DESIGN AND SIMULATION OF INTEGRATED LATERAL AND LONGITUDINAL CONTROLLER FOR AUTOMATED AND CONNECTED VEHICLES

A Thesis

by

RANGEESH VENKATESAN

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Swaroop Darbha
Committee Members,	Sivakumar Rathinam
	Dileep Kalathil
Head of Department,	Andreas Polycarpou

August 2021

Major Subject: Mechanical Engineering

ABSTRACT

This research is concerned with the design of an integrated longitudinal and lateral controller for a platoon of connected and automated vehicles executing an emergency lane change maneuver. The efficacy of the designed controller is shown using a simulation setup that has been built using an open-source simulator, called TORCS, in conjunction with Simulink, where the controller is designed. The longitudinal controller component consists of an Adaptive Cruise Control (ACC) System that employs a Constant Time Headway (CTH) policy. The lateral controller is based on the lead vehicle in a platoon broadcasting its GPS coordinates to all the following vehicles at regular intervals of time. This information is utilized by every following vehicle to construct a desired trajectory to track (in particular, it is a circular spline). Since the same information is used by every following vehicle, the issue of propagation of errors in the lateral direction is circumvented.

Stability robustness to variation in mass and inertia of the vehicle is crucial for ensuring that the controller can be implemented in practice. Vehicle mass and moment of inertia are most affected by the addition on passengers and cargo. The complicating part of studying the robustness stems from the characteristic polynomial being a rational function of the vehicle mass and is not readily amenable to analysis by the well-known root-locus method. A D-decomposition based method was adopt to find the maximum speed for safely/stably tracking a given reference trajectory.

DEDICATION

To my parents, for believing in me.

ACKNOWLEDGMENTS

I am extremely grateful to Professor Swaroop Darbha for his guidance and support throughout my three years at Texas A&M. He has been a great influence in making me realise a strong foundation of knowledge is crucial towards any work. I would also like to thank him for being patient and letting me explore different avenues in autonomous driving, which has let me develop my skills significantly. Being a TA for three semesters has improved my interpersonal skills greatly, and as a consequence, my confidence in presentations as well. I would also like to thank him for making my stay as financially secure as possible.

I thank Professor Sivakumar Rathinam and Professor Dileep Kalathil for guiding me well during their courses and for their thesis reviews, and Professor Swami for his opportunities that has helped me gain a strong foothold as an Autonomous Vehicles engineer.

I am thankful to my parents, Mr.Venkatesan and Mrs.Rani for their love and encouragement throughout my whole journey.

I am grateful to all my friends and roommates at Texas A&M for helping me have a great time here and enabling me to grow, both academically and character-wise. Special mentions to my roommates Jaikrishna and Neelkamal who joined me in working with Professor Swaroop in the same lab. Taking the same courses throughout our years in Master's was reminiscent of our undergrad years. I would also like to thank Kiran Adhithya and Mohamed Naveed for helping me immensely during crucial times in various stages of my graduate years.

My lab mates at the Systems, Controls and Optimization Lab Mengke Liu and Vamsi Vegamoor have contributed a lot towards the completion of my thesis and I'm really grateful for that.

Finally, I would like to thank Professor Chandramouli from IITM for helping me start my Master's journey at Texas A&M

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a thesis committee consisting of Professor Swaroop Darbha [advisor], Professor Sivakumar Rathinam of the Department of Mechanical Engineering and Professor Dileep Kalathil of the Department of Electrical & and Computer Engeineering.

Funding Sources

Graduate study was supported by a fellowship from Texas A&M University.

NOMENCLATURE

CAV	Connected and Automated Vehicles
EOM	Equations of Motion
ELC	Emergency Lane Change
GPS	Global Positioning system
IMU	Inertial Measurement Unit
SSD	Stand Still Distance
x_i	Position of i^{th} vehicle in platoon
h	Time headway
L	Length pf car in platoon
e_i	Spacing error of i^{th} vehicle
m	Mass
I_z	Moment of Inertia about the Center of Mass
C_f	Front Tire Cornering Stiffness
C_r	Rear Tire Cornering Stiffness
α_f	Front Tire Sideslip Angle
α_r	Rear Tire Sideslip Angle
δ_c, δ_f	Steering Command input, output
J_ω	Steering Wheel Inertia
b_ω	Torsional viscous damping coefficient
K_r	Torsional Stiffness of the steering column

v_x, v_y	Longitudinal and Lateral Velocity
θ	Vehicle Heading wrt ground frame
e_{lat}	Lateral Error of the Vehicle from the trajectory
$\tilde{\theta}$	Yaw Error of the vehicle
$\dot{\tilde{\theta}}$	Yaw Rate Error of the vehicle
K_{sg}	Steering Gradient of the vehicle
W_f, W_r	Front and Rear axle loads
k_e, k_θ, k_ω	Gains for lateral error, yaw error and yaw rate error
$\mathbf{K}$	Gain set
m_p	Total additional mass
r_p	Radius of gyration of additional mass

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION AND LITERATURE REVIEW

1.1 Introduction

This thesis deals with the simulation of a combined lateral and longitudinal control for Emergency Lane Change (ELC) maneuvers [1] for a convoy of Autonomous and Connected Vehicles (ACVs). The concept of a platoon is to have a coordinated movement of a group of vehicles, with a lead vehicle, which can be autonomous or human-controller, and the rest being automated follower vehicles. Designing a decentralized robust controller for the vehicles is crucial in realising this, as it plays a direct role in ensuring the safety of the platoon. A simulation of this is created using TORCS (Figure 1.3), an open source simulator, to corroborate the stability of the controller by showing the prevention of amplification of the cross track errors as one traverses down the string.

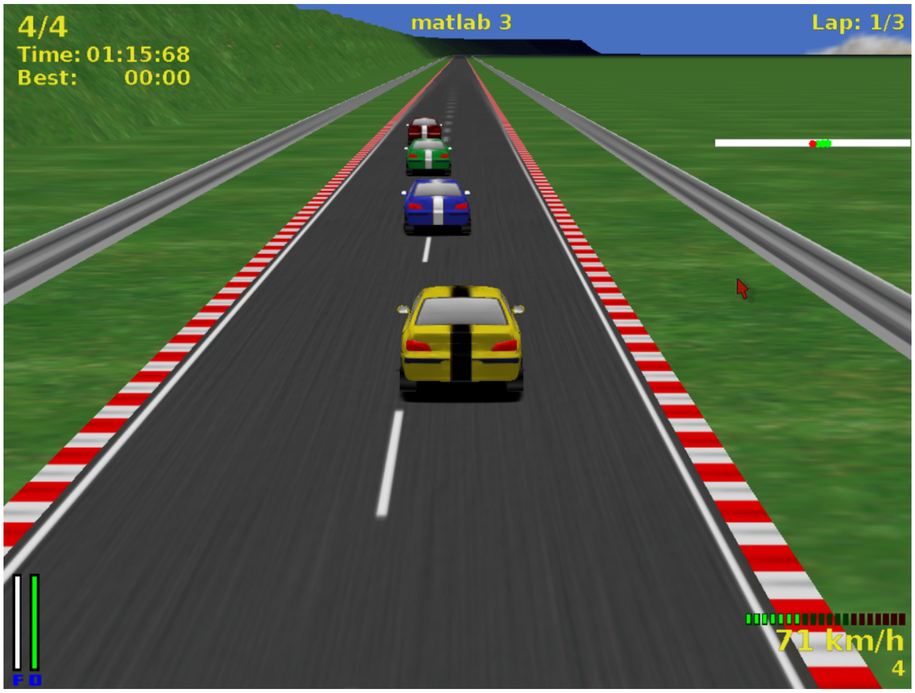


Figure 1.1: Emergency Lane Change Maneuver

1.2 Background and Motivation

1.2.1 Autonomous and Connected Vehicles

Vehicle platooning is an important innovation in the automotive industry that aims at improving the safety, mileage, efficiency and the time needed to travel. Autonomous capable vehicles in tightly spaced platoons will lead to savings in fuel, increased highway capacity and increased passenger comfort[2] . Inter-vehicular communications will enable the vehicles to reduce the desired spacing between the vehicles and therefore, improve the efficiency of the platoon[3].

1.2.2 Platoon Control Problem

The aim is to design an integrated lateral and longitudinal decentralized controller for the platoon that provides stability to the string of vehicles in the platoon. In the seminal paper [4], the definition for individual stability and string stability in the longitudinal direction is defined. Taking these definitions and extending them to the lateral direction, the stability definitions have been modified accordingly[5], [6]. Design of a longitudinal control law with a constant time headway policy has been discussed in [4], [7]. Lateral control with limited time-preview information has been discussed here [8], [9]. The goal of this thesis is to design a controller that is both string stable in the lateral and longitudinal directions and corroborate the same with a simulation. The proposed work differs from a similar work using Sliding Mode Control [10] is [11] in the structure of controller employed and the corroboration of controller efficacy for multiple following vehicles. The maneuvers we intend to simulate with are lane change maneuvers in a straight and curved road.

1.2.3 TORCS Simulation

The Open Racing Car Simulator (TORCS) [12], [13], is a state-of-the-art open source car racing simulator which provides a sophisticated physics engine, full 3D visualization, several tracks and models of cars. The car dynamics is accurately simulated and the physics engine takes into account many aspects of racing cars such as traction, aerodynamics, fuel consumption, etc. Each car is controlled by an automated driver robot. At each control step, a bot can access the current game

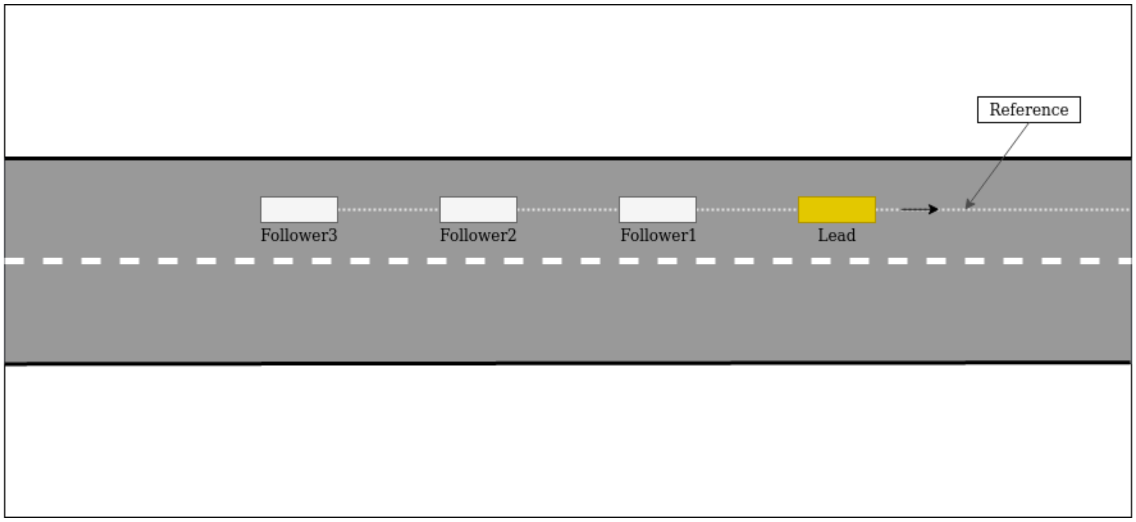


Figure 1.2: Vehicle Platoon

state, which includes several information about the car and the track as well the information about the other cars on the track, and can control the car using the gas/brake pedals, the gear stick, and steering wheel [14]. The game distribution includes several programmed bots which can be easily customized or extended to build new bots.

TORCS has been used in the research community to design controllers for the bots, for quite some time. It's versatile nature makes it easy to use learning techniques to design controllers [15], [16]. With TORCSLink [17], it's functionality has been expanded to include Simulink models to quickly design controllers for Autonomous and Connected Vehicles.

