AI Automation of Vehicle Cruise Control System Design

Goals

Goal 1: Implement working cruise control system for vehicle that can:

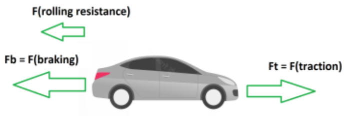
- smoothly transport passengers
- React safely to worst-case braking scenarios of lead vehicles

Goal 2:

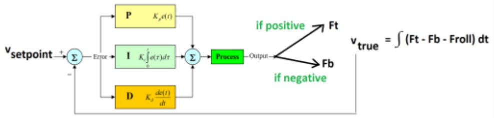
Implement ML-based methods to automate various testing and parameter-tuning phases of the control system design process

Methods

Forces on Car

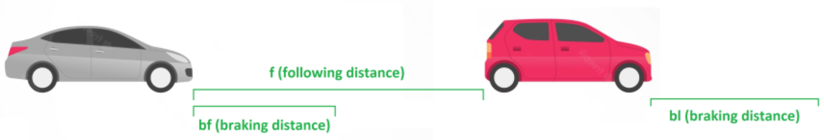


PID Controller



$v_{true} = \int (Ft - Fb - Froll)\, dt$

Cruise Control Mode Switching



mode = f - bf + bl

CASE	CONTROL MODE
if (v(follower) < 4m/s and v(lead) < 4m/s)	crawling
else if (mode > 0)	reference speed tracking
else if (mode < 0)	safe tailing

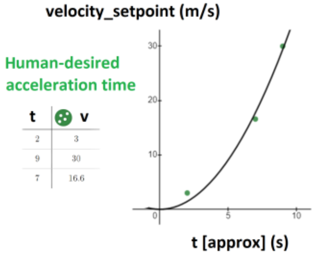
Testing

Test Cases

- we tested our system to handle these actions of the lead vehicle:
  - Immediate braking
  - Gradual braking
  - Unsteady braking
  - etc.

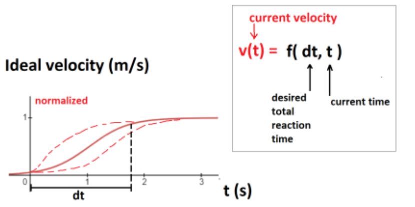
AI Automation

Defining Idealized Behavior (Acceleration)



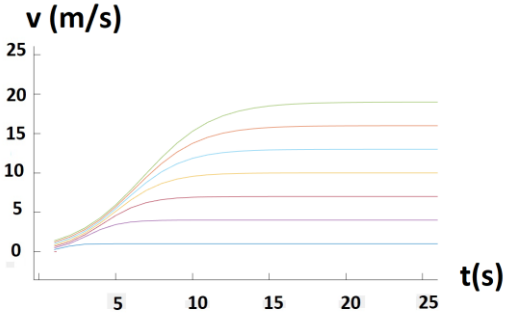
t	v
2	3
9	30
7	16.6

Converting desired reaction time into a simple formula

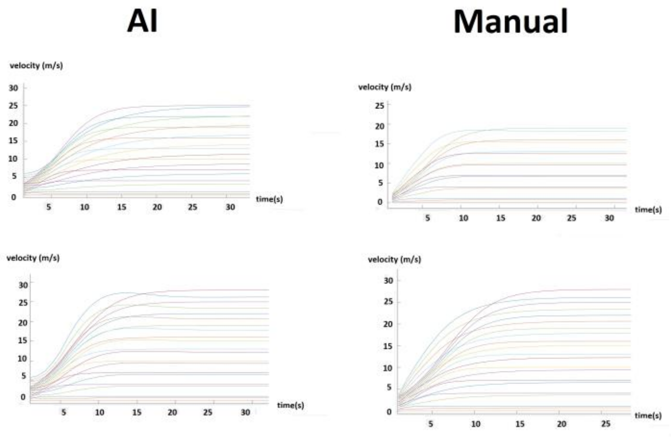


$v(t) = f(dt, t)$

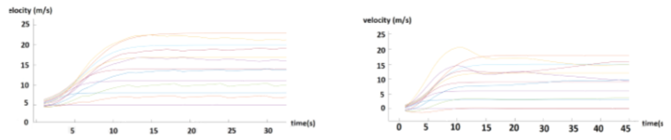
Generating Testcases



Results



Testing on Graded Roads



Summary of Findings

We first developed a cruise control system that used three PID controllers to tail lead vehicles at an optimally close, yet safe, following range. We tested our system's safety in hundreds of test cases, with different lead vehicles and different behavioral patterns of the lead vehicle. After tuning our system for perfection, we decided to build an AI to automate the laborious design process.

Our AI helped us find ideal parameters for our reference speed-tracking PID controller, so that it reacted smoothly to error signals and even helped us tune the system to maintain steady velocities on graded roads. The AI was immensely useful as we tested more and more components.