Navigation of Mobile Robot in the Pybullet Environment

***Abstract*: In this study, we explore the complexities of manipulating mobile robots in the Pybullet simulation environment. Our area of interest is algorithm creation and optimization for efficient robot mobility, obstacle avoidance, and path planning. We solve issues with dynamic surroundings and varied terrains by simulating and analyzing a range of scenarios using Pybullet's physics engine. This work offers useful methods and insights for improving mobile robots' overall navigational abilities through experimentation, with possible applications in autonomous systems and real-world situations.**

***Keywords: Navigation, Lidar sensor, Wheel and Steering control, Path Finding.***

# Introduction

## Background about Autonomous robots

The use of autonomous robots is expanding quickly across a range of industries, including personal support and industrial automation. They have a lot to offer in terms of effectiveness, security, and accuracy. Nonetheless, these robots still face significant challenges when it comes to navigating complicated surroundings, which calls for reliable and flexible algorithms that can manage a variety of scenarios. Numerous industries, including manufacturing, logistics, healthcare, space exploration, and defense, have found use for these robots. Concurrently, cooperative endeavors in swarm robots have investigated the possibility of numerous independent agents cooperating to complete intricate assignments, emulating natural behaviors.

## Procedure of the Project

In order to complete our project, we must simulate a mobile robot using Pybullet that must navigate obstacles in order to get to its destination. We design the robot and obstacles in URDF file format using 3D modeling. Run through the PyCharm editor in the Anaconda environment, our code illustrates the results of the simulation by showing the robot's path from the initial position to the destination. By fusing Pybullet functionality with 3D modeling, this integrated approach guarantees a technically accurate and visually compelling depiction of the mobile robot's simulated journey.

## Summary of the Project

Giving a mobile robot the capacity to navigate independently from its starting point to a desired destination is the main goal of the project. In order to provide safe and effective navigation in dynamic environments, this entails using sensor data for real-time obstacle detection and collision avoidance.

# Project Description And Results

## Procedure

With a focus on the Pybullet simulation environment, this project explores the intriguing field of mobile robot navigation. By giving our virtual robot the capacity to independently traverse its environment, we hope to enable it to arrive at its intended destination without running into any roadblocks. The lidar readings are continuously analyzed and interpreted by the robot's onboard algorithms, which process this data nonstop. The algorithms dynamically modify the wheel velocity and steering angle of the robot to adapt its movement based on the obstacles detected. The robot can maneuver around obstacles with accuracy and agility thanks to the complex interaction between sensor data and control algorithms. Still, it is insufficient to just stay clear of obstacles. We include an advanced pathfinding algorithm to ensure effective and optimal movement. The robot is guided from its starting point to its intended destination by this algorithm, which serves as a roadmap. Robots equipped with on-board algorithms quickly recalculate their trajectory in the event that an unforeseen obstacle suddenly appears on their planned path, ensuring a safe and effective detour around the obstacle. The robot's ability to adapt and react dynamically to unanticipated changes enables it to confidently and maneuver through even the most unpredictable environments.

## Project Algorithm

A proportionate control algorithm is used by the steering control mechanism. It computes the angle that separates the robot's intended orientation toward the target position from its current orientation. A seamless and managed direction change is ensured by the proportional term, which also sets the steering angle adjustment. The robot's adaptive wheel control keeps it moving at the intended speed even when it encounters obstacles or changes course to reach its destination. Using lidar readings, the algorithm adapts dynamically to obstacles. The steering angle is changed to avoid obstacles if they are detected within the designated range. The robot can automatically modify its path to avoid collisions while moving thanks to this dynamic obstacle avoidance mechanism.

Real-time observation and analysis are possible because the entire process is contained within a continuous simulation loop. By repeatedly going over the lidar scans, control choices, and simulation stages, this loop offers a thorough visual representation of the robot's navigational behavior inside the Pybullet environment. The control functions are

for wheel in wheels:

p.setJointMotorControl2(car,wheel,p.VELOCITY\_CONTROL, targetVelocity=targetVelocity, force=maxForce)

for steer in steering:

p.setJointMotorControl2(car,steer,p.POSITION\_CONTROL,targetPosition=steeringAngle)

The steering velocity will be decided based on the sensor readings and then will be passed on to the above control functions.

The target velocity we decided to keep it as a constant, which is fixed for its whole travel.

