate the amount of current supplied to each component. As calculated, the minimum current required to power all of the components is 250 + 80 + 500 + 100 + 4*250 + 30. We also want to make sure that we have some sort of buffer to make sure that the current does not fall below the required current to ensure this we would have enough current to ensure smooth operation. We calculated keeping a 20 percent buffer would be sufficient to make sure all components are properly powered, giving us at least 2352 mA should be supplied to the system at all times.

3. Ethics and Safety

Ethical Concerns

1. Privacy and Data Protection:

The robot's use of cameras to map the golf green and identify divots raises potential privacy concerns. We will implement robust data privacy measures to mitigate these risks, including anonymization techniques and secure data storage. Furthermore, we will obtain permission from golf club owners and operators prior to deploying the robot.

2. Accountability and Liability:

A critical ethical consideration is determining who is responsible for the robot's actions and any potential damages. We will establish clear guidelines for the robot's operation and potential liability to address this issue. Additionally, we will implement safety features to minimize the risk of accidents or damage.

3. Environmental Impact:

The robot's use of batteries and potential for accidental damage to the golf green raises environmental concerns. We will select environmentally friendly battery options and design the robot to minimize its impact on the golf course. We will also implement safeguards to prevent accidental damage.

Safety Considerations

1. Robot Safety:

The robot must be designed and operated safely to prevent accidents or injuries. To ensure robot safety, we will adhere to relevant safety standards, such as those outlined by the Robotics Industries Association (RIA). We will also implement safety features like obstacle avoidance and emergency stop mechanisms.

2. Human Safety:

The robot must not pose a risk to human safety. To mitigate the risk of human injury, we will ensure that the robot is designed and operated to minimize the risk of collisions or other hazards to golfers or course staff.

3. Property Damage:

The robot must be designed to avoid damaging the golf course or its infrastructure. To prevent property damage, we will implement safeguards to prevent damage to trees, irrigation systems, or other valuable property.