Project Report

A Forest Planter Robot

Autonomous Mobile Robotics

ELEC-E8111

Team Members

Niladri Dutta - 890812

Alladi Janesh - 890964

Shubham Jaiswal - 914840

TABLE OF CONTENTS

1 Introduction

Autonomous vehicles and robots are becoming increasingly common in the industrial automation and mobility sector. In order for them to operate in dynamic and complex environments, detailed perception and mapping of the surrounding environment are highly critical. Sharing the same environment with humans, animals, vehicles, obstacles and other dynamically moving objects requires smart decision making capabilities too.

In this project we aim to tackle this problem in the agriculture sector of autonomous mobile robots. We aspire to design a fully autonomous forest planter robot capable of planting seedling cassettes in soil with minimum human interaction. The terrain map and plan for planting is provided by the user.

1.1 Problem Definition

This project tries to solve the problem of reducing the labor of planting seedlings in forest terrain using autonomous systems. The plant seedlings have been given in cassettes. To solve this problem we have used 2 robot systems and one base station. The primary robot will be used for planting the seedlings on the field, the secondary robot is used for restocking, refueling the primary robot, the base station is the place where the seedling cassettes stock and the charging station is present.

1.2 Objectives

- The robot platform
- The perception and measurement devices
- Way of operation
- Algorithms, described at schematic level
- Human Robot Interface
- A priori information: terrain map and plan for planting is given
- Schematic structure, perception systems and locomotion mechanics should be designed
- The way of operation and control systems should be designed
- Energy management for the machine platform
- The seedling logistics is based on cassettes containing single seedlings ready for planting
- Don’t need to solve problems related to watering
- Operation must be safe, especially in the hilly areas

1.3 General Constraints & Assumptions

- The path plan is known beforehand.
- We assume that the map is shaped as a rectangle.
- The user will break down larger regions into rectangles of a given maximum area.
- Two base stations with seedling stock are available at diagonally opposite ends of the rectangular area.
- The edges of the map have a fixed margin which is empty, so that the robot is able to maneuver freely.
- We assume that the robot is a four wheel Ackermann drive model.

- We assume that the robot will plant the seedling equidistantly and in a line parallel to the longer side of the rectangular map.
- We assume that the GPS is reliable and functionable at all times.
- We assume that we know the storage capacity of the seedling cassette tray in the robot and also the battery status at all times.
- We assume that at any given time a secondary robot can navigate to the primary robot.
- We assume that at any given time the secondary robot can communicate with the primary robot.
- We assume that the user will monitor the weather conditions and deploy the system accordingly.

2 System Requirements, Architecture and Specifications

2.1 Basic Definitions

- Primary Robot: The robot which is responsible for planting and holding the seedling cassettes.
- Secondary Robot: The robot which is responsible to restock seedling cassettes and has a fully charged battery pack to be swapped with the primary robot.
- Base Station: A location near the field which provides electrical power and seedling cassettes. Secondary robot will dock with the base station for charging the battery pack and replenishing the seedling cassette stock.
- Docking port: A hardware module present on both the primary and secondary robot, used to dock the two robots with each other.
- Seedling Cassettes: A small container made of biodegradable material containing a seedling along with some soil.

2.2 Requirements

- The primary robot shall identify its position with respect to the surrounding environment.
- The primary robot shall read and monitor the current battery charge and health.
- The primary robot shall read and monitor the current plant cassette status.
- If the primary robot's charge reaches the reserve capacity, a secondary robot shall dock with it and refuel the charge along with the restocking of cassettes.
- The primary robot shall calculate the current speed, steering angle of the vehicle.
- The primary robot shall detect humans/animals and will stop until they clear the path.
- The primary robot shall adjust it's local planner to avoid collision with obstacles.
- The primary robot shall take a turn if it reaches the end of the rectangular map.
- The primary robot shall turn on the headlights when the camera detects low light conditions.
- The primary robot shall remain stationary and stop the planter tool when it is low on battery or when it is low on the plant cassettes.