## Results

Screenshots of various locations of car

Image a

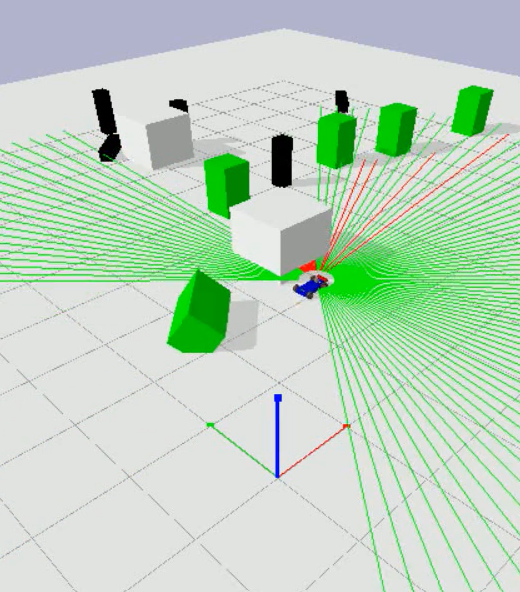


Image b

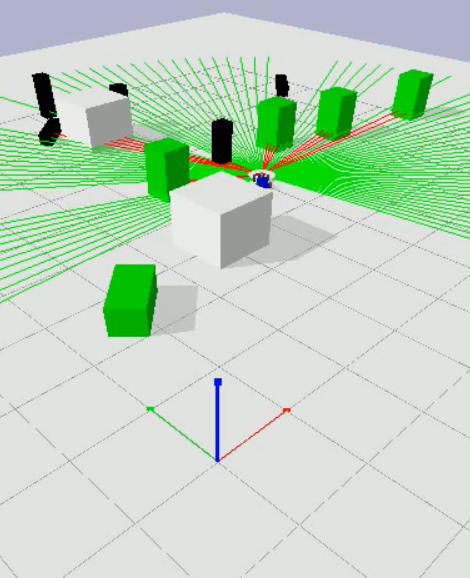


Image c

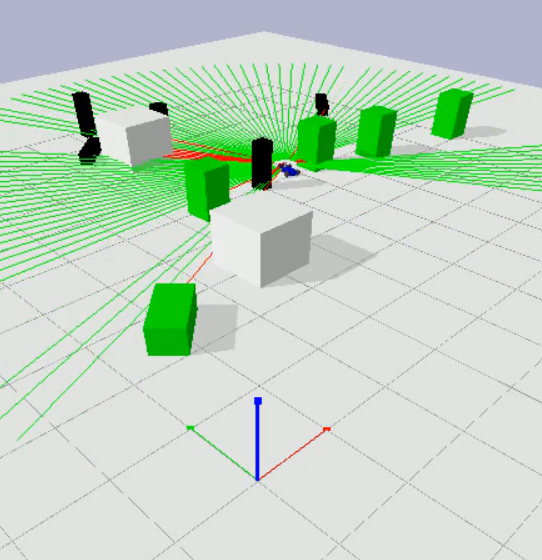
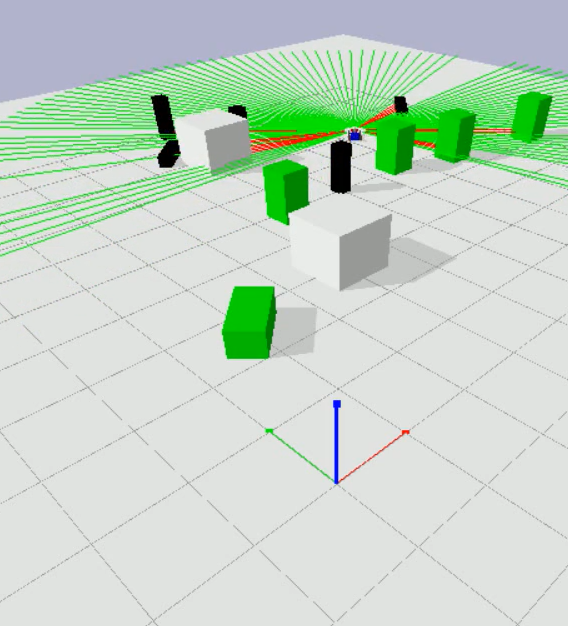


Image d



# Discussion of results

Robotic trajectory adjustments based on lidar scans and environmental conditions enable the algorithm to successfully guide the robot from its starting point to the desired destination. An effective method of avoiding obstacles is demonstrated by the algorithm. Through the use of 20Hz lidar reading analysis, the robot is able to continuously modify its steering angle in order to avoid obstacles in its path. The difference between the robot's current orientation and the intended orientation toward the target position determines how much the steering angle needs to be adjusted. By using this mechanism, the robot is guaranteed to navigate in an efficient and predictable manner towards the target. A realistic visual depiction of the robot's movements and interactions with the surroundings is provided by the rendering capabilities of the Pybullet simulation environment. By facilitating comprehension of the robot's actions, the visualization confirms the efficacy and reasonability of the applied algorithm. The system demonstrates the efficacy and usefulness of the implemented algorithm within the Pybullet simulation framework by being able to navigate through a dynamic environment on its own, avoiding obstacles and arriving at the desired position.

# Conclusion

To conclude, we have implemented an algorithm to find path while avoiding the obstacles, manipulate the steering angle to avoid collision, hence we are successfully able to navigate the mobile robot car from start location to destination location while avoiding all the obstacles in its path based on sensor readings.

# References

[1]. Quick Start Guide

[2]. Pybullet Documentation

[3]. Github.com

[4]. StackOverflow.org