Milestone 4 - Requirements Specification

Project : Propbot Autonomous Robot

Authors : Team- 50,

Amr Almoallim (AA) - 83386714,

Apoorv Garg (AG) - 39485545,

Cole Shanks (CS) - 54950860,

Johanan Agarwal (JA) - 29188166,

Sajjad Al-Kazzaz (SA) - 23401565

Affiliation : UBC Radio Science Lab

Table of Contents

1. Background

The team being led by Dr. Michelson at the UBC Radio Science Lab has been conducting research on 5G wireless technology. Their research has implications in the fields of marine transportation systems, road safety systems, and the national defense sector. A part of their research requires access to wireless propagation data from all around their network. Currently, the collection of this data requires the tedious and manual labor of 1-2 graduate researchers hauling a cart full of expensive equipment around an area of research. That is why our client has proposed Propbot, a robot that can carry out the data collection autonomously.

1.1. Current Status of Propbot

The Propbot project has been worked on by the UBC RSL researchers (2017-2018) and by a previous capstone team (2019). Figure 1.1 shows the timeline of the project.

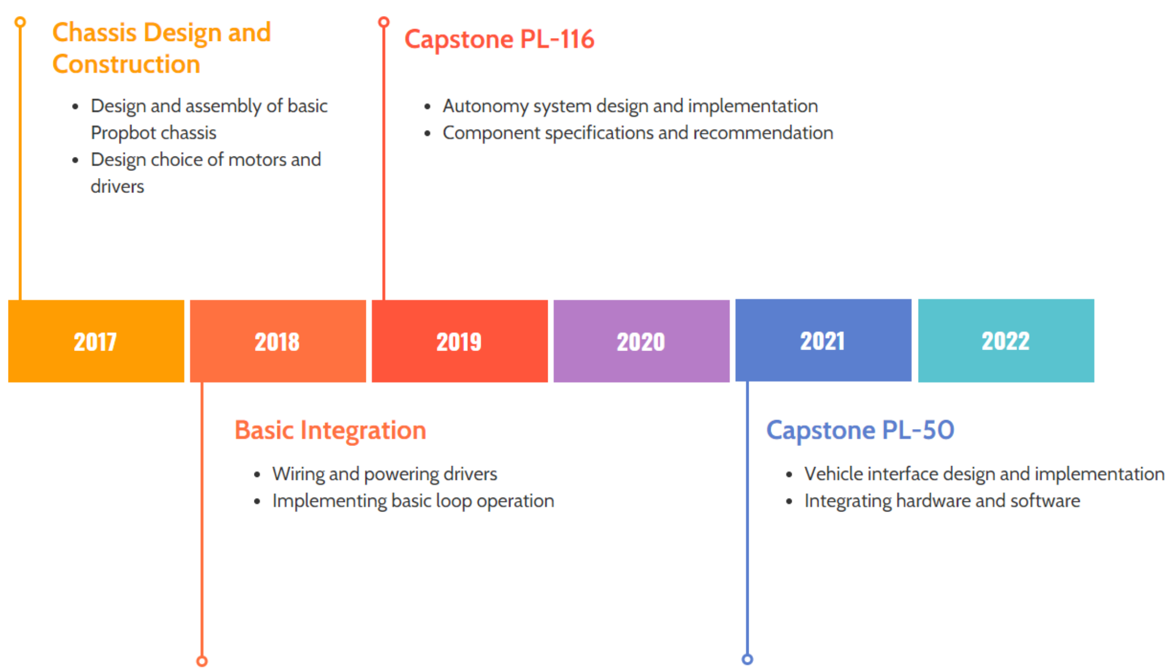

Figure 1.1 Project Timeline

1.1.1. Existing Mechanical System

The current mechanical system of Propbot consists of a full chassis, six wheels, and six hydraulic shock absorbers as depicted in figures 1.2 and 1.3 below.There is also a red enclosure with racks used for carrying a payload. There is additional hardware called the “self-leveling platform” whose responsibility is to ensure the measurement equipment is balanced. The self-leveling platform is currently not operable**.**