1.3 Goal and Objectives

The fundamental goal of this thesis is to design and simulate an integrated lateral and longitudinal controller for ACVs. This will be achieved by the following steps:

1. Design a control law for longitudinal and lateral control for a follower vehicle that has its position, velocity and acceleration information of all its preceding vehicles,

2. examine the robustness of the controller by analyzing its individual and string stability to a change in the car parameters due to variations in the number of passengers and cargo, and

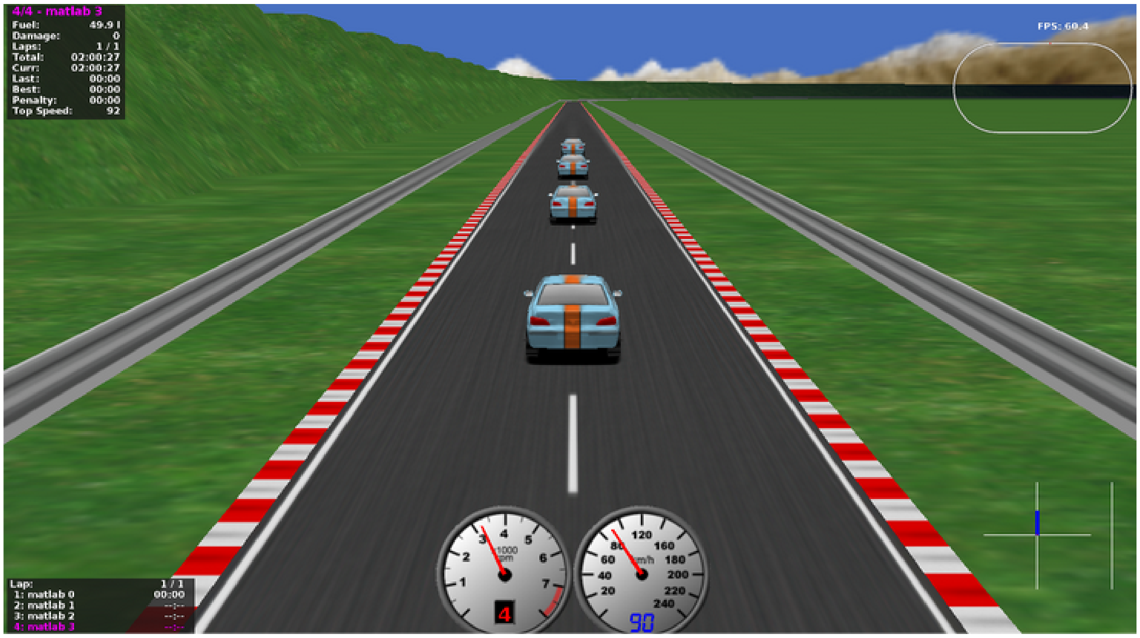


Figure 1.3: TORCS Simulation

3. simulate quick lane change maneuvers at low and high speeds and corroborate string stability in the lateral and longitudinal directions.

1.4 Thesis Outline

The following is a brief description of how the reminder of the thesis is structured.

In Chapter 2, the controller architecture is presented. The vehicle models for each controller, the string stability in the longitudinal direction and the individual stability in the lateral direction are summarised as well. Finally we discuss about the different V2V communication architectures possible and show via simulation later on which one is stable and why.

In Chapter 3, we talk about the simulation study in more detail. In particular, we dive deep into why we chose TORCS as our simulator, what its different features are and how it is modified to serve our purpose of proving the stability of a platoon with different communication architectures. Finally, we provide the results of the simulation runs with different communication architectures and show the stability.

In Chapter 4, the robustness study is described. The objective of this chapter is to capture the different variations that might occur during a real run of the controller in the practical world and how the variations in the parameters would still not be a cause of concern and stability is still guaranteed. In particular, we deal with the scenario of varying mass and moment of inertia and show that the controller is completely stable for a given range of values deemed possible in a generic run.

In Chapter 5, conclusions from this study are presented, and suggestions for future work are given.

2. CONTROLLER ARCHITECTURE

The controller design comprises of designing a longitudinal controller and a lateral controller. For the longitudinal controller, an adaptive cruise control algorithm with a Constant Time Headway Policy is chosen. The reason for this choice of policy is that it is shown that choosing such an algorithm would enable the spacing errors to not increase as we move down the vehicle platoon without any communication requirements [4].

The lateral controller design is that of a fixed structure feedback controller with a feed forward component. The gains corresponding to the error states in the feedback structure are chosen using the D-Decomposition approach so that the whole system is stabilized.

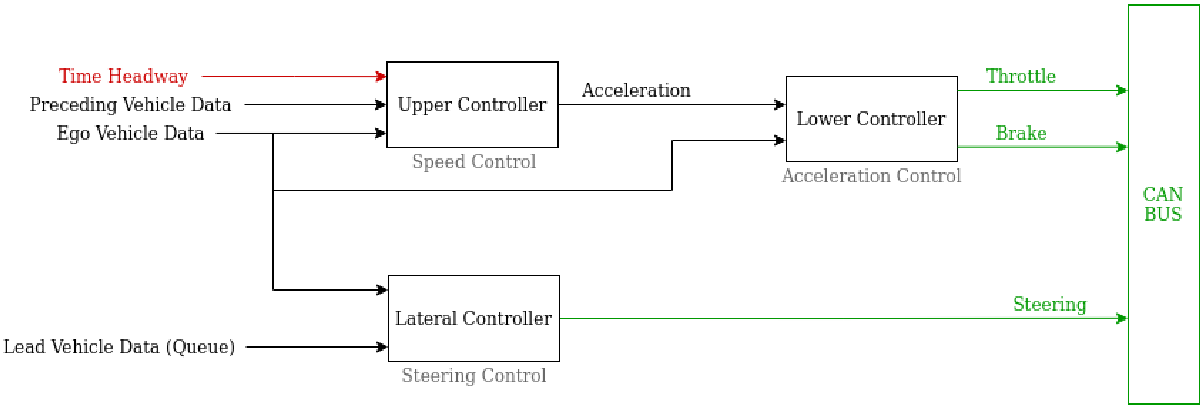


Figure 2.1: Controller Architecture

In the end, we present two V2V communication architectures that are considered for the vehicle platoon and discuss more about their viability in the next chapter

2.1 Longitudinal Controller

It is required that vehicles in a platoon maintain a desired following distance while guaranteeing string stability. The desired spacing is based on a spacing policy and a controller is designed to ensure the following distance of the ego vehicle is maintained based on state information from other vehicles in the platoon. Most often, the vehicle information needed is the predecessor's relative position and velocity which can be obtained most often from on-board sensory instruments or through other means such as V2X communication. If communication is possible and reliable, tighter vehicle spacing can be achieved with communicating the lead vehicle's information to the following vehicles.

In this section, we introduce the design of the string-stable longitudinal controller that is used to control the longitudinal motion of the follower vehicles.

Architecture Design

The longitudinal controller is composed of two parts - the upper level controller, that gives the desired acceleration to be tracked by the vehicle and the lower level controller that gives the throttle and brake commands as an output given the desired and current acceleration (Figure 2.1).

2.1.1 Upper Level Controller

The control algorithm used to design the upper level controller is an Adaptive Cruise Control Algorithm with a constant time headway policy (CTHP) for spacing.

Elaborating on the spacing policies, there are 2 types of spacing policies most often discussed in literature

- CTHP - Constant Time Headway Policy: The idea is to maintain the inter-vehicular spacing proportional to the speed of the ego vehicle. Such a policy has been shown to be string stable.
- CSP - Constant Spacing Policy: The idea is to maintain a constant inter-vehicular spacing distance. it has been shown that unless the lead vehicle's acceleration information is available, the system is string unstable. [18]

Since the constant time headway policy has been shown to be string stable with just the preceding vehicle's info, it has been chosen for the longitudinal controller design.

2.1.1.1 Vehicle Model

The first step in designing the control law is to identify the model that identifies the actual vehicle dynamics of the vehicle. The following nominal vehicle dynamics model is taken into consideration, where each vehicle is considered to be represented as a point mass whose position satisfies the double integrator dynamics:

$$\ddot{x}_i = u_i$$

where x_i is the position of the i^{th} vehicle in the platoon and u_i is the control input to the same vehicle.

The above vehicle model has been has been a popular choice for further stability analysis in the past. However, a more realistic model i needed to take into account the parasitic delays (discussed in further detail in [19]). The lower level controller have physical limits on what commands they can provide for and to compensate for that, we add a first order delay to the upper level controller. [11] Hence, the first order system with a time constant τ is defined

$$\tau \dddot{x}_i + \ddot{x}_i = u_i$$

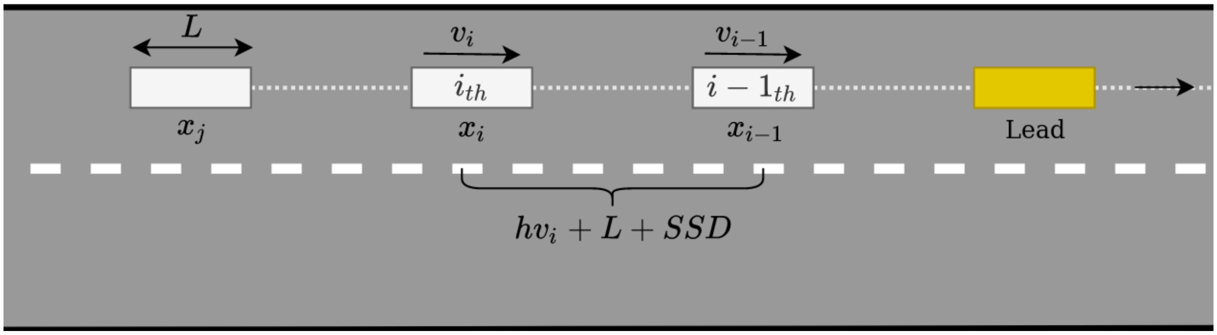


Figure 2.2: Adaptive Cruise Control

Figure 2.2 shows the definitions related to Adaptive Cruise Control. Let us assume there are n vehicles in the platoon and x_i denotes the position of the i^{th} vehicle. L is the length of the vehicle in consideration (The platoon is assumed to be homogeneous) and SSD is the stand still distance, which is the desired inter-vehicular spacing at zero velocity. h is the time headway, which is used to determine the desired spacing as a linear function of the ego velocity.

The spacing between the ego and preceding vehicle is defined as the distance between the rear end of the preceding vehicle to the front of the ego vehicle. Going by this definition, the desired inter-vehicular spacing is $SSD + hv_i$ and the current inter-vehicular spacing is $x_{i-1} - x_i - L$. Therefore, the spacing error is

$$e_i = SSD + L + x_i - x_{i-1} + hv_i$$

2.1.1.2 *Control Law*

Based on the Constant Time Headway Policy, the desired acceleration or the control input u_i is defined as

$$u_i = -k_v(\dot{x}_i - \dot{x}_{i-1}) - k_p(x_i - x_{i-1} + SSD + L + hv_i)$$

The next step is to identify the gains that would be able to stabilize the system. Taking the nominal longitudinal model as the plant model,

$$\ddot{x}_i = -k_v(\dot{x}_i - \dot{x}_{i-1}) - k_p(x_i - x_{i-1} + SSD + L + hv_i) \tag{2.1}$$

2.1.1.3 *Stabilizing Gains*

To ensure string stability of the platoon, we can mathematically state the objective as follows. Let $e_i = x_i - x_{i-1}$and let $H(s) = \frac{E_i(s)}{E_{i-1}(s)}$ be the error propagation transfer function.

