

# Autonomous Mobile Robot via Machine Learning: Proposal

Jacob Blevins, Evan Boekweg, Ilias Baali, Anupama Nair, and Pierros-Christos Skafidas

June 14, 2024

## 1 Introduction/Background

Autonomous vehicles are a large part of today's AI boom [1]. From perception and state estimation to path planning and system control, autonomous vehicles utilize a multitude of algorithms [2]. While hundreds of widely-used algorithms exist for robots in deterministic environments, unique machine learning (ML) algorithms are required for robots whose environments are highly stochastic. ML algorithms are commonly applied in advanced and highly stochastic robotic situations such as in Tesla [3] and Waymo [4] self-driving vehicles. The objective of this project is to develop autonomous, non-holonomic mobile robot ML algorithms that must navigate a Turtlebot3 through an environment to capture a care package as in Figure 1. Similar projects for intelligent mobile robots have been performed in [5] and [6].

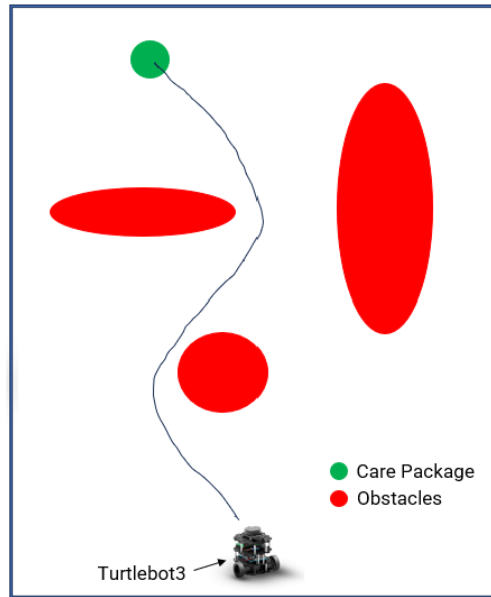


Figure 1: Mobile Robot Objective

**Dataset** Datasets are collected by the Turtlebot's sensors:

- LiDAR (LDS-02 [7])
  - Angular speed
  - Measurement angle
  - 12 point measurements for distance to objects with confidence

- Camera (Raspberry Pi Camera [8])
  - Images
  - Video
- Wheel Encoder (XL430-W250 [9])
  - Positioning
  - Velocity
  - Acceleration
  - Load (torque)
  - Goal position

## 2 Problem Definition

For robots to make decisions in stochastic environments, the aforementioned operations require methods for understanding the random variables and conditions presented to them. In the case of this project, for perception/object detection, a target objective needs to be identified. For path planning, optimized pathing needs to be determined based on the mapped environment [10]. System control should be intelligent such that the mobile robot can compensate for actuator disturbances and sensed error [11]. The solution to these stochastic problems is a fusion of ML algorithms as discussed thoroughly in Section 3.

## 3 Methods

### 3.1 Data Pre-Processing

Filtering and cleaning sensor data is an important step in the ML process and will be performed on sensor data with the following methods:

- **Density-based filtering** (DBSCAN clustering): Density-based clustering non-parametric algorithm: given a set of points in some space, group together points that are closely packed (points with many nearby neighbors), and mark outlier points that lie alone in low-density regions (those whose nearest neighbors are too far away) [12].
- **Kalman Filtering**: Kalman filtering is utilized to estimate system states given noisy data. Assuming a Gaussian distribution of sensor data, the Kalman filter quickly computes estimated states in the presence of noise using sensor measurements and predicted states [13].

### 3.2 ML Algorithms

The robot’s mission can be split into three successive tasks, each one associated to a specific algorithm:

- Object detection: Finding the objective
  - *Model*: Supervised learning - YOLO algorithm [14]
  - *Sensor*: Camera
- Mapping: Understanding the environment
  - *Model*: Unsupervised learning - CNN for Simultaneous Localization and Mapping [15]
  - *Sensors*: LiDAR and camera
- Control: Determining control commands
  - *Model*: Reinforcement Learning - PPO model combined with A\* path finding algorithm [16]
  - *Sensors*: Encoders

## 4 Results and Discussion

The goal of this project is to autonomously navigate a mobile robot to an objective, with the following scoring and expected results. Team Gantt chart (Figure 2) and contribution table (Table 1) are shown below.

- **Scoring:**

- Jaccard score for YOLO, with a goal of 0.9+
- Silhouette clustering, with a goal of 0.7+
- Minimize Cross-Entropy Loss for reinforcement learning, with a goal of  $< 10\%$  error

- **Expected Results:**

- Final position distance from extraction point: Be within 1/3 meter
- Time requirement: Reach the extraction point within 60 seconds