- The primary robot communicates its location to the secondary robot when it is low on battery or when it is low on the plant cassettes.
- The secondary robot shall navigate to the primary robot's location when requested.
- The secondary robot shall plan a path, avoiding planted areas, to dock with the primary robot to refuel and restock.
- On reaching the end of the map, the primary robot shall remain idle and await for further user inputs.
- If the robot cannot plan a path it shall remain idle and alert the user.
- If any of the parts malfunction, the robot shall remain idle and alert the user.
- The primary robot is powered by a Li-Ion Battery which is recharged by a solar panel.
- The secondary robot is powered by a Li-Ion Battery which is recharged by the base station.
- The base station shall use solar energy to recharge the batteries.
- The primary robot shall have a docking port that can dock with the secondary robot for restocking the plant cassettes and swapping the battery.
- The base station shall have a docking port that can dock with the secondary robot for restocking the plant cassettes and swapping the battery.
- The primary robot shall have 4 planter tools, each having the capacity to plant one row of seedlings.
- The primary robot shall have the ability to configure the distance between consecutive 2 rows.
- The primary robot shall stop the planter tool when either the battery or seedling cassette is below a certain threshold.
- The secondary robot shall dock with the primary robot or the base station using the docking port.
- The secondary shall replace the empty battery from the primary robot with a charged battery, after the docking process with it.
- The secondary shall replace the empty battery with a charged battery from the base station, after the docking process with the base station.
- The primary and secondary robots shall monitor its components and alert the user is there is any malfunction.
- The primary and secondary robots shall consider the area planted in the past as obstacles, while planning a path for navigation.