For string stability, the following two conditions need to be satisfied:

1. $||H(j\omega)||_\infty < 1$

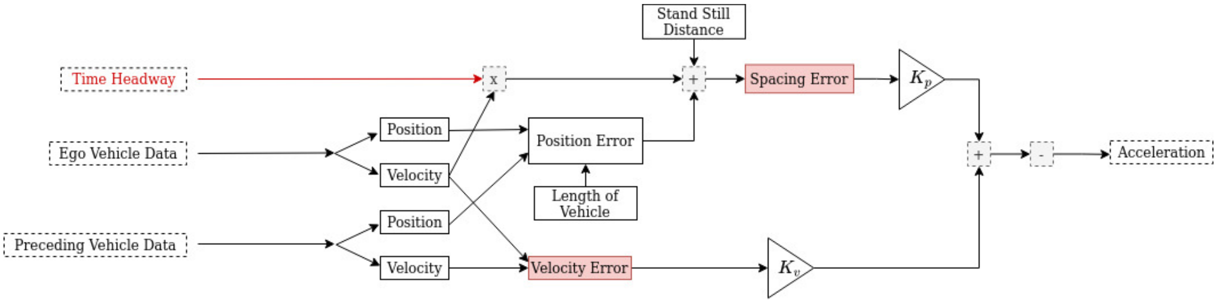


Figure 2.3: Adaptive Cruise Control

2. $h(t) > 0 \; \forall \; t > 0$

The idea behind the first constraint is that when the ∞ norm of the error propagation transfer function goes to 0, then the spacing errors will eventually die down and the string of vehicles will be stable.

The second constraint (mentioned in [4]) requires the impulse response function $h(t)$ corresponding to $\hat{H}(s)$ does not change sign. The idea behind this constraint is that the spacing errors are all either positive (The spacing is more than what is desired) or negative (The spacing is lower than what is expected). The reason being that if consecutive spacing errors are of the opposite sign, there is a chance of collision.

$$\begin{aligned} \ddot{e}_i(s) &= \ddot{x}_i(s) - \ddot{x}_{i-1}(s) \\ &= -K_v\dot{e}_i - K_p e_i - K_p d - K_p h\dot{x}_i - [-K_v\dot{e}_{i-1} - K_p e_{i-1} - K_p d - K_p h\dot{x}_{i-1}] \end{aligned} \tag{2.2}$$

Taking Laplace Transform of (2.2),

$$H(s) = \frac{sK_v + K_p}{s^2 + (K_v + K_p h)s + K_p}$$

The characteristic polynomial is $s^2 + (K_v + K_p h)s + K_p$. A necessary condition for a polynomial to be Hurwitz is that all the coefficients of the polynomial must be of the same sign. Going by that rule, we get that $K_v > -K_p h, K_p > 0$.

Applying the D-Decomposition approach to identify the stabilizing gains, we get that the stabilizing gains must satisfy $K_v > -K_p h, K_p > 0$. More information about the D-Decomposition approach has been discussed in the Lateral Controller section.

2.1.2 Lower Level controller

The lower level controller takes in the output from the upper level controller (desired acceleration) and along with the current acceleration, computes the required throttle or braking that is required to achieve the required performance

2.1.2.1 Control Law

Two manually tuned PI controllers with a switching logic to switch between acceleration and braking was designed. The PI controllers have a saturation block at the end to limit the throttle and brake commands that is sent to the vehicle.

$$\begin{aligned} e_i &= \ddot{x}_{ides} - \ddot{x}_i \\ u_i &= -k_p e_i - k_i \int e_i dt \end{aligned}$$

2.2 Lateral Controller

The lateral controller of the vehicle typically consists of a feed forward and a feedback component. The feed forward controller takes the preview information and deciphers one component of the steering angle input. Due to vehicle model discrepancy and other disturbances, the use of a feedback controller is required. Based on the lateral and heading errors, the other component of the steering angle input is computed.

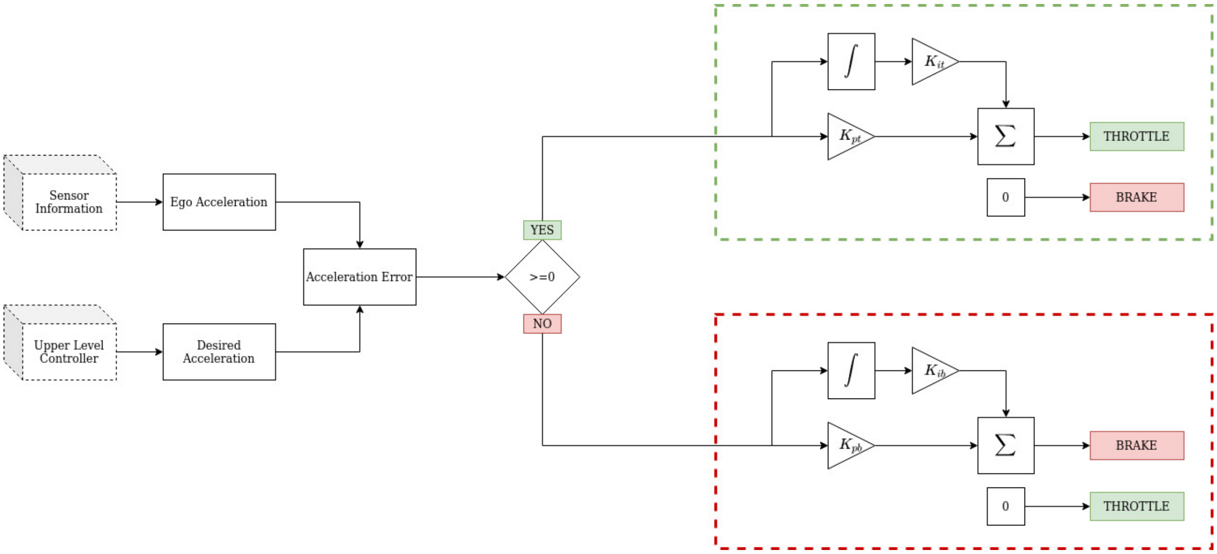


Figure 2.4: Lower Level Controller

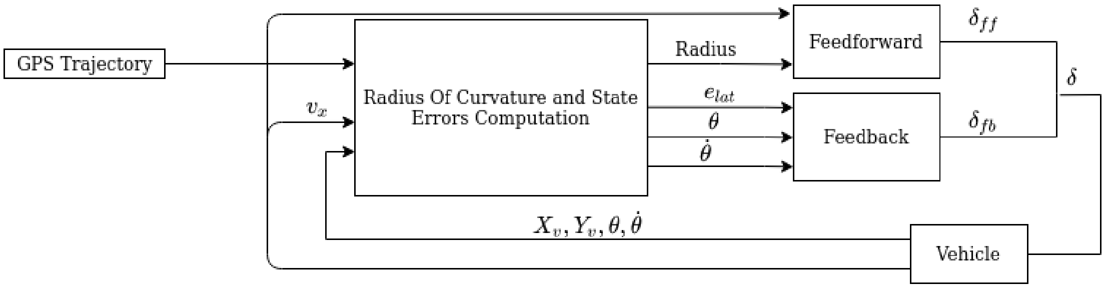


Figure 2.5: Lateral Control

In this section, extracting the required preview information and the steering angle input computation is discussed.

2.2.1 Preview Information Computation

The preview of a vehicle is the lookahead waypoints that the vehicle has access to, that is used during the computation of the input steering angle. The information that is required are the radius of curvature and the error states - lateral error e_{lat}, yaw error e_θ and yaw rate error e_ω.

In earlier works [20], the computation of the radius of curvature of the path ahead (if it is not already classified as a straight line - the maximum deviation from the straight line being a hyperparameter) is done using a least squares method, wherein a geometric error objective function is used - the objective is to minimize the error in the area of the circle thus formed, and as a result, minimize the radius of curvature error (Also called as the *difference-of-squares* geometric error (DOS)). This approach is computationally efficient and well suited for real-time applications when the way points are published from a GPS - RTK system (which has a pretty good accuracy of less than 2 cm when calibrated well). However, if the trajectory way points to be followed have a higher degree of error, it is more preferred that we employ a robust fitting of the way points, thereby effectively discarding outlier points that might otherwise result in a wrong fitting of the data points.

de Guevara et. al [21] suggest an iterative method to fit the curves given the waypoints. The fundamental idea behind this approach is to define the objective function to be minimized as the absolute geometric error. The absolute geometric error criterion is well known to lead to robust estimations of required parameters, however with a caveat being that the function is non-differentiable.

Given the points $\mathcal{P} = \{P_i = (x_i, y_i), i = 1, 2, ..., n\}$, find the center (a, b) and the radius of curvature r that minimizes the error function

$$E(a,b) = \sum_{i=1}^{n} |\sqrt{(x_i - a)^2 + (y_i - b)^2} - r(a,b)| \tag{2.3}$$

where $r(a,b) = med\{\sqrt{(x_i - a)^2 + (y_i - b)^2}\}, k = 1, 2, ..., n$. It is well known that the median of all the distances to the points from the center of the circle is the value that minimizes the absolute geometric error. The median is a resistant statistic. Unlike the mean, whose breakaway point (the percentage of incorrect values after which the statistic shoots up to an arbitrary high value) is 0, the breakaway point of the median is 0.5. Therefore, this means that unless more than 50% of the data is corrupted or wrong, the median would still be robust to outlier points and provide a good estimate.

Notice that (2.3) is non-differentiable. In addition, the equation is non-convex as well (To see how, consider the fact that all the points in the circumference yield a error of 0, but every interior point would give us a positive value of error). Therefore, to solve this unconstrained optimization problem, an iterative method was proposed.

To overcome the non-differentiable nature, an iterative method is designed that uses the right and left side derivatives to compute the new search direction to improve the solution and reduce the objective value.

Taking an initial guess and identifying the median radius r, this value is used to partition the set of point into three sets:

- I - All the points in the interior of the circle
- O - All the points in the exterior of the circle
- C - All the points on the circumference of the circle

Writing down the error function, we get the following

$$E(a,b) = \sum_{i\in I}(\sqrt{(x_i-a)^2+(y_i-b)^2}-r) + \sum_{i\in O}(r-\sqrt{(x_i-a)^2+(y_i-b)^2})$$
$$+\sum_{i\in C}|\sqrt{(x_i-a)^2+(y_i-b)^2}-r|$$

The last term is not differentiable. Therefore, we take the partial derivatives - What does this mean? To obtain the partial derivatives, we slightly move the center along the abscissa axis by Δa and write down the new error function. Δa is chosen in such a way that it makes the points on the circumference fall into either the set I or O, but not any points from those sets into C. This is possible due to the fact that there are only a finite set of point in set $\mathcal{P}$.

The trick here is that when there are no points on the circumference, the error function becomes differentiable.

$$\begin{aligned}E(a+\Delta a,b)=&\sum_{i\in I}(\sqrt{(x_i-a-\Delta a)^2+(y_i-b)^2}-r)+\\&\sum_{i\in O}(r-\sqrt{(x_i-a-\Delta a)^2+(y_i-b)^2})\\&+\sum_{i\in C^-}(\sqrt{(x_i-a-\Delta a)^2+(y_i-b)^2}-r)+\\&\sum_{i\in C^+}(r-\sqrt{(x_i-a-\Delta a)^2+(y_i-b)^2})\end{aligned}$$

Now the right derivative of the error function with perturbations on the abscissa axis can be found since

$$\begin{aligned}E'_{a+}(a,b)=&\lim_{\Delta a\to 0}\frac{E(a+\Delta a,b)-E(a,b)}{\Delta a}\\=&\sum_{i\in I}\frac{(x_i-a)}{\sqrt{(x_i-a)^2+(y_i-b)^2}}\\&-\sum_{i\in O}\frac{(x_i-a)}{\sqrt{(x_i-a)^2+(y_i-b)^2}}\\&+\sum_{i\in C}\frac{|x_i-a|}{\sqrt{(x_i-a)^2+(y_i-b)^2}}\end{aligned}$$

Similarly, the other three derivatives can be found as well - $E'_{a-}(a,b), E'_{b+}(a,b), E'_{b-}(a,b)$. A vector $\mathbf{d}=(d_1,d_2)$ is called the descent direction of $E(a,b)$ at (a,b) if there exists a $\epsilon>0$ such that

$$E[(a,b)+\lambda(d_1,d_2)]<E[(a,b)],$$
$$\text{for each}\lambda\in[0,\epsilon)$$

[21] show that there definitely exists a ϵ that reduces the error function cost. Therefore, in the iterative algorithm, unless such a λ is reached, the step size is iteratively multiplied by 0.9. The direction $\mathbf{d}$ is a product of the step size and the partial derivatives computed earlier.