Figures 1.2 and 1.3 Propbot Chassis

1.1.2. Existing Hardware System

The current hardware system of Propbot contains six motor drivers, six brushless DC (BLDC) motors, two 36V 15Ah batteries, an Nvidia Jetson TX1, and an Arduino Uno (basic microcontroller). The 2 x 36V 15Ah batteries are out of commission and cannot be used. The current wiring of components on the Propbot is unclear and there are no schematics for the same. The motor drivers do not have a datasheet and only have a product label in Mandarin. The motor and wheel form a single unit and do not contain a brake rotor to facilitate mechanical braking. This constraint makes mechanical braking unfeasible for the scope of this project but remains a potential feature to be addressed by future capstone teams. The previous capstone team, who worked on Probot before us, made some hardware component suggestions but were not able to implement or purchase them due to COVID-19 restrictions.

1.1.3. Existing Firmware System

Due to COVID-19 risks, the previous team decided to implement a small-scale RC(Remote Control) [4] car to mimic the movement of Propbot. The RC car's motors are dead. The firmware written to operate the car contains two libraries, one library receives information from a remote controller and converts it into commands (move forward, move backward, turn right, turn left). The other library uses PWM signals to interface with the motor driver that is on the car.

1.1.4. Existing Software System

There exists a standalone autonomy package that is responsible for managing object detection, localization, control, and communication of Propbot. It is developed using the ROS framework in C++. It is run on the autonomy computer, which is an Nvidia Jetson TX1. The autonomy package is not integrated with any other component of Propbot and has only been validated through simulation. In addition to the autonomy package, a Mission Command Center (G.7) was developed by the previous team. The Mission Command Center is responsible for communicating with Propbot at a distance. It contains a GUI that a user can use to send Propbot on a new mission. The GUI also receives and visualizes information received from Propbot.

1.2. Our Contribution

Our team's contributions to Propbot can be broken down into three distinct categories. First, we intend to validate and verify the previous team's design. Second, we intend to integrate the hardware that was proposed by the previous team with the existing firmware and also add new design elements that we have selected. Finally, we intend to incorporate the autonomy software into the completed design so that Propbot can move autonomously.

1.2.1. Validation and Verification of Existing Systems

In this project we validate the existing hardware component choices and the autonomy software package that the previous capstone team has developed.

1.2.2. Legacy Device Integration and Proposed Additions/Changes

In this project we procure necessary hardware components and integrate them with the existing state of Propbot. This would involve:

- Writing firmware on the Arduino Mega to navigate Propbot in response to incoming RC commands
- A power management system consisting of new batteries and power circuitry
- Creating firmware for integrating the sonar sensors and E-stop

1.2.3. Autonomy Package Interfacing

We intend to connect the autonomy package to the hardware infrastructure. This means we will provide the autonomy package with data it needs to operate by interfacing a LiDAR, depth camera, GPU, and IMU devices with the autonomy computer. The autonomy package's output will be communicated to the rest of the system to allow it to influence the movement of Propbot.

2. Project Outcome

Traditional data collection for propagation measurements is a very time-consuming and tedious process. Figure 1.4 shows why this process of measurement is very inefficient. There are various factors to consider when calculating the total amount of time for a data collection session. These include the type of measurements, the number of points of interest (POI), the environment in which the measurements are being collected, and the proximity of the points of interest to each other. Propbot intends to provide a solution to this lengthy process and decrease the time it takes to travel to and from Waypoints (G.12) as well as eliminate the need for researchers to be directly involved with the measurements.