2.3 System Architecture

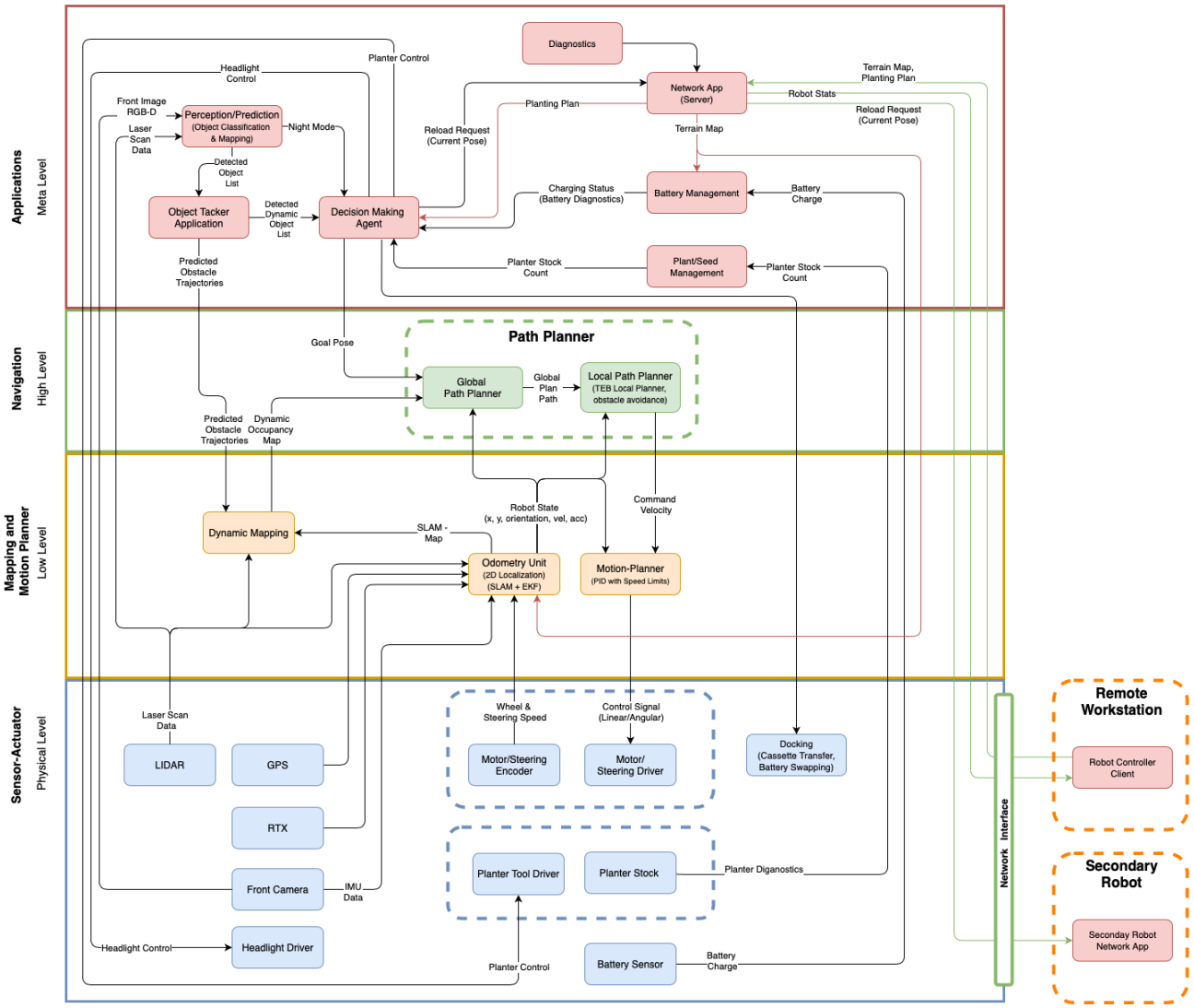

Figure 1: *System Architecture*

2.4 System Specifications

2.4.1 General Framework

The system design is based on the publisher-subscriber model for maintaining communication between the various hardware and software subsystems. The open source meta- operating system - Robotic Operating System (ROS), is used as the framework to operate such a model. Using a dis- tributed network, where each subsystem(hardware or software) is called a "node", which runs a process for the specified task. Each subsystem node can exchange data with other subsys- tems by publishing topics or subscribing to topics which are continuous streams of data in a specified message format.[22]

2.4.2 Methodology

Physical Level

Sensors

The sensors are the input to the robot so as to adjust to the environment it is in. The sensors used here include LiDAR, GPS, RTX, Front Camera. The LiDAR is used for detection and avoidance of obstacles on its path. GPS is used for localization and to navigate through the field when it is in motion. RTX belongs to the family of GPS which is used to provide more accuracy and precision for location and mapping when the robot is in motion. Front Camera is used for detecting the path and obstacles using image and depth sensors present in it. The IMU sensor is used for measuring the pose of the robot.

Actuators

These parts consist of Motor Encoder, Motor Driver which is used to maneuver the robot on the field. Planter tool driver to rotate the planter tool. The wheel speed and steering direction is given as an input to the Odometry Unit. Motor Driver gets the control signal input from the motion planner and adjusts the speed and steering direction according to the planner.

Planter Tool & Stock

Planter Stock: Seedling cassettes are assumed to be in the form of plant tape as shown in figure below. Each cassette is connected to each other in series in the form of a belt. The material of the belt itself is biodegradable and gets dissolved in the soil after a certain period of time. These series of cassettes are stored in trays as shown in figure below[34].

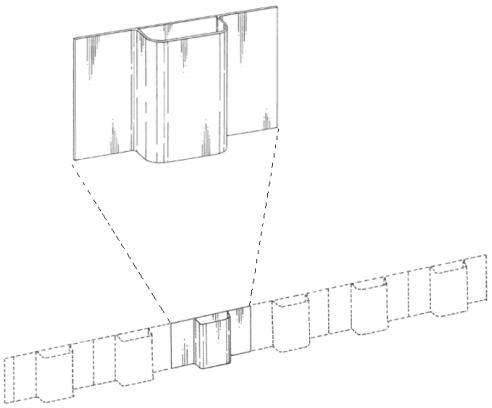

***Figure 2**: Seedling cassette tape and a single seedling cassette*

***Figure 3**: Image of an actual seedling cassette tape*

***Figure 4**: Image showing seedling cassette tray*

Planter Tool:

A tool which is responsible for sowing the seedling cassette inside the soil. Figure below[33] shows the planter tool we are using for this project. The tool consists of a series of spikes arranged in a circular fashion. The planter tool motor driver rotates the tool at a certain speed. The speed determines the distance between two cassettes that are sown in the ground. With this tool, our robot can plant the seedling at a very high rate and with a consistent accuracy.

Planter stock also communicates with the Plant management module regarding status of availability of seedling cassette, which in turn communicates with the Decision making agent. If seedling cassettes are at critical level, then the decision making agent will stop the planter tool and robot and communicate with the network server to given seedling cassette command to the secondary robot.

