ACKERMANN STEERING CONTROL

Maitreya Kulkarni
UID: 117506075

Mohammed Maaruf Vazifdar
UID: 117509717

Pratik Sunil Bhujbal
UID: 117555295

Project Overview - As per collective decision group MT15 will design and develop the Ackermann steering system. Acekrmann steering condition arises when axes of all wheels in a vehicle intersect at a single turning point. In the case of Ackermann steering the wheels of the vehicle do not skid while turning as opposed to parallel steering. This reduces the tyre wear and increases the energy efficiency. Ackermann steering can be used in various autonomous systems such as self driving cars, delivery robots which can increase the efficiency of the robot.

Ackermann steering assumes that the steering angle is calculated with respect to a common centre. In this case, rear wheels are considered to be the driving wheels without any steerage. The steering centre is assumed to be on the line extended from the rear axle.

During the development the following assumptions are made:

- All robot parameters are known.
- Initially the robot is on the origin of the world frame facing the x axis.
- Robot's desired heading and velocity are given by the user as input.
- Maximum steering angle is 45 degrees.
- Friction and wheel slippage are considered negligible.

Development Process - This project involves three programmers, following the Agile Iterative Process and programming in-pairs.

To ensure product quality roles of driver, navigator and design-keeper will be switched amongst the team regularly.
To prevent occurrence of bugs and failing of sub-modules the team will adapt to Test Driven Development philosophy.
All programming will be done using modern C++, following Google Style Sheets and abiding to the practices of OOPS.

Project Technologies -

- Programming Language - Modern C++
- Ubuntu 18.04
- Build system - Cmake
- Version control - GitHub
- Software Tools - VS Code
- Build Check - Travis
- Code coverage - Coverall
- Gazebo API - To interface with the robot model in the Gazebo environment.

Algorithm -

- Ackermann Steering Model Kinematics.
- Initial and Goal values of heading and velocities are provided by the user.
- Implementation of PID controller for heading correction.

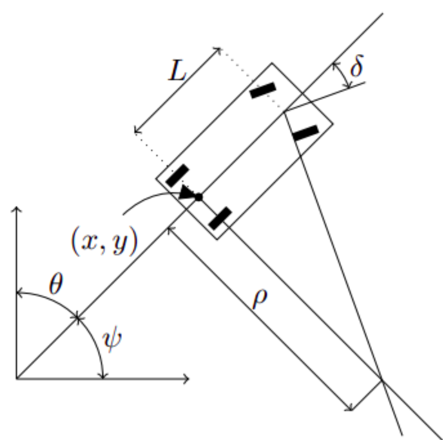

Fig1. Ackermann model

$$\dot{x} = v\cos(\psi) = v\cos\left(\frac{\pi}{2} - \theta\right) \quad (1a)$$

$$\dot{y} = v\sin(\psi) = v\sin\left(\frac{\pi}{2} - \theta\right) \quad (1b)$$

$$\dot{\theta} = \frac{v}{L}\tan(\delta). \quad (1c)$$

Fig2. Model Equation

Figure 1 shows the vehicle scheme and the main variables. The pose is defined by its position (x, y) and the bearing θ. The actuations are the car speed v and the steering angle δ. The only parameter is the wheelbase L. The model equations are shown in

Figure 2.

Fallback plan -
Implementation aims to demonstrate convergence to the setpoint. The plan is to simulate the project in Gazebo simulator using the Gazebo API. But the fallback plan will demonstrate without the simulation.

Deliverables -
A robust ackermann steering controller that once integrated will help steer the robot to the desired setpoints.

References -

- Kelly, Alonzo & Seegmiller, Neal. (2014). A Vector Algebra Formulation of Mobile Robot Velocity Kinematics. Springer Tracts in Advanced Robotics. 92. 613-627. 10.1007/978-3-642-40686-7_41.
- A. J. Weinstein and K. L. Moore, "Pose estimation of Ackerman steering vehicles for outdoors autonomous navigation," 2010 IEEE International Conference on Industrial Technology, 2010, pp. 579-584, doi: 10.1109/ICIT.2010.5472738.