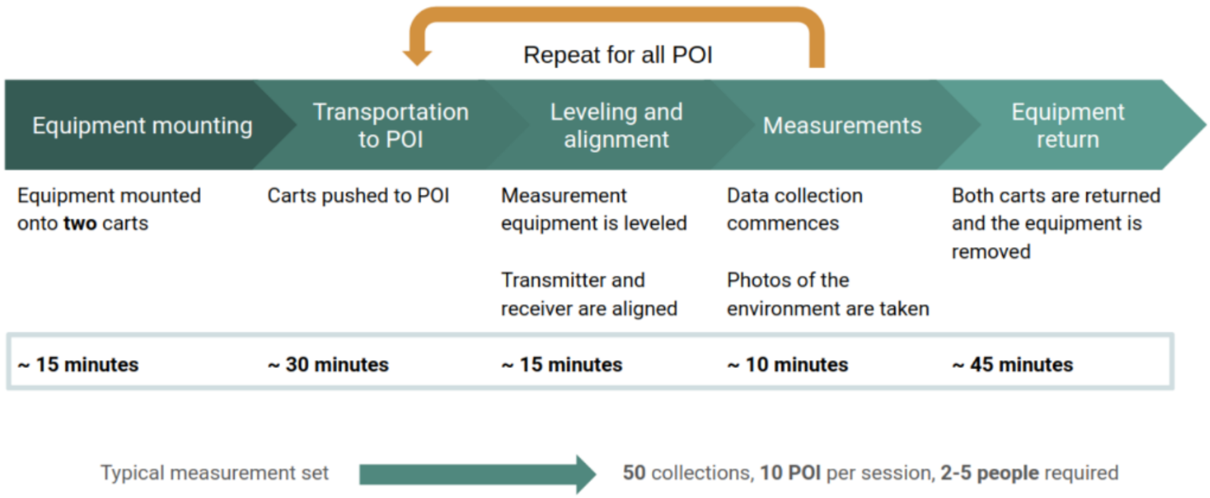

Figure 1.4 Traditional Propagation Measurement Data Collection Process

Our contribution to Propbot will significantly aid the UBC Radio Science Lab's ability to obtain high-quality propagation data at various waypoints around the UBC campus (Figure A.1). Teleoperation of Propbot will improve both the efficiency and reliability of this data collection because the RSL team will no longer have to resort to burdensome methods such as wheeling a cart around carrying measurement equipment. Furthermore, the modular design of the various sub-systems will allow future researchers to swap out or enhance individual components thus leading to increased flexibility in terms of future design decisions. Improvements and advancements to the legacy system will also be of great benefit. For example, by upgrading the autonomy computer to the Nvidia Jetson NX, Propbot will now be fully able to run its advanced sensors and autonomy software without limitations due to computing power. The battery system overhaul will increase the runtime of Propbot, allowing for longer duration data collection sessions and therefore more data for UBC RSL to interpret and process. The re-worked power management system will also simplify the existing connections and allow for future ease of use for adding new components. Additionally, The added safety features will ensure that Propbot can operate reliably without posing a threat to itself or the surrounding environment. Finally, by verifying and validating the simulations done by the previous capstone team, in addition to the hardware advancements we implement, future teams will be in excellent condition to complete Propbot's desired goal of being a data collection robot capable of SAE level 3 autonomy (Figure A.2).

3. Methodology

Since this project is a continuation of a previous team's work, the majority of core requirements remain unchanged. We were influenced by the same NHTSA framework [1] that the other team followed, as well as an updated version of Transport Canada's guidelines for automated driving systems [2]. The main idea was to list Operational Design Domains (ODD) (G.8), which would provide us with the right frame to identify capabilities Propbot needed to have.

3.1. Extensions to ODD

The following extensions to the operational design domains were made. They make it explicit what we expect from the user in terms of where and when Propbot should be operated, based on the current limitations of the system.

Refer to the Requirements_Specification document from the previous team for a list of the original operational design domains. This document can be accessed from the previous teams Github repository:
https://github.com/hannahvsawiuk/PropBot/tree/master/docs/Reports

Tag	Requirements
ODE1	Propbot shall not operate on inclines steeper than 6 degrees
ODE2	Propbot shall run for a max allotted time of 1.5hrs

Table 1.1 Extensions to ODD Requirements

4. Project Constraints

Constraints can be divided into two categories: constraints associated with the project definition, and constraints that are based upon the design decisions made by the previous team (legacy).