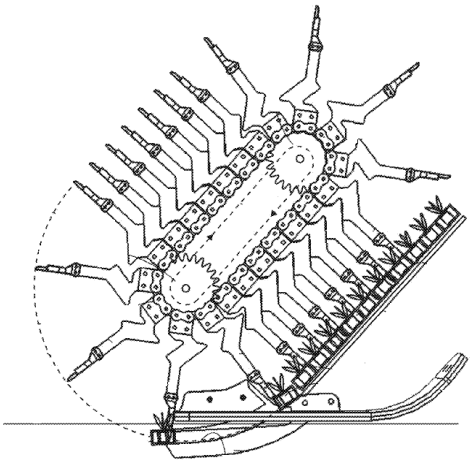

***Figure 5**: Design of the planter tool*

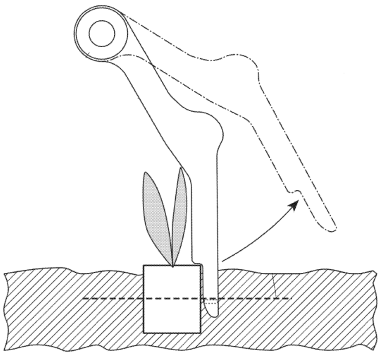

***Figure 6**: Image showing a single seedling cassette being planted by a single spike of planter tool*

The robot will have 4 planter tools working parallely and sowing four rows of seedling cassettes at a time as shown in figure below. The seedling transfer unit is responsible to feed in the seedling cassette to planter tool. Arrows on the green strips indicate the motion of seedling cassette tapes towards the planter tool. Each cassette of seedling is cut from the series as soon as the spike of planter tool gets attached with the individual cassette.

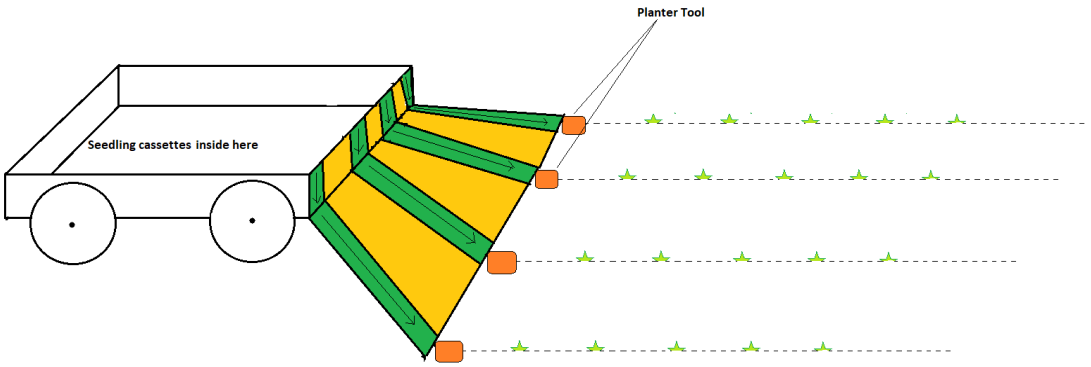

***Figure 7**: Primary robot with trailer and plater tool.*

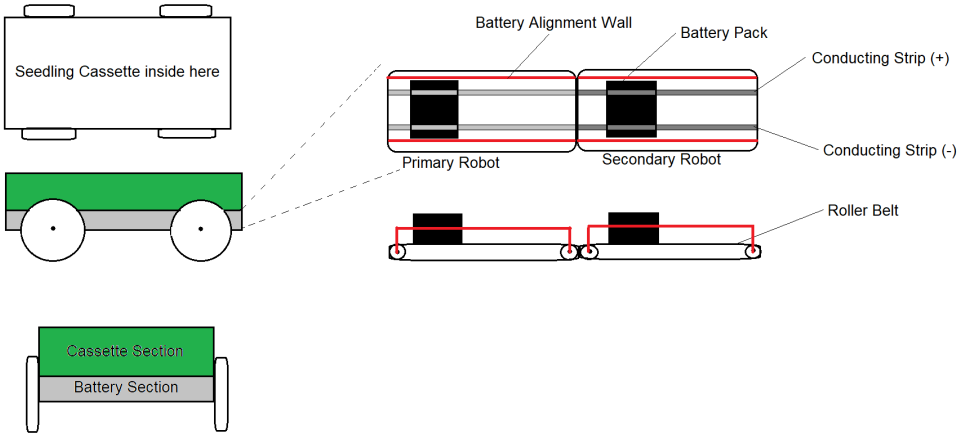

Figure 8: Battery section of primary robot

Docking System

The docking system consists of two sub modules which are activated after docking with the secondary robot. These are the plant Cassette Transfer module and the battery swapping module.