The intuition is as follows: it tries to move the center towards the outside points and away from the inside points. At the same time, the points at the circumference try to avoid the displacement of the center.

As with every iterative method, this is also sensitive to the initial guess provided; however it is often noticed that the average of all the points given is sufficient to reach convergence quickly.

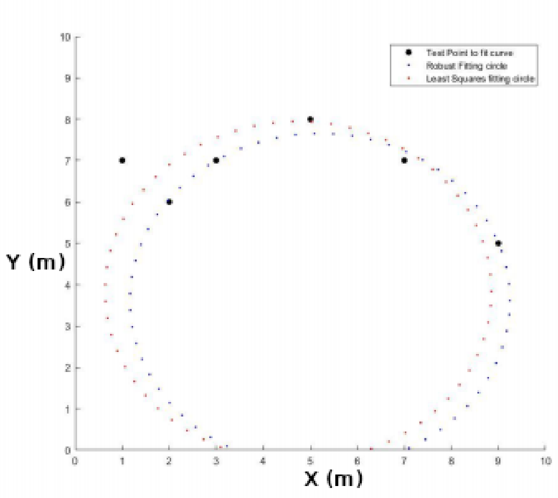


Figure 2.6: Comparing both methods with outliers present

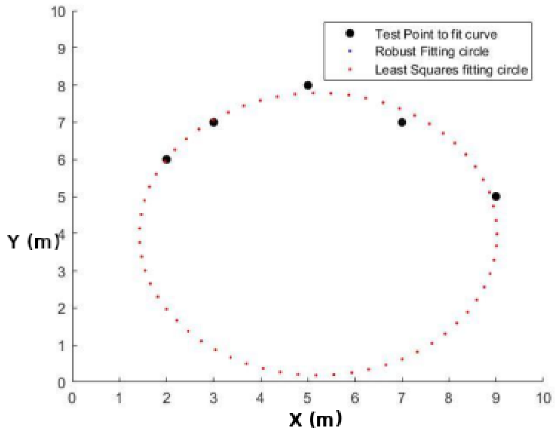


Figure 2.7: Comparing both methods with outliers absent

Once the preview trajectory is identified, it is then straight forward to compute the error states mentioned earlier.

2.2.2 Vehicle Model

The vehicle dynamics model used is a standard dynamic bicycle model with a linear tire model. The steering dynamics is modelled as a second order system.

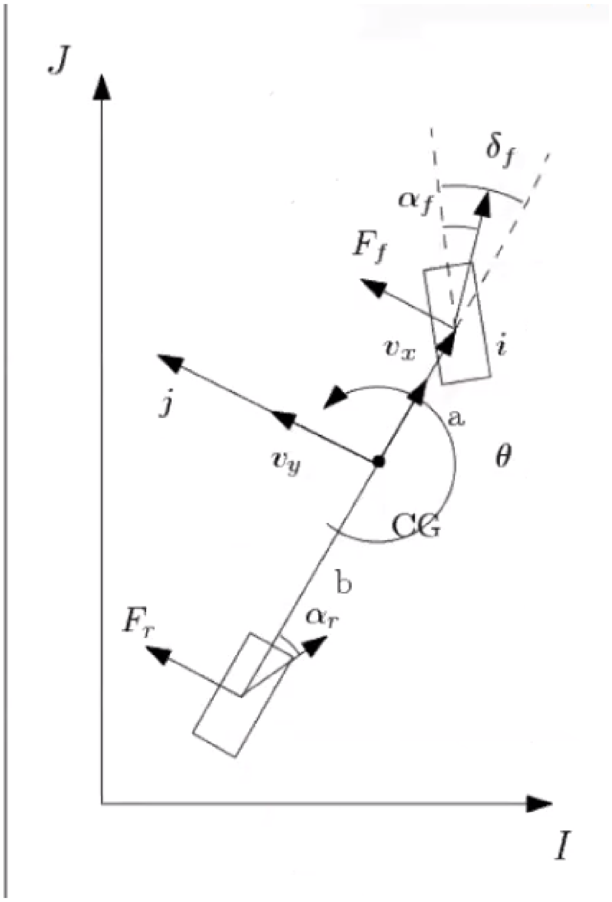


Figure 2.8: Dynamic Bicycle Model

Based on Newton's Laws of Motion, we get the following equations:

Vehicle Dynamics Model

$$m(\frac{dv_y}{dt} + v_x\dot{\theta}) = C_f\delta_f - \frac{C_f + C_r}{v_x}v_y - \frac{aC_f - bC_r}{v_x}\dot{\theta} \tag{2.4}$$

$$I\ddot{\theta} = aC_f\delta_f - \frac{aC_f - bC_r}{v_x}v_y - \frac{a^2C_f + b^2C_r}{v_x}\dot{\theta} \tag{2.5}$$

We consider a few assumptions for further simplification:

- Vehicle's longitudinal speed is a constant, say $v_x = V_0$
- The different between yaw angle and track yaw angle is relatively small, i.e.. $|\theta - \theta_R| << 1$
- The radius of curvature is piece-wise constant, i.e.. $R \approx$ constant
- The radius of Curvature is much greater than the lateral deviation from the given trajectory, i.e. $R >> e_{lat}$
- Ignore quadratic and higher order terms
- $\tilde{\theta} = \theta - \theta_R$

Based on the assumptions, we get the simplify equations as:

$$R\ddot{\theta}_R \approx V_0 \tag{2.6}$$

$$\dot{e}_{lat} \approx V_0(\theta - \theta_R) + v_y \tag{2.7}$$

$$R\ddot{\theta}_R \approx 0 \tag{2.8}$$

$$R\dot{\theta}_R^2 + \ddot{e}_{lat} \approx \frac{dy}{dt} + V_0\dot{\theta} \tag{2.9}$$

Ultimately, we get the final state space equations for (2.4) and (2.5) as follows:

$$M = \begin{bmatrix} m & 0 \\ 0 & I_z \end{bmatrix}$$

$$C = \begin{bmatrix} \frac{C_f + C_r}{V_0} & \frac{aC_f - bC_r}{V_0} \\ \frac{aC_f - bC_r}{V_0} & \frac{a^2C_f + b^2C_r}{V_0} \end{bmatrix}$$

$$L = \begin{bmatrix} 0 & C_f + C_r \\ 0 & aC_f - bC_r \end{bmatrix}$$

$$B = \begin{bmatrix} 1 \\ a \end{bmatrix}$$

$$F = \begin{bmatrix} mV_0^2 + (aC_f - bC_r) \\ a^2C_f + b^2C_r \end{bmatrix}$$

$$x = \begin{bmatrix} e_{lat} \\ \widetilde{\theta} \end{bmatrix}$$

$$\mathbf{M}\ddot{x} + \mathbf{C}\dot{x} + \mathbf{L}x = \mathbf{B}C_f\delta_f - \mathbf{F}(\frac{1}{R}) \tag{2.10}$$

Steering System Model

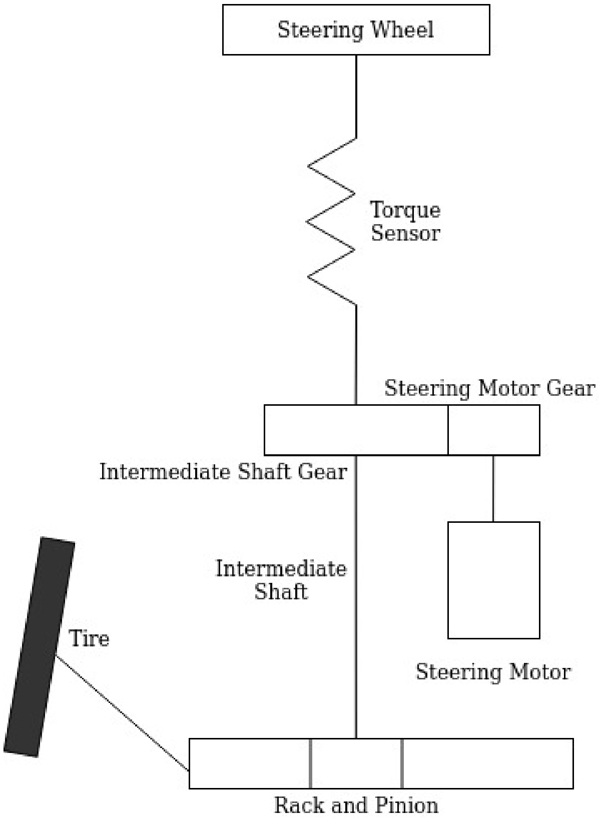


Figure 2.9: EPS Model

Parasitic dynamics of the steering actuator supplies upper bound for the values of the gains k_e, k_θ, k_ω. Therefore, modeling the steering dynamics is important. The electric power steering (EPS) dynamics can be described as follows: Defining the following terms,

δ_f - Tire steering command

δ_c = Tire steering angle

J_ω - Steering intermediate shaft inertia

b_ω - Viscous Damping Factor

K_r - Torsional stiffness

we can model the steering dynamic as a second order system [20].

$$H(s) = \frac{\delta_f}{\delta_c} = \frac{K_r}{J_\omega s^2 + b_\omega s + K_r}$$

The ODE used in the simulations can be described as

$$\tau_2 \ddot{\delta}_f + \tau_1 \dot{\delta}_f + \delta_f = \delta_c \tag{2.11}$$

where

$$\tau_2 = \frac{J_\omega}{K_r}, \tau_1 = \frac{b_\omega}{K_r}$$

2.2.3 Control Law

The steering control input is defined as

$$\delta_f = \delta_{ff} + \delta_{fb} \tag{2.12}$$

The feed forward controller can be determined geometrically as follows:

$$\delta_{ff} = \frac{a+b}{R} + K_{sg} \frac{V^2}{gR} \tag{2.13}$$

where K_{sg} is defined as the steering gradient of the vehicle. The steering gradient can be computed as

$$K_{sg} = \frac{W_f}{C_f} - \frac{W_r}{C_r} \tag{2.14}$$

The fixed structure feedback controller has 3 terms - each corresponding to the 3 error states multiplied with their respective gain values. The lateral velocity error is not considered as one of the error states since it cannot be measured directly or is too noisy when measured. Therefore, the control law turns out to be

$$d_{fb} = -k_e e_{lat} - k_\theta e_\theta - k_\omega e_\omega \tag{2.15}$$

2.2.4 Stabilizing gains

For identifying the gains values in the gain parameter space that provide stability to the vehicle at all speeds, we try to identify the stable signature invariant parameter set at different speed values and sample a point that is present in the stable set at all speeds.