Tag	Constraint	Constraint	Description

		Type	
C1	ROS Software Framework	Legacy	ROS constrains us to use a publisher/subscriber design pattern or ROS library for any improvements to the software
C2	Simulation Package	Legacy	A legacy system that is the main method of validating autonomy
C3	Mission Command Center	Legacy	Mapviz library and GUI has already been created and are not within our scope to change
C4	Autonomy Package	Legacy	A significant amount of code has been implemented for the autonomy system. Selected libraries include the use of Google Cartographer for localization and YOLO for object detection.
C5	Chassis	Project Definition	The chassis has already been designed and fabricated
C6	Motors/Wheels	Project Definition	The wheels and motors form a single unit and are integrated with the chassis
C7	Differential Drive System	Legacy	Software for both the vehicle interface system and autonomy system has been implemented to support a four motor differential drive system (G.4)
C8	Suspension System	Project Definition	The suspension shocks are integrated with the chassis
C9	Teleoperation Protocol and RC controller	Legacy	The Arduino code and teleoperation communication protocols have been largely established by the legacy system. This informed our purchase decision for the

			transmitter and receiver
C10	Intel Realsense Depth Camera	Legacy	The component has already been scoped and purchased by the previous team
C11	Motor Controllers	Project Definition	The existing motor controllers are being used instead of the proposed Roboteq motor controllers due to supply-chain related issues in acquiring them
C12	Equipment Rack Module	Project Definition	The rack module which is used to store RF data collection and measurement equipment is already fabricated and mounted to Propbot
C13	Self-Leveling Platform	Project Definition	Takes up space on the back of the red box, where LiDAR could be placed
C14	Daytime operation	Project Definition	The robot shall operate during the daytime because the camera and LiDAR sensors needed adequate light to identify objects
C15	Operational location	Project Definition	The robot shall operate within the boundaries of the UBC Vancouver Campus
C16	Ultrasonic Sensors	Legacy	The sonar sensors have already been scoped and purchased by the previous team

Table 1.2 Project Constraints

5. Functional & Non-Functional Requirements

5.1. Navigation

Tag	Related Constraints	Functional Requirements
F1.1	C6, C8	The robot shall be able to accelerate, decelerate and stop when in

		remote control mode (G.10) or autonomy mode (G.1)
F1.2	C10	The robot shall be able to move longitudinally when in remote control mode or autonomy mode
F1.3	C4	The robot shall be able to execute turns
F1.4	C18	The robot shall not collide with any object while in motion
F1.5	C4	The robot shall be able to take a set of coordinate waypoints and derive a continuous path that connects those waypoints
F1.6	C4	The robot shall come to a full stop when it reaches a specified waypoint
F1.7		The robot shall have enough battery power to run all core navigation components (G.3) for 1.5 hours
F1.8	C4	Robot speed shall be configurable in both autonomy and remote control mode

Table 1.3 (Navigation) Functional Requirements

Tag	Related Constraints	Non - Functional Requirements
NF1.1	C4	The path generated by the robot shall be the shortest valid path that is within the geofence and ODD (Figure 1.1)
NF1.2	C18	The robot shall come to a full stop within 0.5m of the brake being triggered
NF1.3		The robot shall be able to execute locally generated control signals from the autonomy module

NF1.4		All navigation components must be supplied with power according to their specifications
NF1.5		The minimum robot speed when in motion shall be 1km/hr
NF1.6		The maximum robot speed when in motion shall be 5km/hr
NF1.7		The robot’s movement shall be fluid

Table 1.4 (Navigation) Non-Functional Requirements

5.2. Decision Making

Tag	Related Constraints	Functional Requirements
F2.1	C4	The robot shall navigate around static objects in front of it when in autonomy mode
F2.2	C4	The robot shall come to a full stop if no suitable path is available
F2.3		The robot shall come to a full stop if an object cannot be avoided
F2.4	C4	In the absence of obstacles, the robot shall move towards its next waypoint when it autonomy mode
F2.5		The robot shall have enough battery power to run all core decision making components (G.2) for 1.5hrs

Table 1.5 (Decision Making) Functional Requirements

Tag	Related Constraints	Non - Functional Requirements
NF2.1		The robot shall maintain a distance of at least 0.5m from surrounding objects

NF2.2	C4	Movement decisions shall be updated at no less than 10Hz
NF2.3		The robot shall be able to take in data from an array of sensors and perform localization
NF2.4		All decision components must be supplied with power according to their specifications