Seedling Cassette Transfer

After docking, the cassette transfer happens usings a piston which pushes the seedling cassettes from the secondary robot to the primary robot. The visuals of the process in given in the figure below.

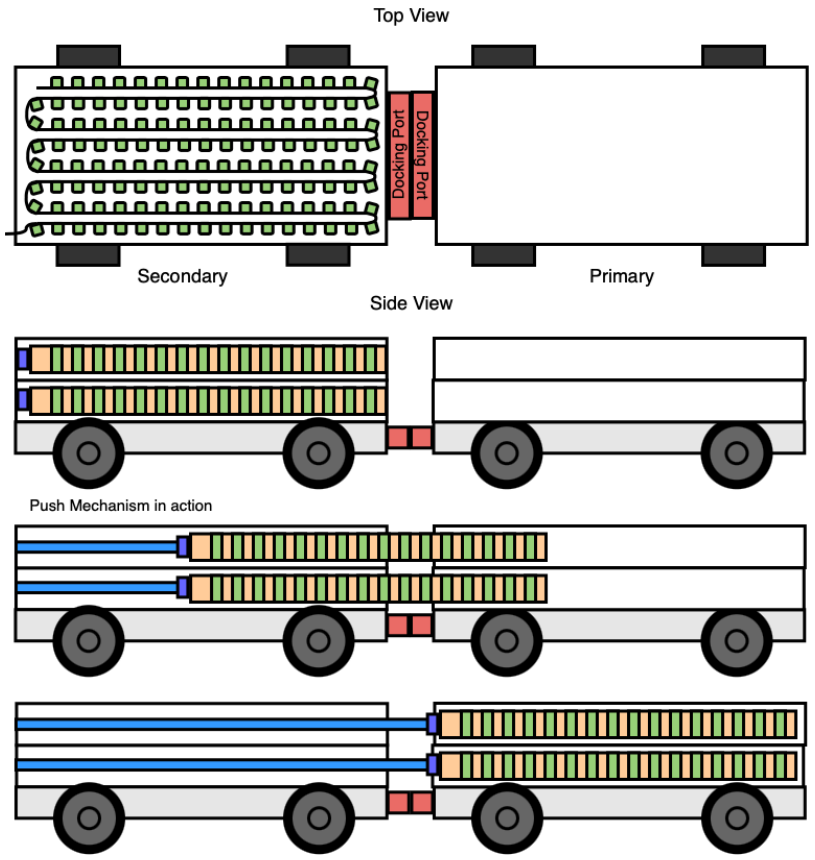

Figure 9*: Swapping of battery pack between Primary and Secondary robot*

Battery Swapping Module

Recharging the batteries of the primary robot with minimum delay is a challenging task. Instead, a fully charged battery pack is swapped with the drained out battery pack of the primary robot. The primary robot will have a battery section as shown in figure. In our system, the robot will do an automatic battery swap with a secondary robot. For this, a secondary robot will dock with

the primary robot and perform the swapping operation as shown in figure below. The primary robot also has a non-removable, rechargeable backup battery pack which is only used during swapping of battery. The battery section will have a roller belt with two conducting strips on it. The battery pack is such that its anode and cathode terminal will be in contact with the conducting strips at all times. When a swapping is required, a secondary robot will dock with the primary robot, a fully charged battery pack is transferred to the primary robot, and the drained battery pack is then transferred to the secondary robot as shown in figure below.