The closed loop characteristic polynomial can be written as

Closed Loop Characteristic Polynomial

$$\Delta(s) = A_6 s^6 + A_5 s^5 + A_4 s^4 + A_3 s^3 + A_2 s^2 + A_1 s + A_0 \tag{2.16}$$

where

$$\begin{aligned}
A_6 &= \tau_2 Im \\
A_5 &= \tau_1 Im + \tau_2 \frac{(I + ma^2)C_f + (I + mb^2)C_r}{V_0} \\
A_4 &= \tau_1 \frac{(I + ma^2)C_f + (I + mb^2)C_r}{V_0} + \tau_2 (\frac{(a+b)^2 C_f C_r}{V_0^2} - m(aC_f - bC_r)) + mI \\
A_3 &= \tau_1 (\frac{(a+b)^2 C_f C_r}{V_0^2} - m(aC_f - bC_r)) + \frac{(I + ma^2)C_f + (I + mb^2)C_r}{V_0} + k_\omega C_f ma \\
A_2 &= \frac{(a+b)^2 C_f C_r}{V_0^2} + k_e C_f I + k_\theta C_f ma - m(aC_f - bC_r) + \frac{k_\omega C_f C_r (a+b)}{V_0} \\
A_1 &= \frac{C_f C_r (a+b)}{V_0} (bk_e + k_\theta) \\
A_0 &= k_e C_f C_r (a+b)
\end{aligned}$$

The Characteristic Polynomial must be Hurwitz

The coefficients of all the terms in the characteristic polynomial must be of the same sign, which is a necessary condition for a polynomial to be Hurwitz. Since $A_6 > 0$, the rest of the terms must be positive.

This gives rise to certain inequalities that must be satisfied by the gain values in order for the system to be necessarily stable.

$$
\begin{aligned}
k_\omega &> \frac{\tau_1(\frac{(a+b)^2 C_f C_r}{V_0^2} - m(aC_f - bC_r)) + \frac{(I+ma^2)C_f + (I+mb^2)C_r}{V_0}}{C_f m a} && (2.17)\\
k_\theta &> -bk_e && (2.18)\\
k_e &> 0 && (2.19)\\
k_e C_f I + k_\theta C_f m a + \frac{k_\omega C_f C_r (a+b)}{V_0} &> m(aC_f - bC_r) - \frac{(a+b)^2 C_f C_r}{V_0^2} && (2.20)
\end{aligned}
$$

Even-Odd Decomposition

The closed loop characteristic polynomial is written as a sum of two polynomials - the even degree terms comprise of one polynomial and the odd degree term comprise of the other. The reason for doing this would be more apparent as we explore the domain decomposition approach to identify the stabilizing gains.

$$\Delta(s) = P_e(s^2) + sP_o(s^2) \tag{2.21}$$

where

$$
\begin{aligned}
P_e(s^2) &= A_6 s^6 + A_4 s^4 + A_2 s^2 + A_0\\
P_o(s^2) &= A_5 s^4 + A_3 s^2 + A_1
\end{aligned}
$$

Domain Decomposition Approach

Given a parameter space to sample a gain value from, it is required to identify a set of gain values that is stable. The method classifies the space into disjoint signature invariant sets (sets that have the same signature), from which possible candidate sets are found and further investigation on their stability is conducted by sampling points from the set. Any point in a stable set would result in a stable system.

The signature of a set is the ordered tuple of values that correspond to the number of zero real roots, positive real roots and negative real roots (n_0, n_+, n_-).

The boundaries where the roots change the sign or become zero are identified. Such boundaries are the locations where the signature of the set would change. The real roots boundary and the imaginary roots boundary are the two boundaries where the signature would change.

$\Delta(0)$ is the one of the real roots boundary. Also, when the leading coefficient vanishes, it produces another real roots boundary.

Real roots boundary

$$\Delta(0) = 0 \implies A_0 = (a+b)C_f C_r k_e = 0$$

$k_e = 0$ is ot a feasible solution, so this boundary condition is ignored

Leading Coefficient term can become zero, but this doesn't offer a boundary as well

$$A_6 = \tau_2 Im = 0$$

Now coming to the complex roots boundary, we find that the boundary is given by:

$$\Delta(j\omega) = 0, \omega \in (0, \infty) \tag{2.22}$$

Values of λ that would solve both the polynomials simultaneously would result in an expression with the gain parameters. Substituting distinct values of k_e, two expressions with two variables

k_θ, k_ω are obtained and solved to get the imaginary boundary in the k_e plane.

There are two ways to go about solving the system of equations from this point. If the two parameters in consideration are decoupled, then they can be explicitly stated as a function of the third parameter and can be plotted for all $\omega > 0$

The closed loop characteristic polynomial can be written in the following manner as shown below. The gain parameters are decoupled, as shown, and $D_1(s), D_2(s), D_3(s), D_4(s)$ are the respective coefficient polynomials.

$$\Delta(s, k_e, k_\theta, k_\omega) = D_1(s) + k_e D_2(s) + k_\theta D_3(s) + k_\omega D_4(s) \tag{2.23}$$

Even-odd decomposition can be applied to the coefficient polynomials as shown below:

$$\begin{aligned} D_1(s) &= D_{1e}(s^2) + sD_{1o}(s^2) \\ D_2(s) &= D_{2e}(s^2) + sD_{2o}(s^2) \\ D_3(s) &= D_{3e}(s^2) + sD_{3o}(s^2) \\ D_4(s) &= D_{4e}(s^2) + sD_{4o}(s^2) \end{aligned}$$

Substituting $s = j\omega$,Equation 2.21 can be split and written it as two separate equations, corresponding to the real and imaginary part of the equation. Writing them in matrix form:

$$\underbrace{\begin{bmatrix} D_{3e}(\omega) & D_{4e}(\omega) \\ D_{3o}(\omega) & D_{4o}(\omega) \end{bmatrix}}_{A_{\theta\omega}} \begin{bmatrix} k_\theta \\ k_\omega \end{bmatrix} = \begin{bmatrix} -D_{1e}(\omega) & -k_e D_{2e}(\omega) \\ -D_{1o}(\omega) & -k_e D_{2o}(\omega) \end{bmatrix}$$

Ultimately, the values of k_θ, k_ω as a function of k_e are found

$$\begin{bmatrix} k_\theta \\ k_\omega \end{bmatrix} = \frac{1}{|A_{\theta\omega}|} \begin{bmatrix} D_{4o}(\omega) & -D_{4e}(\omega) \\ -D_{3o}(\omega) & D_{3e}(\omega) \end{bmatrix} \begin{bmatrix} -D_{1e}(\omega) & -k_e D_{2e}(\omega) \\ -D_{1o}(\omega) & -k_e D_{2o}(\omega) \end{bmatrix}$$

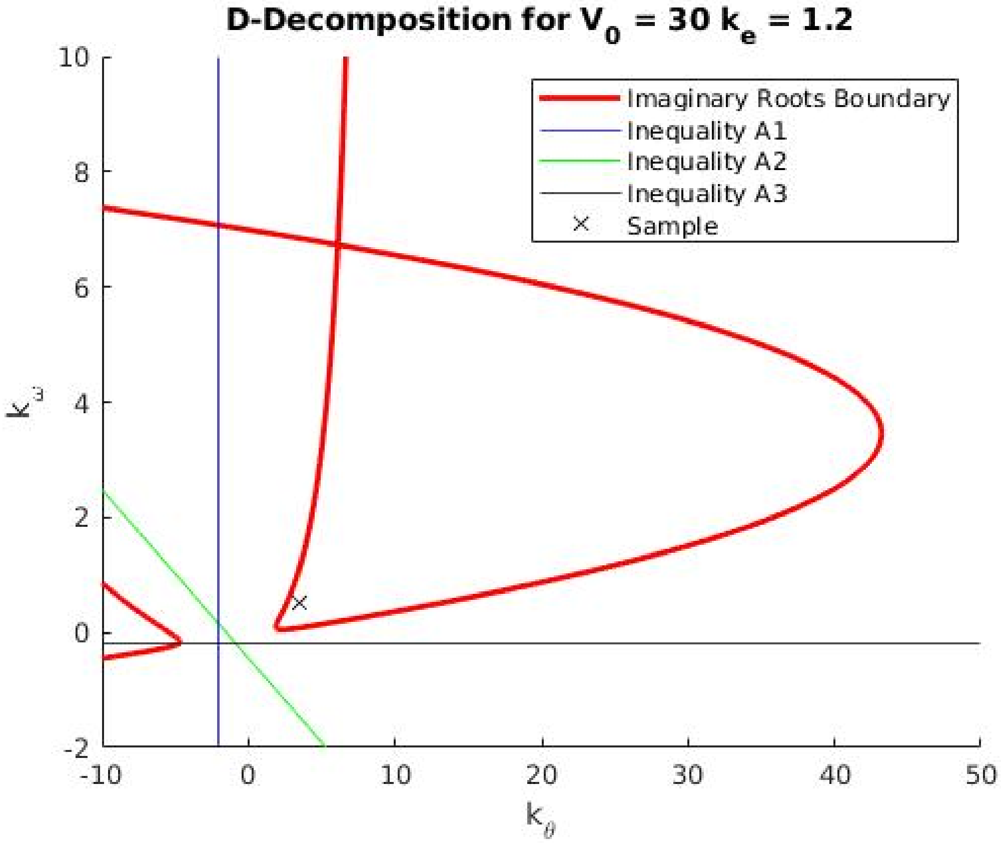


Figure 2.10: Domain Decomposition of gain parameter space

The gain parameter space for $V_0 = 30m/s$ and $k_e = 1.2$ is shown above. As shown, the space is divided into a couple of signature invariant sets and sampling from the feasible sets, it is found that there is one set that results in a stable set, where all the roots are in the LHP. The next step would be to build the plots for various speeds and identify the gains stable at all speeds. As shown below, the gain sample stable at $V_0 = 30m/s$ would not be stable at $V_0 = 50m/s$.

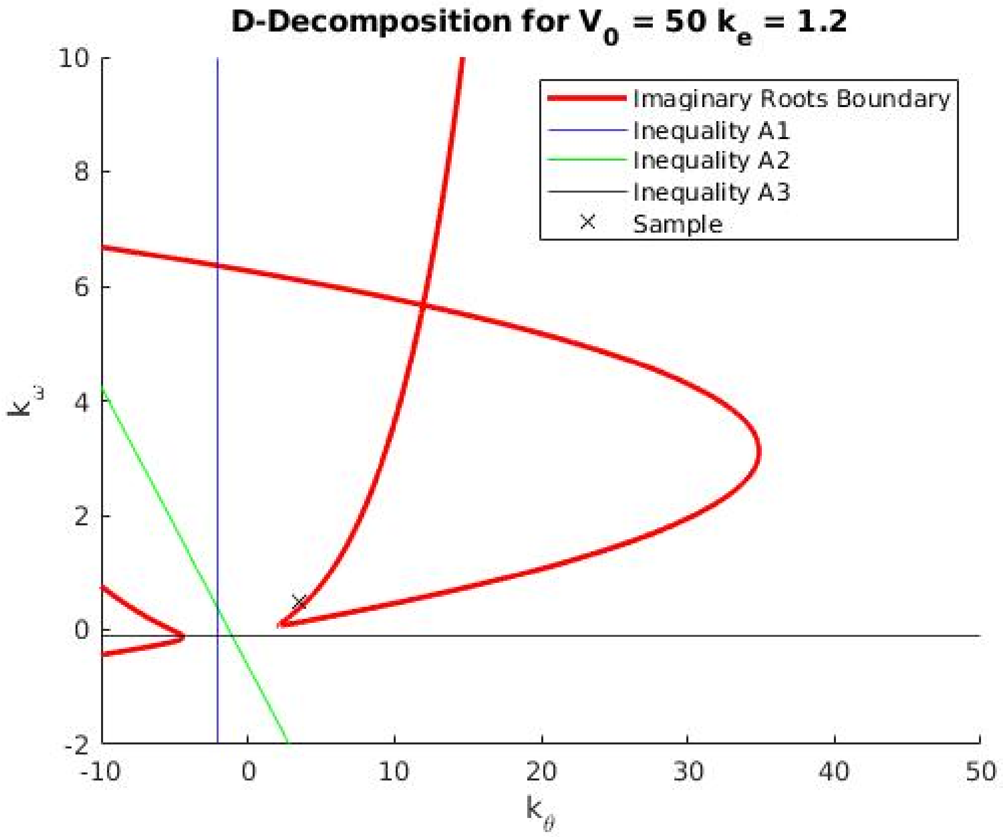


Figure 2.11: Domain Decomposition of gain parameter space - Unstable gain sample

Another method to set the signature invariant gain parameter space is to use the concept of resultant polynomials, which is discussed in the Robustness study section.

