ENPM808X - Software Development for Robotics

Midterm - Phase 0 (Proposal)

Submitted by: Karan Sutradhar (117037672) & Ajinkya Parwekar (117030389)

Motivation:

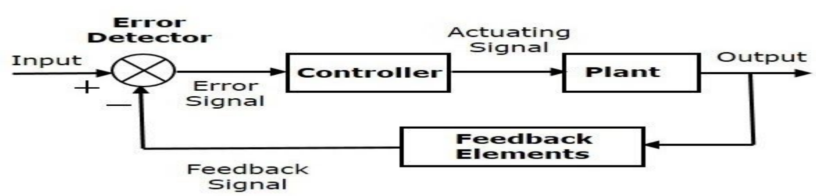
The recent developments in the field of robotics have gone from science fiction fantasy to world-bound reality. Robots are consistently substituting humans to do mundane, hazardous, repetitive tasks with higher precision. However, we understood with our experiences, research background that the applications of Robots are much broader than the aforementioned reasons. Yearn for constant learning and continuous advancement in the field of Autonomous Driving drove us to choose the Ackerman PID (proportional–integral–derivative) controller. The interest in autonomous systems and control theory motivated us to opt for the project of the Ackerman steering control system.

Scope and Design:

We intend to implement a PID based controller for an Ackermann steering system (kinematic model) which works on a feedback loop. It has a maximum steering angle constraint. The inputs are robot target heading and velocity, output steering angle and the two drive wheel velocities, demonstrating the convergence of the actual components to the set points.

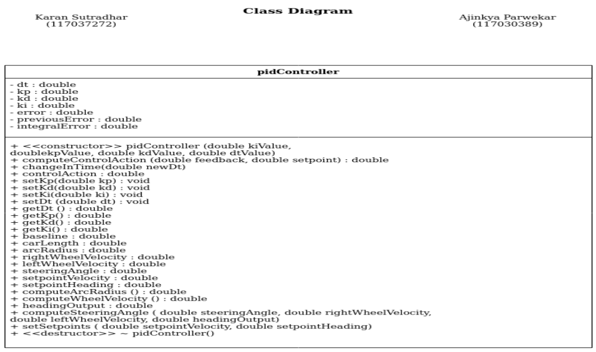
The proposed work will implement a design of a PID controller working on a feedback mechanism which will compute the error between the set point variable and the actual variables.

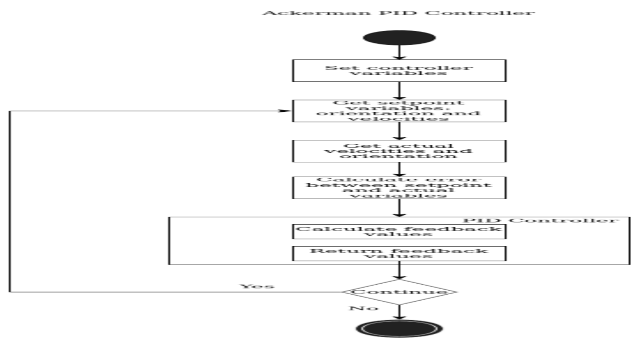
A control system is a system, which provides the desired response by controlling the output. In an open loop system, the input acts on a plant for the system to be controlled and then some output signal is generated. Often these kinds of system performance is not good enough to meet the requirements. Open loop systems are good when there are no variations in components like speed, steering angle etc. Thus, we consider a closed loop system in which the output from the controller drives the vehicle and it works on a feedback mechanism.



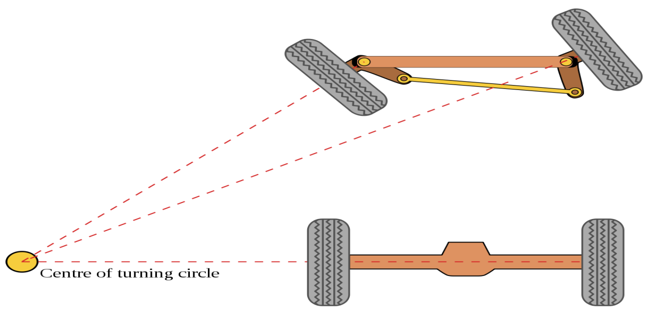
source : :https://www.tutorialspoint.com/control_systems/control_systems_introduction.htm

Our project is implemented with C++ codes, we are considering a single class pidController. There are various inputs of proportional–integral–derivative components and we intend to find the steering angle using the feedback loop of velocity, heading and orientation. Also the rate of turning depends on the two variable velocity components i.e., the right wheel velocity and left wheel velocity.





Ackerman Steering Mechanism:



(image source: https://en.wikipedia.org/)

When a vehicle is turning, the inner front wheel needs to turn at a different angle to the outer because they are turning on different radii. The Ackermann steering mechanism is a geometric arrangement of linkages in the steering of a vehicle designed to turn the inner and outer wheels at the appropriate angles. The intention of Ackermann geometry is to avoid the need for tires to slip sideways when following the path around a curve. The geometrical solution to this is for all wheels to have their axles arranged as radii of circles with a common centre point. As the rear wheels are fixed, this centre point must be on a line extended from the rear axle. Intersecting the axes of the front wheels on this line as well requires that the inside front wheel be turned, when steering, through a greater angle than the outside wheel.

Pair Programming:

Ajinkya (Driver) and Karan (Navigator)

Risk Involved:

The risk involved is overshooting. It can be solved by obtaining an open-loop response and determining what needs to be improved. Adding a proportional control to improve the rise time. Adding a derivative control to reduce the overshoot. Adding an integral control to reduce the steady-state error. Adjusting each of the gains.

Method and Tools:

Programming is done C++. We plan to simulate the system and environment on ROS.

References:

https://en.wikipedia.org/wiki/Ackermann_steering_geometry
https://www.maplesoft.com/products/maplesim/ModelGallery/detail.aspx?id=56
http://ctms.engin.umich.edu/CTMS/index.php?example=Introduction&section=ControlPID