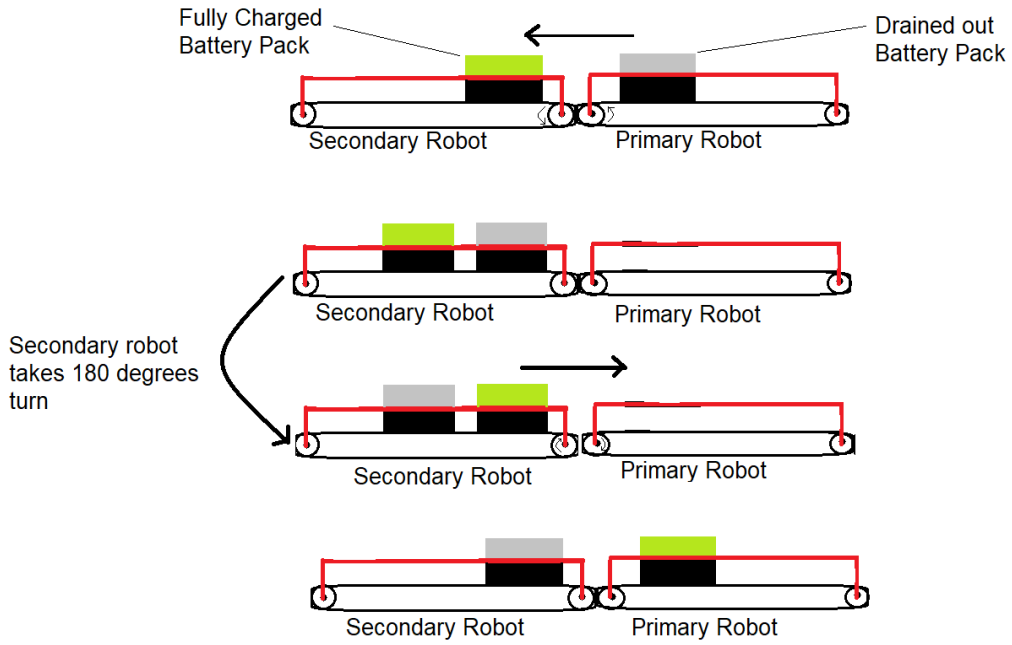

***Figure 10**: Swapping of battery pack between Primary and Secondary robot*

2.4.2.1 Low Level

Odometry Unit

This unit is used to estimate the change in position of the robot with respect to the starting location. This unit is configured using Fast SLAM to localize the robot on the map. Fast SLAM uses Extended Kalman Filter which belongs to a family of particle filters. The algorithm of Fast SLAM follows by repeating the prediction and measurement update M times, where M is the number of particles and resampling the data until each particle is sampled. The sampled data is then fed into the Motion planner block and the Dynamic mapping block.

Dynamic Mapping

It is a unit that uses the data from the Odometry unit, the object tracker application and the path planner to constantly update the map to adjust to the environment.

Linear and Angular Motion Planner

This unit contains a PID controller that uses velocity data from the path planner and the odometry data to provide the control signal to the motor driver unit.

2.5 High Level

Path Planner

This part explains the navigation methods chosen in the project. Here we present our concept of a subsystem forming the Path planner together with units supplying and receiving from it. It can be seen that it is directly connected with the Odometry Unit which processes the odometry data and feeds it to the other units seen in the figure.

a) ***Global planner:*** For that purpose the standard approach has been chosen- the global planner included in the navigation package also called navfn.[29] It's default configuration uses Dijkstra's algorithm (breadth-first search to check the costmap cells for the optimal path). It can be replaced by A* algorithm.

Optimal here is defined by the path's potential (relative cost of a path based on the distance from the goal and from the existing path itself). A helper (Quadratic approximation) is used in the calculations of the potential. The package also offers a substitute to that called Simple Potential Calculation.

Main drawback of A*'s path is that it may not be optimal in an 8-connected sense, but it can be faster. Frequency and patience of a global planner is usually 3 times lower than that of a local path planner.

b) ***Local planner:*** We decided on TEB local trajectory planner (Timed Elastic Bands) which allows configuration of different parameters like trajectory, goal tolerance etc.[30] The configuration of this package is also easy and the only issue occurs when localization is noisy, which leads to overall aggravated performance.

2.6 Meta Level

Perception/Prediction

This block takes in an RGB-D image from the Front Camera, Laser scans data from LiDAR as inputs to predict and percept the environment around it. It is useful for the robot to find a different path to avoid obstacles. This path can be mapped by feeding the output data of this block to the Object Tracker Application. The precepted obstacles can be classified into dynamic and static obstacles. Depending on the type of obstacle the robot's behaviour is defined.

Object Tracker Application

This application block is used to keep track of the objects that are obstacles on the path so that the robot can avoid it during its motion. The data from this block is fed into the Dynamic Mapping block to constantly update the map according to newly found obstacles.

Battery Management

As shown in the Figure, the battery management module receives the battery parameters such as Voltage, Current , Temperature, SOC and DOC from the battery sensor. Afterwards, the battery management outputs charging status to the decision making agent. There are three possible charging status(CS):

Low Battery Mode (L) - If the current charge is not enough to reach the final goal.
Charging Mode (C) - If the vehicle is currently at a charging point and the battery level is increasing.
Battery Full (F) - If the battery level is 100%.

