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.

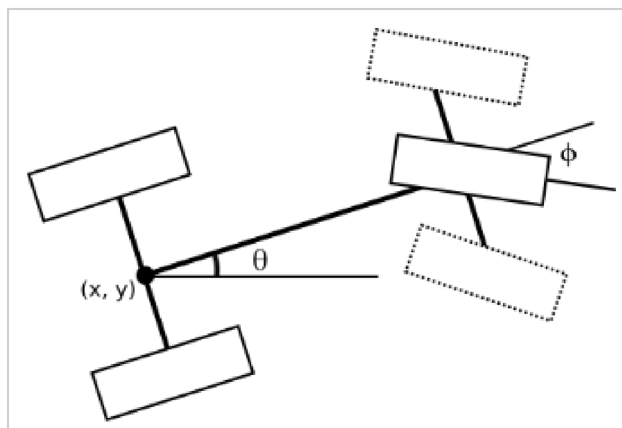


Fig1. Ackermann Forward Kinematics

$$\dot{x} = s\cos\theta$$
$$\dot{y} = s\sin\theta$$
$$\dot{\theta} = \frac{s}{l}\tan\phi \approx \frac{s}{l}\phi$$

Fig2. Forward Kinematics Equations

As the state of the vehicle is represented as a quadruple (x,y,θ,ϕ), with θ being the heading and (x,y) the position in the world frame with s as speed of vehicle gives equations shown in figure 2.

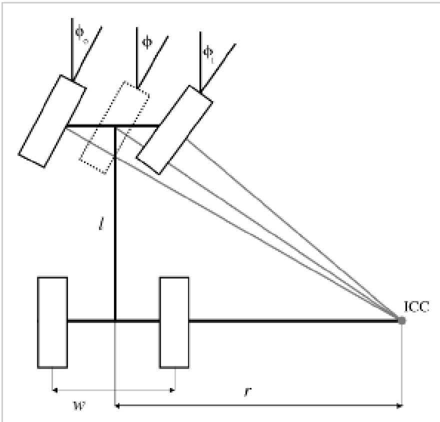


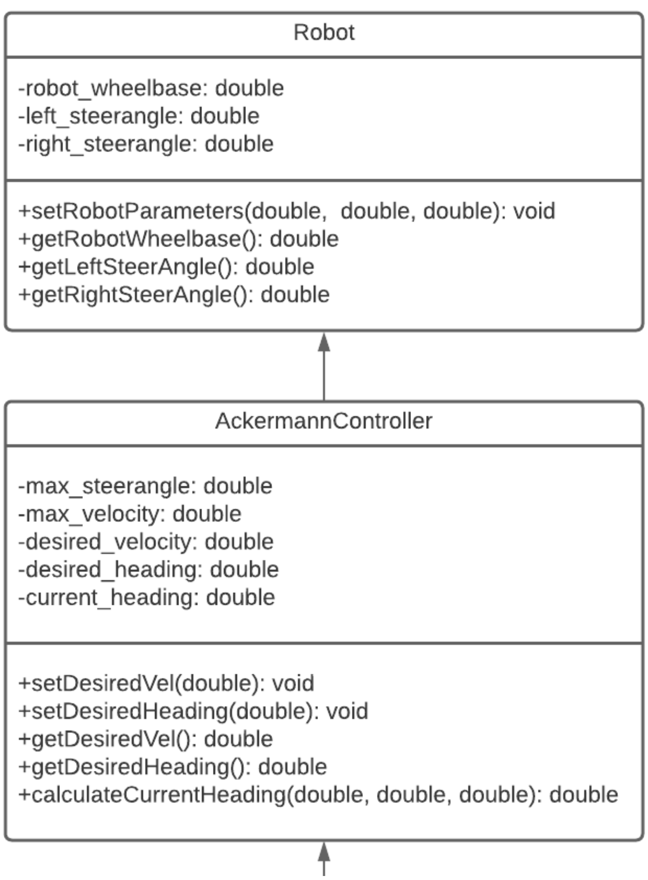
Fig3. Steering Model

$$\phi_i = \tan^{-1}\left(\frac{2l \sin\phi}{2l\cos\phi - w\sin\phi}\right)$$
$$\phi_o = \tan^{-1}\left(\frac{2l \sin\phi}{2l\cos\phi + w\sin\phi}\right)$$

Fig4. Steering Equations

With the track width *w* (the lateral wheel separation), the wheel base *l* (the longitudinal wheel separation), ϕi the relative steering angle of the inner wheel, ϕo the relative steering angle of the outer wheel and *r* the distance between (instantaneous center of curvature) and the center of the car we get equations for ϕi & ϕo as shown in figure 4 .