### GANTT CHART

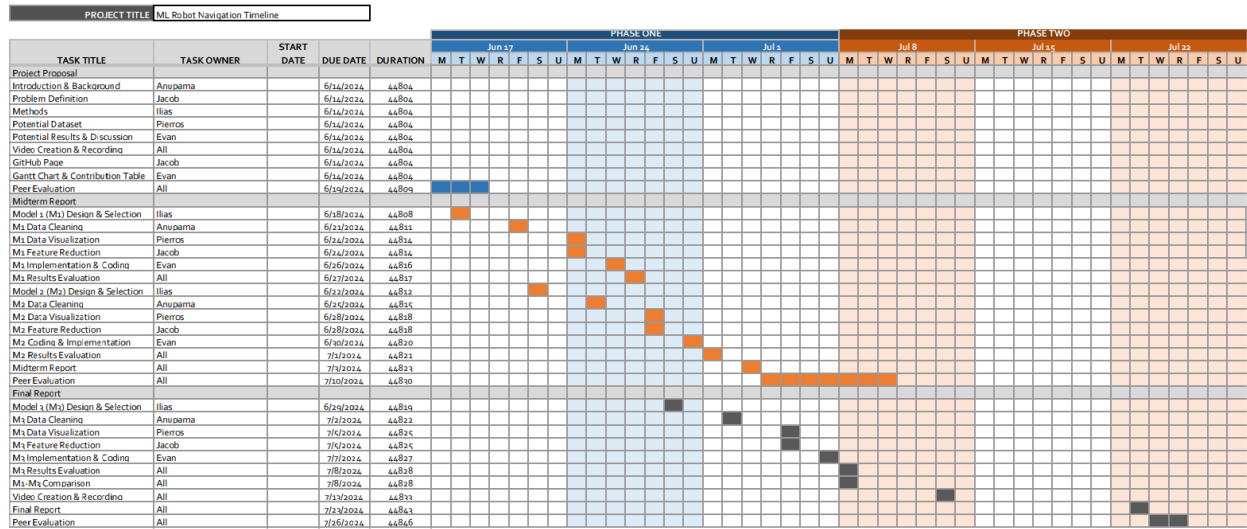


Figure 2: Team Gantt Chart

Table 1: Team Contributions

Name	Proposal Contributions
Jacob Blevins	Literature review, problem definition, and initial robot testing
Evan Boekweg	Metrics, goals, and results
Ilias Baali	ML algorithms and models
Anupama Nair	Data pre-processing
Pierros-Christos Skafidas	Dataset review

## References

- [1] Cann, D. O., 2016, “These are the top 10 emerging technologies of 2016,” *Media Relations World Economic Forum*.
- [2] Parekh, D., Poddar, N., Rajpurkar, A., Chahal, M., Kumar, N., Joshi, G. P., and Cho, W., 2022, “A Review on Autonomous Vehicles: Progress, Methods and Challenges,” *Electronics*, **11**(14).
- [3] 2024, “Tesla,” Accessed: June 11, 2024, <https://tesla.com>
- [4] 2024, “Waymo,” Accessed: June 11, 2024, <https://waymo.com>
- [5] Xia, W., Li, H., and Li, B., 2016, “A Control Strategy of Autonomous Vehicles Based on Deep Reinforcement Learning,” *2016 9th International Symposium on Computational Intelligence and Design (ISCID)*, Vol. 2, pp. 198–201, doi:10.1109/ISCID.2016.2054.
- [6] Ruan, X., Ren, D., Zhu, X., and Huang, J., 2019, “Mobile Robot Navigation based on Deep Reinforcement Learning,” *2019 Chinese Control And Decision Conference (CCDC)*, pp. 6174–6178, doi: 10.1109/CCDC.2019.8832393.
- [7] 2024, “LDS-02,” [https://emanual.robotis.com/docs/en/platform/turtlebot3/appendix\\_lds\\_02/](https://emanual.robotis.com/docs/en/platform/turtlebot3/appendix_lds_02/)
- [8] 2024, “Raspberry Pi Camera,” [https://emanual.robotis.com/docs/en/platform/turtlebot3/appendix\\_raspi\\_cam/](https://emanual.robotis.com/docs/en/platform/turtlebot3/appendix_raspi_cam/)
- [9] 2024, “Encoder,” <https://emanual.robotis.com/docs/en/platform/turtlebot3/appendixes/#more-info>
- [10] Zamanakos, G., Tsochatzidis, L., Amanatiadis, A., and Pratikakis, I., 2021, “A comprehensive survey of LIDAR-based 3D object detection methods with deep learning for autonomous driving,” *Computers & Graphics*, **99**, pp. 153–181.
- [11] Spielberg, N. A., Brown, M., Kapania, N. R., Kegelmann, J. C., and Gerdes, J. C., 2019, “Neural network vehicle models for high-performance automated driving,” *Science Robotics*, **4**(28), p. eaaw1975.
- [12] Scikit-learn Developers, 2024, “DBSCAN: Density-Based Spatial Clustering of Applications with Noise,” <https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html>, Accessed: 14-Jun-2024.
- [13] Wilson, D., 2024, “The Kalman Filter: An Example of Using 1-D Filters,” <https://www.kalmanfilter.net/kalman1d.html>, Accessed: 2024-06-14.
- [14] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A., 2016, “You Only Look Once: Unified, Real-Time Object Detection,” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, doi:10.1109/CVPR.2016.91.
- [15] Wang, X., 2018, “Autonomous Mobile Robot Visual SLAM Based on Improved CNN Method,” *IOP Conf. Ser.: Mater. Sci. Eng.*
- [16] Jin, X. and Wang, Z., 2022, “Proximal policy optimization based dynamic path planning algorithm for mobile robots,” *Electronics Letters*, **58**(1), pp. 13–15.