As shown in Table , different levels of battery readings correspond with different battery charge statuses. When the battery readings are between 30% and 50%, battery status changes to low level. The warning of low battery level will be sent and the PWM(Pulse Width Modulation)applied to the wheel will be reduced by 10%. Similarly, when the battery readings are between 0% and 30%, battery status changes to critical low level.

Battery Reading	Battery Status	Operation
70% ~100%	High	Normal
50% ~70%	Medium	Normal
30% ~50%	Low	Reduce 10% PWM to wheel
0% ~30%	Critical Low	Reduce 20% PWM to wheel

The warning of critical low battery level will be sent and the PWM applied to the wheel will be reduced by 20%.

The battery management sends all the battery parameters to the Decision making agent module. When battery charge reaches critical level, the decision making agent stops the planter tool and robot. It also sends the signal to the network server to send a battery swapping command to the secondary robot.

Decision Making Agent

The logic for control of the primary robot is based on a state machine, which runs using the inputs from various applications as shown in the system architecture diagram.

The state machine consists of the following states(modes):

1. Idle Mode: Wait for waypoints.
2. Initial Navigation Mode: Navigate to the first waypoint.
3. Planting Mode: Plant trees

4. Reload/Recharge Mode: Send recharge/reload request with current location to the secondary robot. Wait to be recharged/reloaded.
5. Moving Obstacle Mode: Stop the vehicle and wait for the obstacle to pass.

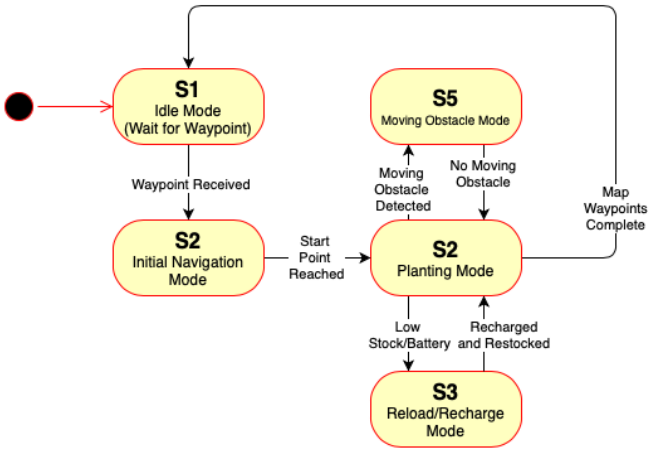

***Figure 11**: Decision Maker Module State Machine*

This module also controls the activation of headlights based on the visibility and the planter tool based on the stock of the seedling cassettes.

Network Server

The network application sets up a server which enables the clients like the is used for communication between the robot and the user who run an application as the client to provide the terrain map and the planting plan for the planter robot.

Diagnostics

The diagnostics package collects the information from the hardware components and makes it accessible to the user. Also raising an alert whenever the components have a failure.

3 Frameworks and Tools

The following tools, frameworks and libraries will be used in this project.

- Path tracking cubic splines using a stanley controller.
- ROS framework will be used to integrate all the project's software modules and components.
- OpenCV will be used for image processing and object recognition
- Tensorflow will be used for modeling the trajectory prediction of obstacles and traffic sign recognition.
- Cartographer (Filipenko et al. 2018)

- Sign Detection CNN based DNN like YOLO.
- Localize using Cartographer slam using LIDAR.
- Use OpenCV where possible for perception like lane lines.
- Combination of OpenCV and SVM/DNN for segmentation/Classification of cars + free occupied spaces.
- Obstacles: Lidar data with bounding box rules may be used to re-route to avoid dynamic and static obstacles.
- Global Planning: Using ros global_planner and A* taking into account map obstacles.
- Local Planning: base_local_planner. Should also avoid detected obstacles.
- Stanley controller and polynomial splines between intermediate goals for trajectory generation.