Table 1.6 (Decision Making) Non-Functional Requirements

5.3. Safety

Tag	Related Constraints	Functional Requirements
F3.1	C3	The robot shall be operable in autonomy mode and fall-back user mode
F3.2		The fall-back user should be able to manually switch the robot operating mode
F3.3		The robot shall have an onboard switch that kills power to the system
F3.4		The robot shall be enabled with a Physical E-Stop (G.9) and Remote E-Stop (G.11)
F3.5	C4	The robot shall send the mission command center a warning whenever it switches over to remote control mode
F3.6		The robot should only operate if an RC transmitter and receiver are paired
F3.7		The robot shall have visual warning device to alert nearby people when moving

F3.8	C4	The robot shall detect an incline/decline greater than 2 degrees

Table 1.7 (Safety) Functional Requirements

Tag	Related Constraints	Non - Functional Requirements
NF3.1		Switching between remote and autonomy mode should occur in no more than 0.5s
NF3.2		Physical E-Stop should be in an accessible place in case of emergency

Table 1.8 (Safety) Non-Functional Requirements

5.4. Communication

Tag	Related Constraints	Functional Requirements
F4.1		The robot shall be teleoperable with an RC transmitter up to a distance of 100m.
F4.2	C4	The robot shall communicate with the mission command center over a cellular network
F4.3		The robot shall communicate real-time location to the mission command center
F4.4	C3	The user shall be able to upload missions via Mission Command Center
F4.5		The robot shall communicate cellular signal strength data to the mission command center

Table 1.9 (Communication) Functional Requirements

5.5. Data Collection

Tag	Related Constraints	Functional Requirements
F5.1		The robot shall collect time-stamped data at a sampling rate of 1Hz for the location
F5.2		The robot shall collect time-stamped data at a sampling rate of 1Hz for the motor speed for each motor
F5.3		The robot shall collect time-stamped data at a sampling rate of 1Hz for the command status from both the mission command center and manual control transmitter
F5.4		The robot shall collect time-stamped data at a sampling rate of 1Hz for the cellular signal strength

Table 1.10 (Data Collection) Functional Requirements

Tag	Related Constraints	Non - Functional Requirements
NF5.1		The time-stamped location data must be accurate within 0.5m

Table 1.11 (Data Collection) Non-Functional Requirements

6. Glossary

ID	Term	Definition
G.1	Autonomy Mode	Autonomous mode of control in which Propbot operates via a preprogrammed path specified by the mission command center

G.2	Core decision making components	Refers to the Jetson NX, Velodyne Puck, Intel Realsense, GPU, IMU, and cellular link
G.3	Core navigation components	Refers to the motor, motor driver, and Arduino
G.4	Differential Drive System	Vehicle drive system which facilitates independent control for each set of wheels. Sets of wheels on each side of Propbot are controlled independently
G.5	Electronic braking	Braking of a brushless DC motor performed electronically from the motor controller
G.6	Fall-back User	User present within line of sight of the robot who has access to the RC transmitter and can control the robot in remote control mode
G.7	Mission Command Center	The user interface that allows user to monitor and send location commands to Propbot in autonomy mode
G.8	Operational Design Domain (ODD)	Constraints regarding the operation of Propbot as set forth by previous capstone teams and briefly expanded upon
G.9	Physical E-Stop	Physical switch that when activated cuts power from the battery to all electronics on Propbot
G.10	Remote Control Mode	Teleoperable mode of control in which Propbot can be navigated using an RC transmitter
G.11	Remote E-Stop	Software-based control that when activated stops operation of all motors

G.12	Waypoints	Specified stopping locations for data collection during autonomy mode. Waypoints are set via the mission command center.

