Acme Ackermann Steering Controller

Description

Ackermann steering is a mechanism used in wheeled vehicles to keep the wheels from slipping while turning. The basic principle of Ackermann steering is that as a vehicle rotates around a single point, the wheels must turn at different angles. Figure 1 shows how when turning right, the front right wheel will have to be angled more than the front left in order to keep the same center of turning. The opposite holds true when the vehicle has to turn to the left.

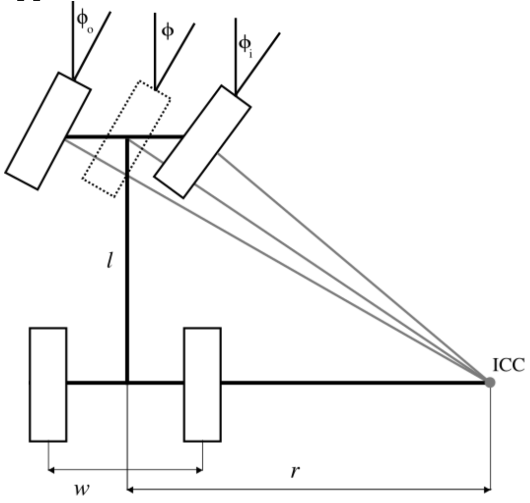

Figure 1. Ackermann Steering Geometry

In addition to different turn angles, each wheel must also rotate at different speeds. In figure 1, the front right wheel must travel a shorter distance compared to the front left wheel. This means the front left wheel must rotate faster in order to keep the vehicle on track.

Ackermann steering is especially important for autonomous vehicles as they perform better with little error. If the vehicle keeps slipping while it's turning, the controller has to be able to constantly correct itself in order to reach the desired target. Having reliable mobility is also critical to other on-board systems performing their roles accurately. For example, if a vehicle has an onboard manipulator and is designed to interact with objects while moving, any slipping may cause the manipulator to miss or damage its target. This project will be useful for any future Acme robot that has Ackermann steering.

Design and Development Process

For this project, we will be using an agile iterative process to develop this software. A product backlog will be generated from the given requirements. Every day we will meet to discuss the tasks on hand and any issues either of the developers have. During these meetings, each developer will be assigned specific tasks to complete in the software. All development will be implemented using parallel programming where both developers will be working on the software at the same time. Conflicts will be minimized and addressed using Github version control where each programmer will work in a separate branch and push completed software to the master branch. After each iteration, both members will meet again to discuss the software changes.

The project will be built on a Linux Ubuntu 18.04 system and will be programmed in c++. Open source build tools such Travis-CI will be utilized to keep the track of build of updated programs. Coveralls will be used to verify the coverage and Valgrind will be used to check for memory leaks and undesired behaviors. To get the precise tuning and calculations for our controller we will be using cmath (<cmath> (math.h)) which is an inbuilt library of c++ . To help visualize the functionality of our software, we will be using open source libraries like open cv for data processing to see if the vehicle is following the expected trajectory. If time permits, we might try to visualize the model in SDL (Simple DirectMedia Layer) which is a commonly used visualization tool for game development.

Algorithms

The Ackermann Steering model uses the following equations to calculate the steering angle for both the inner and outer wheels:

$$\phi_i = tan^{-1}\left(\frac{2lsin(\phi)}{2lcos(\phi) - wsin(\phi)}\right) \phi_o = tan^{-1}\left(\frac{2lsin(\phi)}{2lcos(\phi) + wsin(\phi)}\right)$$

where $\phi = tan^{-1}(\frac{l}{r})$, w is the track width, l is the wheelbase, and r is the desired turn radius.

The forward kinematics will also be implemented in order to find the vehicle's heading and the wheel speeds: $\frac{d\Theta}{dt} = \frac{v}{l}tan(\varphi)$, $v_i = \frac{d\Theta}{dt}\left(\frac{r-\frac{w}{2}}{sin^{-1}(\phi_i)}\right)$ $v_o = \frac{d\Theta}{dt}\left(\frac{r+\frac{w}{2}}{sin^{-1}(\phi_o)}\right)$

Risks

This project will be using openCV as a visualizer due to its vast amount of resources. While openCV is a widely used tool, neither partner has much experience using it. There are always risks associated with using new software but we believe the knowledge base is large enough for us to successfully implement it into our project.

Deliverables

The final deliverables will be a software that can be adapted to any robot using Ackermann steering. By altering the geometries associated with each robot such as the wheelbase, wheel radius, and track width, the software will be able to output the steering angles and velocities of each wheel in order to achieve a desired vehicle heading and velocity. The visualizer should give roboticists the ability to simulate their robot missions before the real-world implementation.

Pair Programming

In order to maintain high quality code, we will be implementing pair programming techniques. The two developers, Loic Barret and Divyam Garg, will be switching between the roles of driver and navigator. The driver will be the person programming while the navigator reviews the driver's code while they're writing it. Not only does this lead to well written code, it also frees up the driver to focus on the technical aspects of the code while the navigator can focus on the strategy of the code. Due to the coronavirus pandemic, all pair programming will be done virtually.

References

Eisele, Robert. "Ackerman Steering." *Ackerman Steering • Open Source Is Everything*, www.xarg.org/book/kinematics/ackerman-steering/.

"Pair Programming." *Wikipedia*, Wikimedia Foundation, 9 Sept. 2020, en.wikipedia.org/wiki/Pair_programming.