4 References

1. T. Litman, “Autonomous vehicle implementation predictions: Implications for transport planning,” 2015. PwC US, “Industrial mobility: How autonomous vehicles can change manufacturing,” https://www.pwc.com/us/en/ industries/industrial-products/library/industrial-mobility. html.
2. PwC US, “Industrial mobility: How autonomous vehicles can change manufacturing,” https://www.pwc.com/us/en/ industries/industrial-products/library/industrial-mobility. html.
3. I. Yaqoob, L. U. Khan, S. Kazmi, M. Imran, N. Guizani, and C. S. Hong, “Autonomous driving cars in smart cities: Recent advances, requirements, and challenges,” IEEE Network, 08 2019.
4. David Z. Morris, “Today’s cars are parked 95% of the time,” https://fortune.com/2016/03/13/cars-parked-95- percent-of-time.
5. Charles Q. Choi, “How self-driving cars might transform city parking,” https://spectrum.ieee.org/cars-that-think/ transportation/self-driving/autonomous-parking.
6. E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, “A survey of autonomous driving: Common practices and emerging technologies,” IEEE Access, vol. 8, pp. 58 443– 58 469, 2020.
7. “Ros tutorials,” https://github.com/ros/ros tutorials.
8. A. Sakai, D. Ingram, J. Dinius, K. Chawla, A. Raffin, and A. Paques, “Pythonrobotics: a python code collection of robotics algorithms,” 2018.
9. “Pythonrobotics: Repository,” https : / / github . com / AtsushiSakai/PythonRobotics.
10. M. Filipenko and I. Afanasyev, “Comparison of various slam systems for mobile robot in an indoor environment,” 09 2018.

11. M. Korkmaz, N. Yılmaz, and A. Durdu, “Comparison of the slam algorithms: Hangar experiments,” vol. 42, 12 2015.
12. R. Yagfarov, M. Ivanou, and I. Afanasyev, “Map com parison of lidar-based 2d slam algorithms using precise ground truth,” in 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV), 2018, pp. 1979–1983.
13. G. Grisetti, “Introduction to navigation using ros,”http//www.diag.uniroma1.it/˜nardi/Didattica/CAI/matdid/robot - programming - ROS - introduction - to - navigation.pdf.
14. Q. Luo, Y. Cao, J. Liu, and A. Benslimane, “Localization and navigation in autonomous driving: Threats and countermeasures,” IEEE Wireless Communications, vol. 26, no. 4, pp. 38–45, 2019.
15. I. M. D.Miklic, “Localization and navigation,” ´ https:// www. fer. unizg. hr/ download/ repository/lec05-ros-navigation.pdf.
16. M. Bjelonic,“YOLO ROS: Real-time object detection forOS,”https://github.com/leggedrobotics/darknet ros, 2016–2018.
17. Tzutalin, “labelimg,” https : / / github . com / tzutalin / labelImg. ODSC - Open Data Science, “Road lane lines detection using advanced computer vision techniques,”https://medium.com/@ODSC/overview- of-the- yolo- object detection-algorithm-7b52a745d3e0.
18. “Udacity self driving cars advanced lane lines repository,” https:// github. com/ udacity/CarND - Advanced - Lane-Lines.
19. Mithi, “Overview of the yolo object detection algorithm,” https:// medium . com/@mithi/ advanced - lane - finding - usingcomputer-vision-techniques-7f3230b6c6f2.
20. A. Borkar, M. Hayes, and M. T. Smith, “Robust lane detection and tracking with ransac and kalman filter,” in 2009 16th IEEE International Conference on Image Processing (ICIP). IEEE, 2009, pp. 3261–3264.
21. “Ros documentation,” http://wiki.ros.org/Documentation.
22. Brian Gerkey, “slam gmapping,” http://wiki. ros.org/gmapping.
23. Stefan Kohlbrecher, Johannes Meyer, “hector slam,” http://wiki.ros.org/hector slam”
24. Brian Gerkey, “slam karto,” http://wiki.ros.org/slamkarto.
25. Brian Gerkey, “slam karto,” http://wiki.ros.org/slamkarto.
26. Wim Meeussen, “amcl,” http://wiki.ros.org/robot poseekf.
27. Brian P. Gerkey, “amcl,” http://wiki.ros.org/amcl.
28. Tully Foote, “laser filters,” http://wiki. ros.org/laserfilters.
29. Kurt Konolige, Eitan Marder, “navfn,” http://wiki.ros.org/navfn.

30. Christoph Rosmann, “teb ¨ local planner,” http://wiki.ros.org/teb local planner.
31. Mateusz Przybyla, “obstacle detector,” https://github.com/tysik/obstacle detector.
32. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi,“You only look once: Unified, real-time object detection,”2015.
33. Plant Tape:Bram Gerardus Stroot: Patent No.: USD731917S
34. Plant Tape: Bram Gerardus Stroot Patent No . : US 9763382 B2