Table 1.12 Project Glossary

7. References

1. “Automated Driving Systems - NHTSA.” [Online]. Available: https://www.nhtsa.gov/sites/nhtsa.gov/files/documents/13882-automateddrivingsystems_092618_v1a_tag.pdf. [Accessed: 18-Oct-2021].
2. “Automated Driving Systems: A vision for safety - NHTSA.” [Online]. Available: https://www.nhtsa.gov/sites/nhtsa.gov/files/documents/13069a-ads2.0_090617_v9a_tag.pdf?ref=hackernoon.com. [Accessed: 18-Oct-2021].
3. Hannahvsawiuk, “Home · Hannahvsawiuk/Propbot Wiki,” GitHub. [Online]. Available: https://github.com/hannahvsawiuk/Propbot/wiki. [Accessed: 18-Oct-2021].
4. “SAE International Releases Updated Visual Chart for its ‘levels of driving automation standard for self-driving vehicles,” SAE International, 12-Dec-2018. [Online]. Available: https://www.sae.org/news/press-room/2018/12/sae-international-releases-updated-visual-chart-for-its-%E2%80%9Clevels-of-driving-automation%E2%80%9D-standard-for-self-driving-vehicles. [Accessed: 18-Oct-2021].
5. “Tc.canada.ca.” [Online]. Available: https://tc.canada.ca/sites/default/files/2021-09/automated_driving_system_report_en.pdf. [Accessed: 18-Oct-2021].
6. “The University of British Columbia,” UBC Radio Science Lab. [Online]. Available: http://rsl.ece.ubc.ca/. [Accessed: 18-Oct-2021].
7. “The University of British Columbia,” Wayfinding at UBC Vancouver. [Online]. Available: http://www.maps.ubc.ca/PROD/index.php. [Accessed: 18-Oct-2021].

A. Appendix

Figure A.1 - Map of UBC Vancouver campus

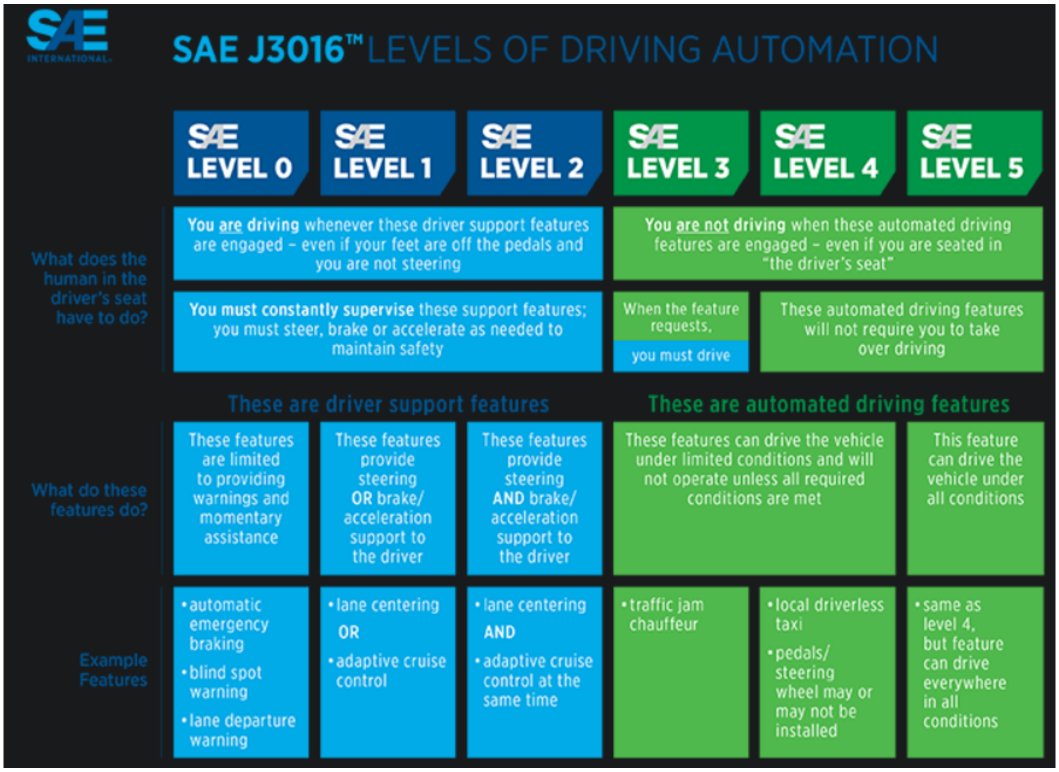

Figure A.2 - SAE levels of Autonomy