ENPM808X Mid-Term Project Proposal

Darshit Desai
`darshit@umd.edu`

Shivam Sehgal
`ssehgal7@umd.edu`

Patrik Pordi
`ppordi@umd.edu`

October 17, 2023

Contents

0.1 Introduction

The three basic components for a functioning robotic system are perception, planning and controls. Based on the inputs from these 3 sub systems the robot's behavior is defined. In this project we have chosen the topic of controls and decided to design a software which acts as a module of a larger self-driving car software stack. The ackerman steering is the most widely used steering constraints in robots to large automobiles. To design a module which takes into account the steering constraints and designs a controller on top of it to achieve a goal pose of the vehicle queried by any other system of the robot like perception or navigation systems.

The proposal document is divided into two main sections, one is the problem statement and methodology, and the second one is the software development organization. The first section describes the methodology of the team and the underlying processes and the second section goes through the organization and the potential risks of the project.

0.2 Problem Statement and Methodology

0.2.1 Problem Statement

For a robot moving in a confined indoor space or an outdoor space the control module plays a critical role since the ability of the robot to perceive and respond to the robot's surroundings highly depends on the ability of accurately following the reference signal or achieving the target point as soon as possible. On top of that the robot also has some physical constraints like the ackerman steering constraint which limits it's ability to swiftly move from one position to other position discreetly and makes the robot to move in a more smoother trajectory, Taking all that into consideration we have defined a few equations of the ackerman steering model which we plan to use later during development.

0.2.2 Methodology

The section describes the use of various equations and the required controller for testing the algorithm,

The following are the notations relevant to the equations defined below:

- R - Turning radius
- α_i - Turning angle for the inner wheel
- α_o - Turning angle for the outer wheel

- L - Wheel base (distance between two axles)
- r_w - Wheel radius
- θ_t - Target heading of the robot
- v_t - Target velocity of the robot
- T - Track length (distance between wheels on an axle)

We can calculate the optimal steering angles for both the inner and outer wheels based on a specific turning radius:

$$\alpha_i = tan^{-1}(\frac{L}{R - \frac{T}{2}}) \tag{1}$$

$$\alpha_o = tan^{-1}(\frac{L}{R + \frac{T}{2}}) \tag{2}$$

The robot will travel in a circular path determined by the turning radius measured from the center of the rear axle. As a result, the inner and outer wheels will trace concentric arcs. The lengths of these arcs are as follows:

$$s_i = \left(R - \frac{T}{2}\right)\theta_t \tag{3}$$

$$s_o = \left(R + \frac{T}{2}\right)\theta_t \tag{4}$$

Assuming that the turn has been successfully executed, we can reset the wheels to their straight position and convert the entire rotational velocity into linear velocity. We then use the target velocity to calculate the time required to finish the turn.

$$v_t = \omega_{robot} R \tag{5}$$

$$\omega_{robot} = \frac{\theta_t}{t} \tag{6}$$

$$t = \frac{R\theta_t}{v_t} \tag{7}$$

We can calculate the rotational speed of each wheel by considering the arc's length, the wheel's radius, and the time it takes to traverse the arc.

$$\omega_i = \frac{S_i}{r_w \cdot t} \tag{8}$$

$$\omega_0 = \frac{S_0}{r_w \cdot t} \tag{9}$$

For the controller we have chosen a PID controller, which would take into account the robot's current state and the given velocity and time step to move the robot in the direction towards the goal.

0.3 Software Development Organization

0.3.1 Development Process

The project is divided into three weeks with each week having a different phase of the software development. The project aims to follow AIP or Agile Iterative process followed through three iterations of the project design throughout 3 weeks. During each week of development, there is a persistent emphasis on UML design and the overarching project design. The approaches for following the process would be to conduct deliberative sprint meetings and collaborate for version control using Git as well as practice pair programming to enhance robustness of individual iterations.

0.3.2 Software Technologies

For the software technologies to follow the above development process, the below tools will be used:

- Programming Languages: C++
- Development Tools: CMake, Git, Github Desktop GUI Cppcheck, Cpplint, Makefiles, VSCode, Emacs
- Testing tools: Valgrind, Google Test Suite, Test driven development process
- Documentation: Doxygen
- Continuous Integration and Code Coverage: GitHub CI, Github Actions, CodeCov

0.3.3 Team Organization

For phase 0, the following roles are assuming this roles:

1. Patrik Pordi - Navigator
2. Shivam Sehgal - Driver
3. Darshit Desai - Design Keeper

The team has decided to follow test driven development approach, As we proceed to the next phase the roles will be switched.

0.3.4 Potential Risks:

- Implementation Complexity: The implementation of the Ackerman steering control module, along with the PID controller and other components, may turn out to be more complex than initially anticipated.
- Visualisation Software: It is difficult to visualise an ackerman steering based robot with sufficient accuracy, which might lead to development and testing delays