2.2.5 V2V Communication in a Platoon

This subsection talks about the communication strategy used while designing the controller for the follower vehicles. It is assumed that every vehicle in the platoon is supplied with the information of a nominal trajectory to track while moving in the lanes. In addition, it is assumed that the platoons often travel in already mapped regions and as a result would have information pertaining to the lane center line.

In case of an emergency lane change (ELC) maneuver, the lead vehicle starts to communicate the new trajectory information that it traverses in. Every vehicle in the platoon has access to the reference trajectory and therefore, communicate just enough information to help the follower vehicles track the new path. The heavy computation is done by the lead vehicle by fitting a trajectory given the ELC maneuver way points. Once the trajectory is fit, the next step would be to compute the radius of curvature and the error states and ultimately compute the required steering angle.

Once these values are computed, the radius of curvature along with the required lateral and heading positions (encoded as the deviation from the reference trajectory) are passed on as information to the follower vehicles with the current GPS coordinates of the lead vehicle.

The follower vehicles collect this information that they receive via V2V communications in a buffer. Based on their GPS position, the vehicle is matched with the closest GPS position available in the buffer and the relevant trajectory information are extracted. The next steps involves computation of the current lateral error and heading error with respect to the baseline, identification of the effective errors (required lateral error - current lateral error) and final synthesis of the input steering angle.

2.3 V2V Architectures

There are two main V2V architectures possible that is explored as part of our efforts to identify a stable way to do the ELC maneuver. There are as follows

1. Follower vehicle using only preceding vehicle's information: In this scenario, every follower vehicle uses only the preceding vehicles radius of curvature and error states information, as shown in the Figure 2.12.

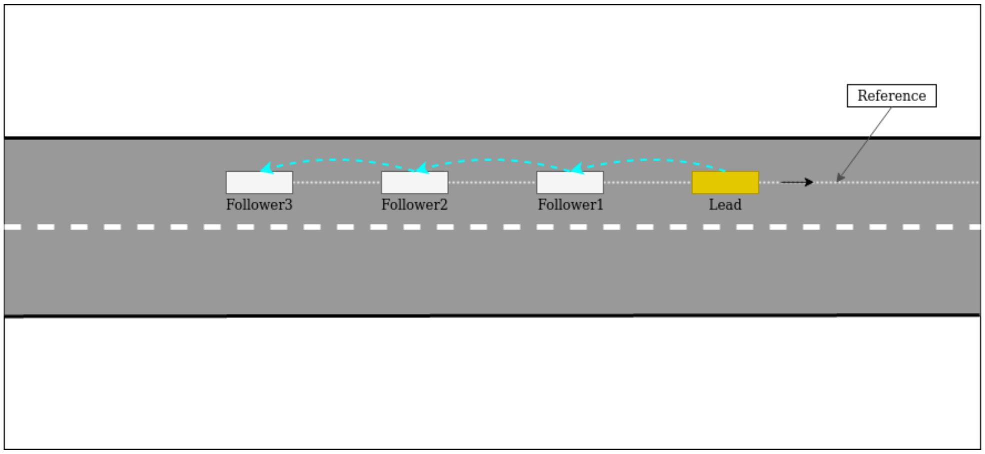


Figure 2.12: V2V with preceding vehicle information

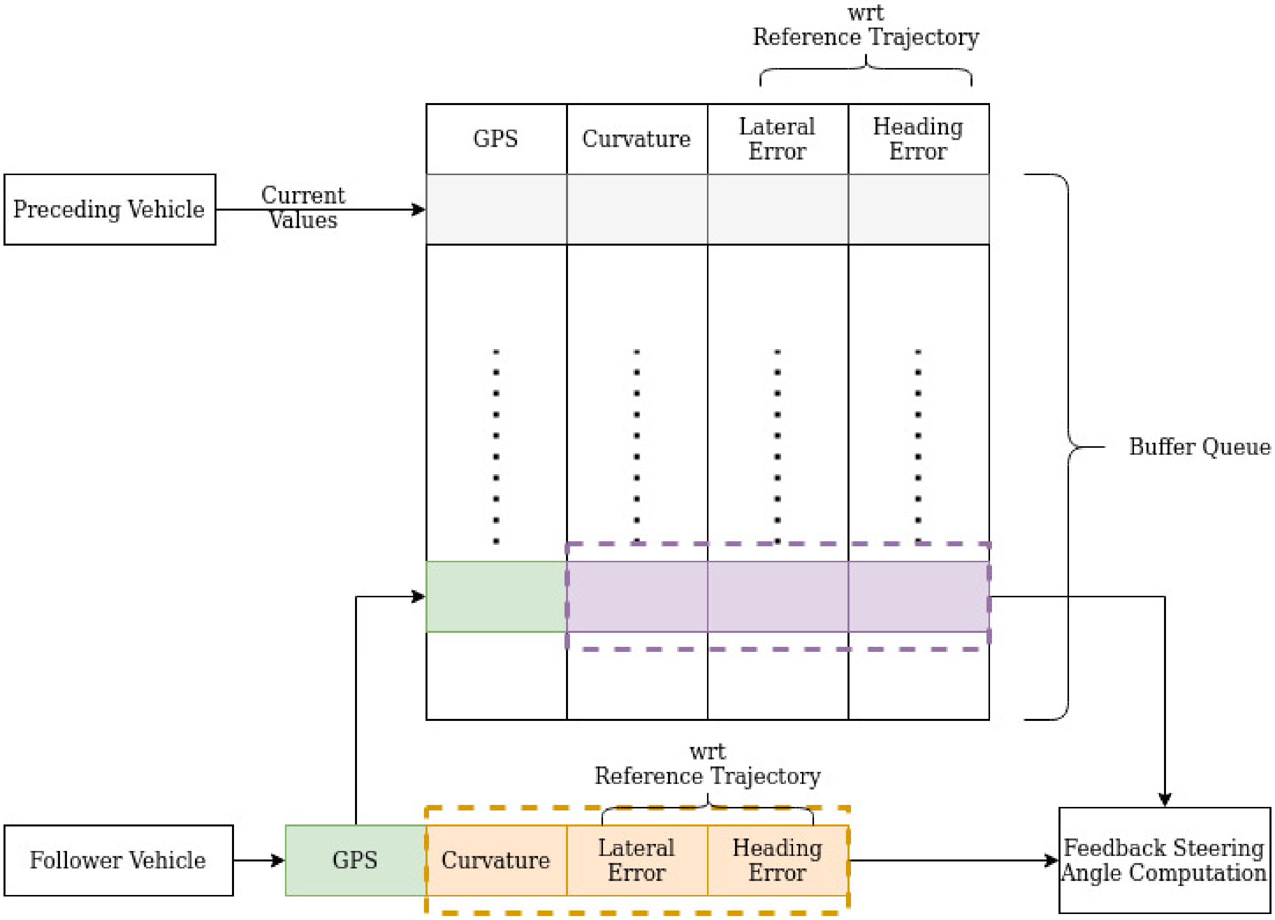


Figure 2.13: Preceding Vehicle V2V Communication

2. Follower vehicle using only lead vehicle's information: In this scenario, every follower vehicle uses only the lead vehicles radius of curvature and error states information, as shown in the Figure 2.14.

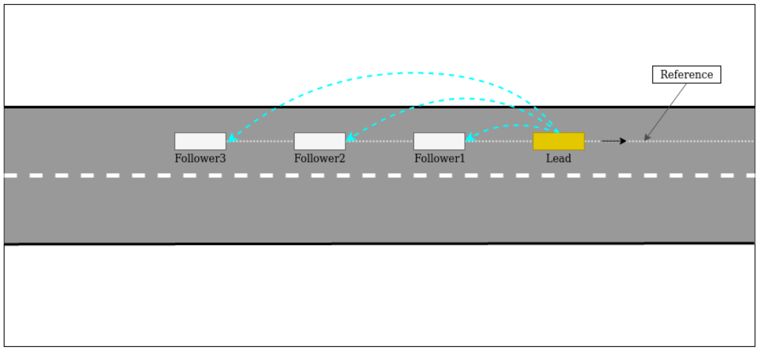


Figure 2.14: V2V with lead vehicle information

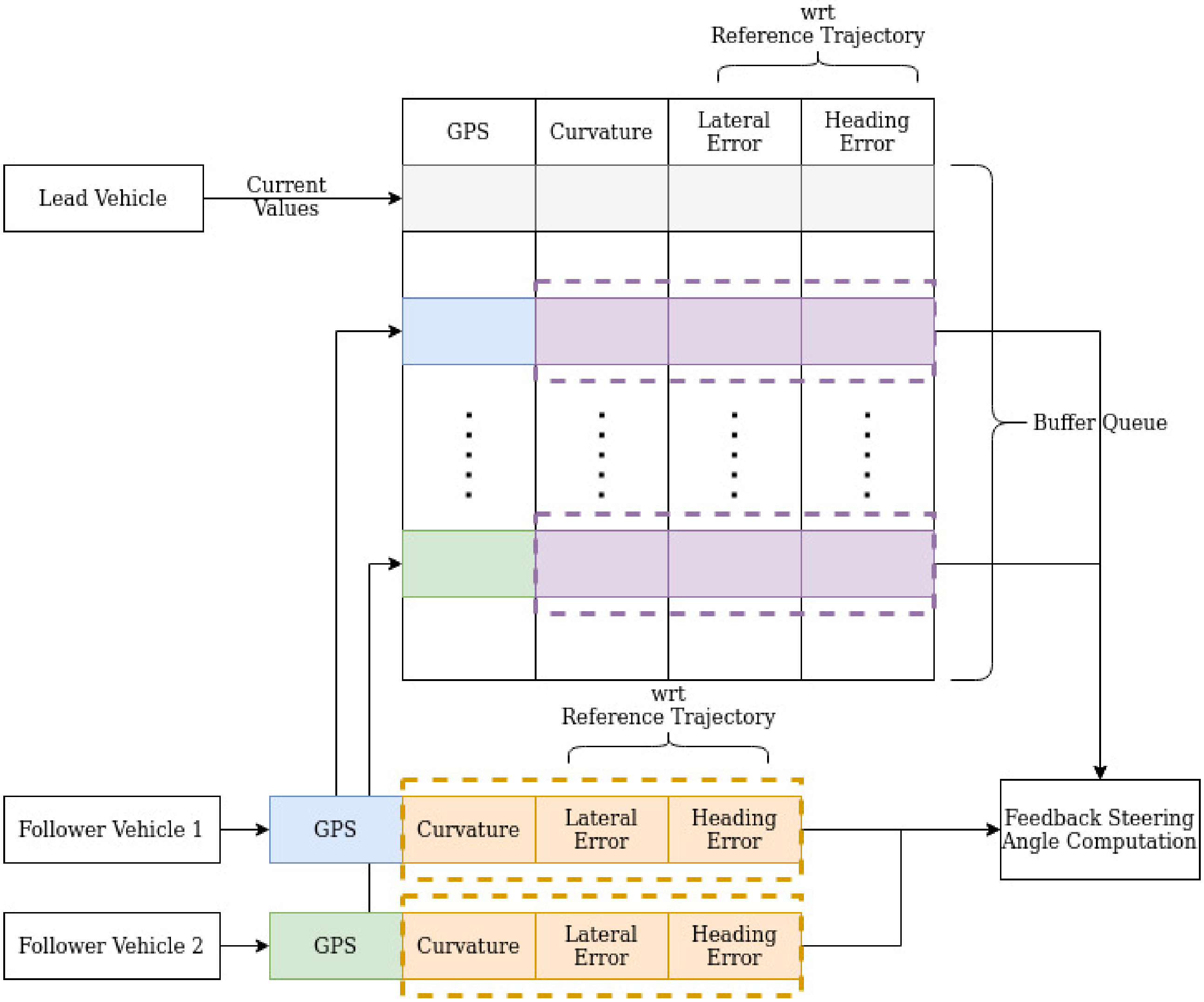


Figure 2.15: Lead Vehicle V2V Communication

3. TORCS SIMULATION

3.1 About TORCS

The open racing car simulator (TORCS) is a modern, modular, highly-portable multi-player and multi-agent car simulator. It is very flexible to use and has featured in many artificial intelligence research papers. Completely written in C++ and available for anyone to use under the GPL license, the use of TORCS for academic research is multi-fold. Some of the major reasons are:

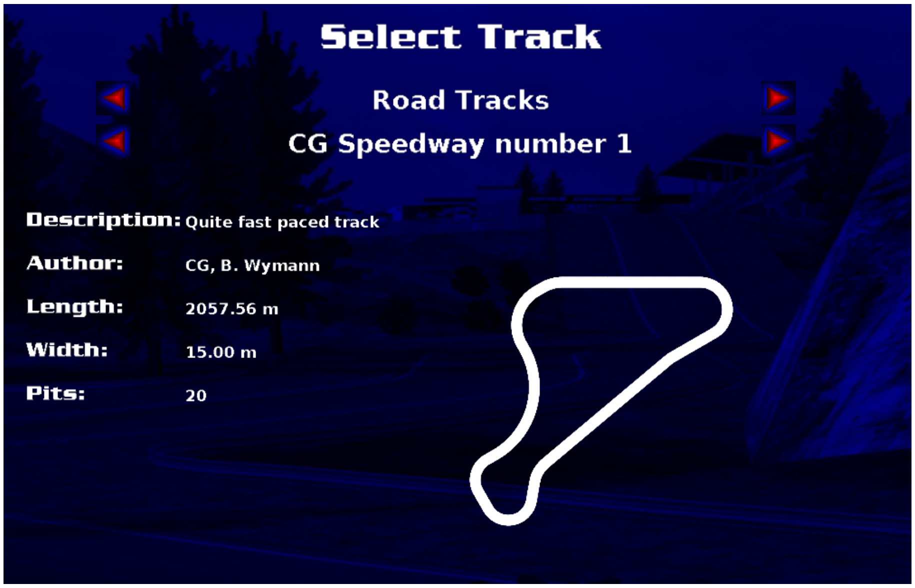
- It features a sophisticated physics engine (aerodynamics, fuel consumption, suspension, differential, etc.) and a pretty good graphics engine that helps us visualize the races pretty well from multiple angles
- The ease of quickly obtaining sensor data from the vehicles and sending back in the control inputs
- Since it is open source, the source code is freely available and many modifications can be easily made to customize the simulator to our requirements

Figure 3.1: The Open Racing Car Simulator

Since 2008, TORCS has also played an important role in various research fields within the IEEE Conference on Computational Intelligence and Games, where it appears as a base for 4 to 6 projects every year.

Features of the simulator

- Starting with the track design, custom tracks can be quickly designed to test out the various types of maneuvers as and when needed. Currently straight line segments and segments with constant or linearly increasing radius of curvature can be created using their trackgen software.



Select Track

Road Tracks

CG Speedway number 1

Description: Quite fast paced track

Author: CG, B. Wymann

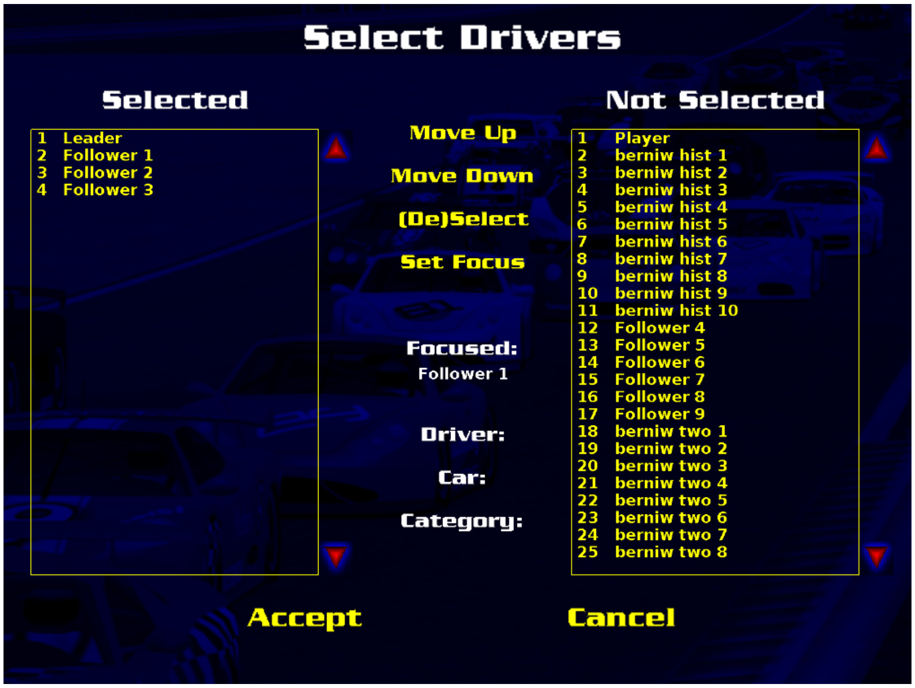
Length: 2057.56 m

Width: 15.00 m

Pits: 20

Figure 3.2: The Open Racing Car Simulator - Tracks

- Multiple new players can be created and can run with their own controllers, based on various sensor information available. Algorithms involving decision making, based on the actions of other agents in the environment have been simulated in the past; so has many learning-based techniques that sometimes involve end-to-end learning which require the camera information to be feed as an input.



Selected

1 Leader
2 Follower 1
3 Follower 2
4 Follower 3

Focused:
Follower 1

Not Selected

1 Player
2 berniw hist 1
3 berniw hist 2
4 berniw hist 3
5 berniw hist 4
6 berniw hist 5
7 berniw hist 6
8 berniw hist 7
9 berniw hist 8
10 berniw hist 9
11 berniw hist 10
12 Follower 4
13 Follower 5
14 Follower 6
15 Follower 7
16 Follower 8
17 Follower 9
18 berniw two 1
19 berniw two 2
20 berniw two 3
21 berniw two 4
22 berniw two 5
23 berniw two 6
24 berniw two 7
25 berniw two 8

Figure 3.3: The Open Racing Car Simulator - Players

- The physics engine is pretty sophisticated. The vehicle dynamics modeling includes the aerodynamics, traction control etc. The simulation features a simple damage model, collisions, tire and wheel properties (springs, dampers, stiffness, ...), different drive trains (AWD, FWD, RWD), different types of differentials, aerodynamics (ground effect, spoilers, ...) and much more. It integrates differential equations with Euler steps, time step is 0.002 seconds (500Hz, simulation time). The race engine runs as well with this rate, but the robots(cars) are just called all 0.02 seconds (50Hz).

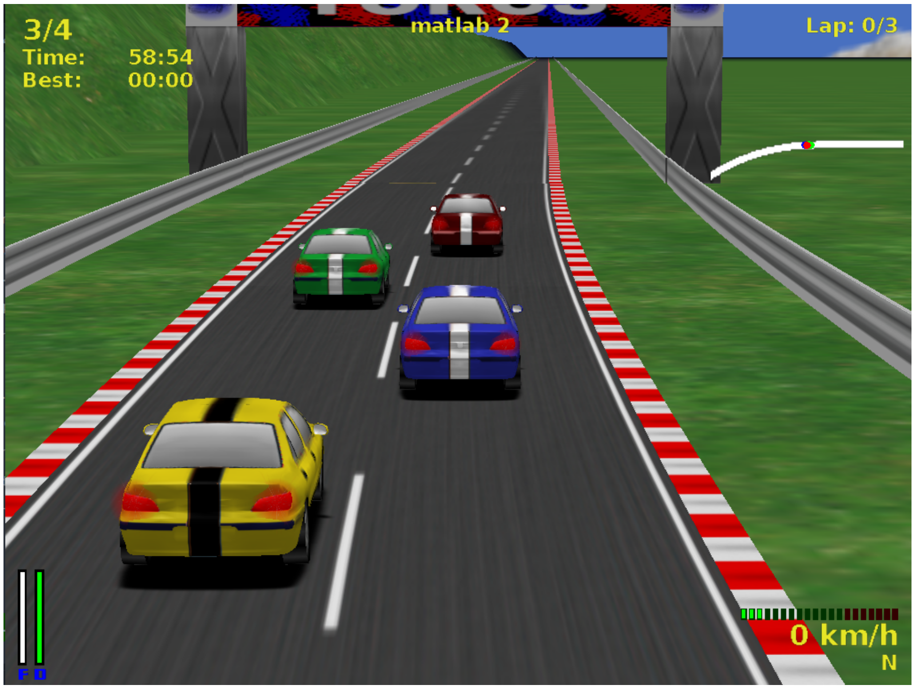


Figure 3.4: The Open Racing Car Simulator - Start

- Its open source nature allows us to create interfaces with other software, such as Simulink for quick prototyping. In addition, the information available to the user can also be modified. For eg., assuming the center of the road is the reference trajectory, additional sensor data parameters like lateral error and heading error can be created and used directly as an input to the controller, instead of computing them outside every time.

3.2 Client-Server Architecture - Data Transfer

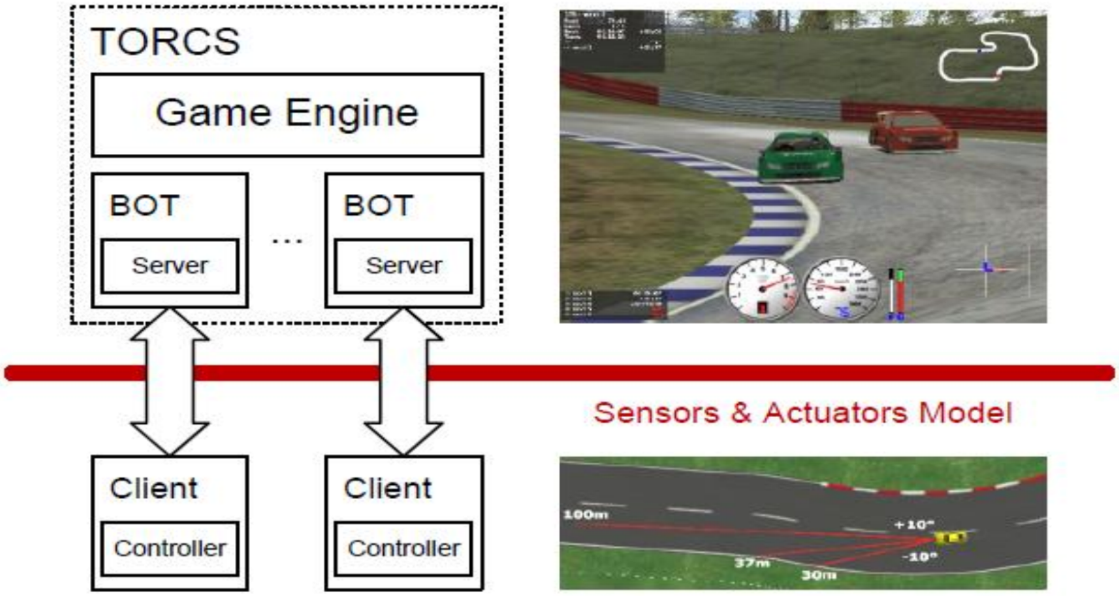


Figure 3.5: The Open Racing Car Simulator - Architecture

As shown above3.5, the TORCS architecture is that of a client-server architecture. Essentially if multiple robots(cars) are spawned at once, that many servers are created and run using the same physics engine. On the client side, every vehicle has their own controller. Based on the sensor data from the server to the client, the controller decides the best control inputs to be sent to the server, which is then used to propagate the car ahead. This happens at a frequency of 50 Hz.