Class Structure -



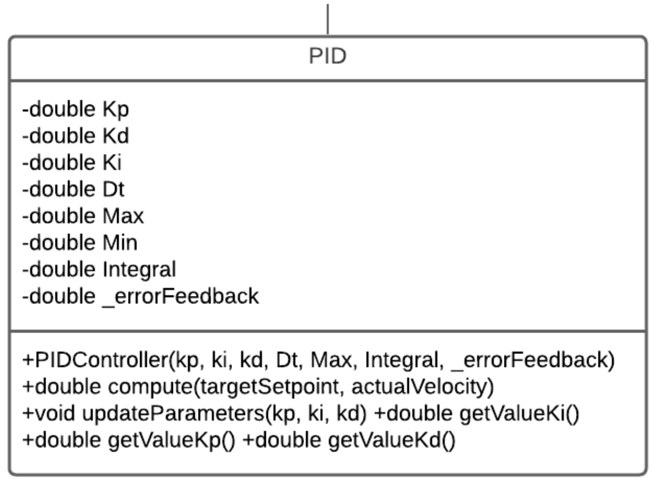


Fig5.UML Class Diagram

Figure 5 shows initial class structure for implementation.

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.
- https://www.xarg.org/book/kinematics/ackerman-steering/

Group 15 - Maitreya Kulkarni, Maaruf Vazifdar, Pratik Bhujbal
Controller for an Ackermann Kinematic Model with a maximum steering angle constraint of 45 degrees

Unique ID	Task	Target time (minutes)	Actual Time	Status
1	**Plan and Design**	**390**	**571**	
1.1	Research on Ackermann Steering	120	145	Completed
1.2	Research on Gazebo APIs	75	78	Completed
1.3	Creating a GitHub repo	15	13	Completed
1.4	Create Proposal Document	120	157	Completed
1.5	Creating Activity Diagram	45	48	Completed
1.6	Creating UML diagrams	30	33	Completed
1.7	Create stubs according to the UML diagrams	95	93	Completed
1.8	Create pull request for stubs	5	2	Completed
1.9	Create pull request for Quad chart and AIP work log	5	2	Completed

Group 15 - Maitreya Kulkarni, Maaruf Vazifdar, Pratik Bhujbal
Controller for an Ackermann Kinematic Model with a maximum steering angle constraint of 45 degrees
Work/ Time log

Process Enactment		Pair Programming Roles						Interupt		
Task	**Comment**	**Driver**	**Navigator**	**Design Keeper**	**Date**	**Start**	**End**	**#**	**Time**	**Clean time**
Iteration 1										
Research on Ackermann Steering	Online Resouces for Ackermann Steering and related Kinematics	Maaruf Vazifdar	Maitreya Kulkarni	Pratik Bhujbal	10/4/2021	22:20	0:45	N/A	N/A	145
Research on Gazebo APIs	Interfacing Gazebo with C++ scripts	Pratik Bhujbal	Maaruf Vazifdar	Maitreya Kulkarni	10/4/2021	22:20	23:38	N/A		78
Creating a GitHub repo	A GitHib repo with the name "Ackermann_Steering_Control" was created	Pratik Bhujbal	Maaruf Vazifdar	Maitreya Kulkarni	10/5/2021	23:45	23:58	N/A	N/A	13
Create Proposal Document	Mid-term project proposal document was created.	Maaruf Vazifdar	Pratik Bhujbal	Maitreya Kulkarni	10/5/2021	18:15	20:45			157
Creating Activity Diagram	Activity diagrams were created with the basic structure	Maitreya Kulkarni	Maitreya Kulkarni	Maaruf Vazifdar	10/5/2021	18:15	19:03			48
Creating UML diagrams	UML class and relationships were added between classes	Pratik Bhujbal	Maitreya Kulkarni	Maaruf Vazifdar	10/5/2021	18:15	18:48	4	13	33
Create stubs according to the UML diagrams	Stubs are created for the UML diagrams	Maaruf Vazifdar	Pratik Bhujbal	Maitreya Kulkarni	10/5/2021	22:12	23:45	N/A	N/A	93
Create pull request for stubs	Pull request for the repo was created and merged	Maaruf Vazifdar	Pratik Bhujbal	Maitreya Kulkarni	10/5/2021	23:45	23:47	N/A	N/A	2
Create pull request for Quad chart and AIP work log	Pull request for the repo was created and merged	Maitreya Kulkarni	Maaruf Vazifdar	Pratik Bhujbal	10/5/2021	23:45	23:47	N/A	N/A	2
										571