3.2.1 Sensor Data

From the game engine, once the states have been propagated for 0.02 seconds based on the previous control inputs, the sensor information is sent to the controllers to determine the next control input. The flexibility of the simulator makes sure that we can access the sensor information of not just the ego vehicle, but also of all the other vehicles in the simulation. This enables the user to implement algorithms involving V2V communications pretty quickly as well. In addition, the source code can be quickly modified to include new sensor information, such as lateral and heading errors with respect to the center line of the lane. The following values have been used in the simulations as sensor information

Value	**Type**	**Description**
position	(x, y, z)	GPS position of the vehicle
velocity	(v_x, v_y, v_z)	Velocity of the vehicle
acceleration	(a_x, a_y, a_z)	Acceleration of the vehicle
angle	θ	Yaw Angle of the vehicle
angularVelocity	ω	Yaw Rate of the vehicle
headingError	e_θ	Yaw Angle Error wrt road baseline
lateralError	e_{lat}	Lateral Error wrt road baseline
roadCurvature	R	Radius of Curvature at that instant
roadDistance	s	Station value starting from beginning of race
engineRPM	ω_s	Current Engine Speed

Table 3.1: Sensor Information

3.2.2 Control Data

Control data is the data that is sent from the client(controller) to the server(robot) based on the sensor information that it received earlier. The following values can be sent in as a control input

Control Input	Range	Description
accel	[0,1]	Throttle
brake	[0,1]	Braking
clutch	[0,1]	Clutch
gear	[-1,7]	Gear Value
steering	[-1,1]	Steering: -1 to 1 means from complete left to complete right, corresponding to an angle of 0.785398 rad

Table 3.2: Control inputs

3.3 Source Code Modifications/Additions

Being open source, the source code can be modified to suit our needs. The following modifications and additions were made to convert the simulator into one that could be used for our platooning application

- Use a Simulink Plugin to access the sensor data from TORCS and as a result, be able to design the controller in Simulink. [14]

- Create multiple players (Leader and follower vehicles) with their own individual controllers. The controller can be written in C++ or MATLAB(Simulink) and based on what information the controller has access to, different V2V architectures can be simulated.

- Modify source code to be able to access the heading error and lateral error with respect to the road baseline directly from the server instead of computing it on the client side.

- Create multiple cars with different parameters (such as varying mass and moment of inertia) to simulate different passenger count

- Create custom tracks with straight line, constant and linearly increasing/decreasing radius of curvature for simulating various scenarios.

3.4 Results

In this section, the results of the simulation for the different V2V architectures are presented. For the simulations, a homogeneous platoon is taken with the following parameters for each car in the platoon.

Parameter	**Description**	**Value**	**Unit**
m	Vehicle Mass	1605	kg
I	Vehicle Inertia	2045	$kg.m^2$
a	Distance from front axle to C.G	1.488	m
b	Distance from rear axle to C.G	1.712	m
C_f	Front Tire Cornering Stiffness	77	kN/rad
C_r	Rear Tire Cornering Stiffness	77	kN/rad
K_r	Steering intermediate Shaft stiffness	71.4	$N.s/rad$
b_ω	Steering intermediate shaft and rack damping factor	3.7515	$N.m.s/rad$
J_ω	Steering intermediate shaft and rack inertia	0.01258	$kg.m^2$

Table 3.3: Car Parameters

For the V2V architecture using just the preceding vehicle information alone, the following are the results:

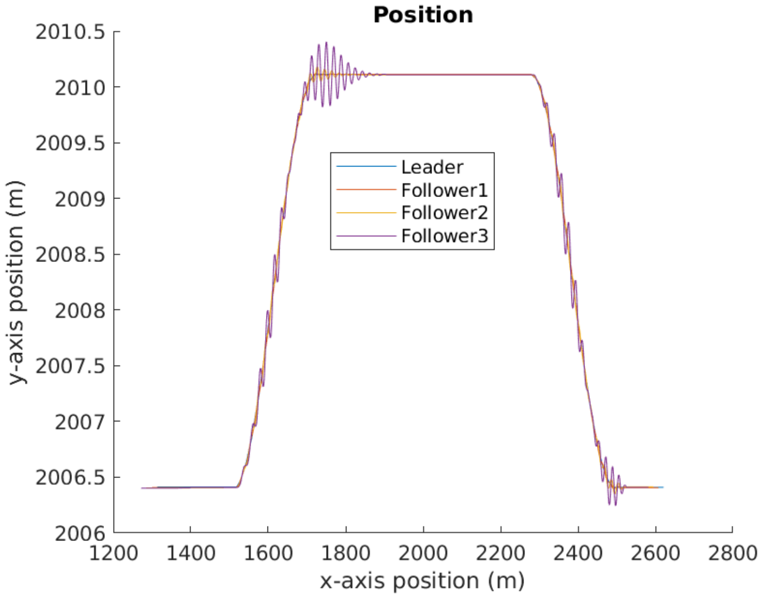


Figure 3.6: Preceding vehicle V2V - Lateral Error

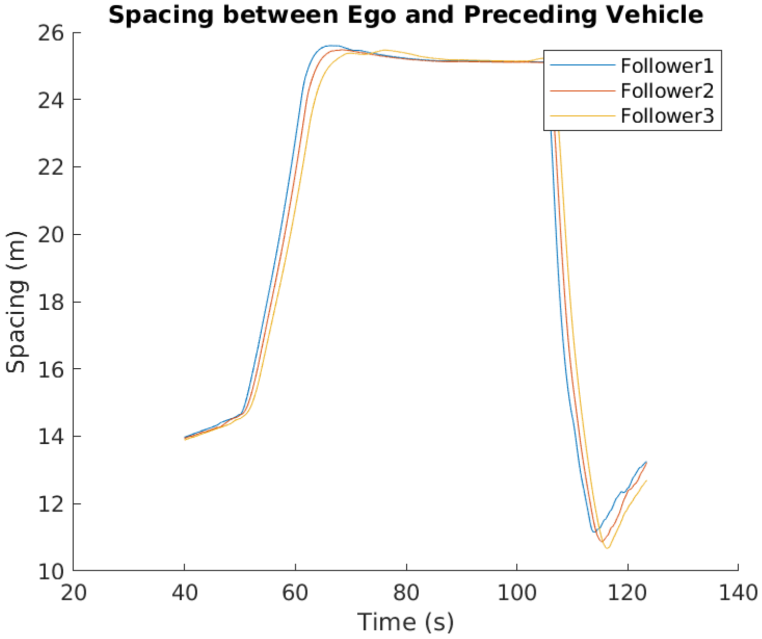


Figure 3.7: Preceding vehicle V2V - Spacing

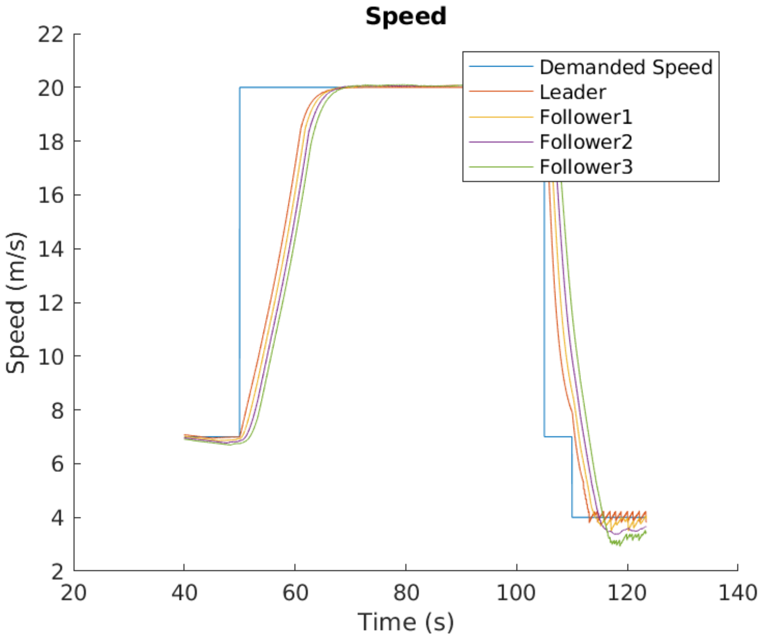


Figure 3.8: Preceding vehicle V2V - Speed

From these results, it is quite apparent that in the case where the preceding vehicle's information was the only information being used, the lateral errors continued to grow as we moved along the length of the platoon, suggesting that the string is unstable in the lateral sense for this architecture. The reason for this would be the coupled lateral dynamics that is involved by using the preceding vehicle's information.

For the V2V architecture using just the lead vehicle information alone, the following are the results:

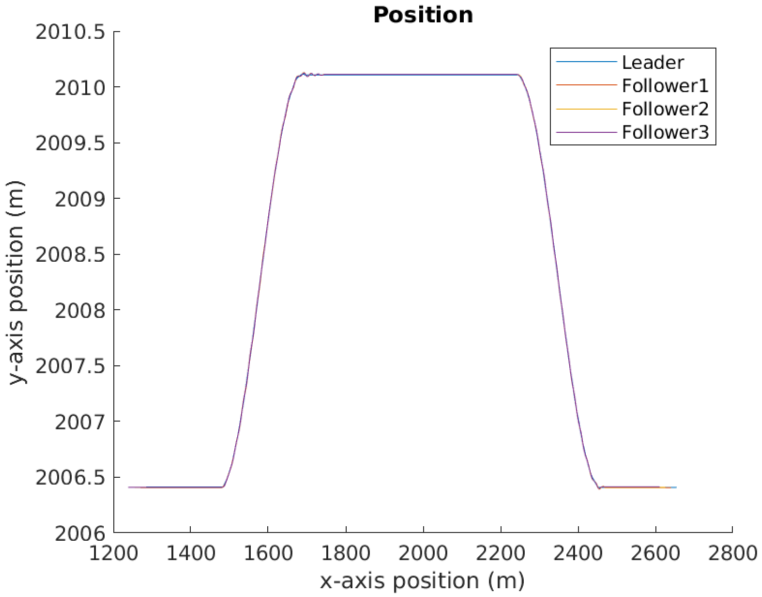


Figure 3.9: Leading vehicle V2V - Lateral Error

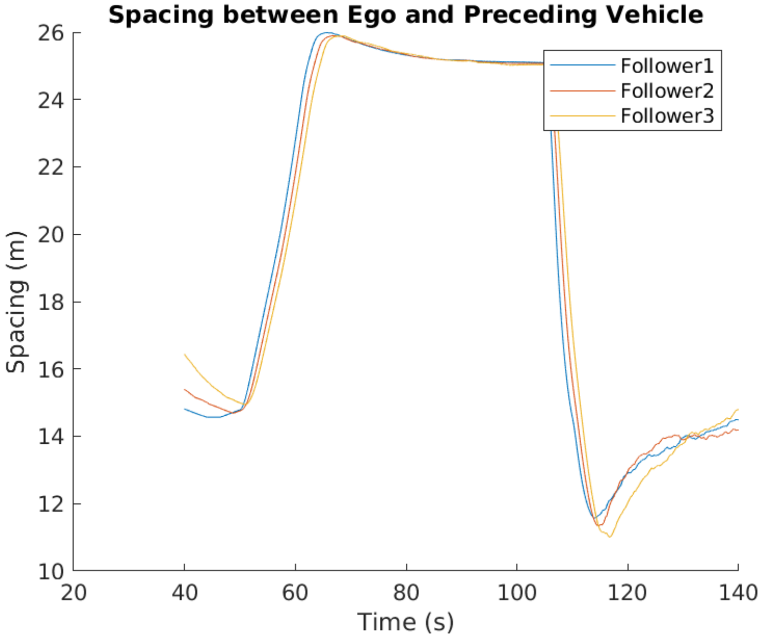


Figure 3.10: Leading vehicle V2V - Spacing

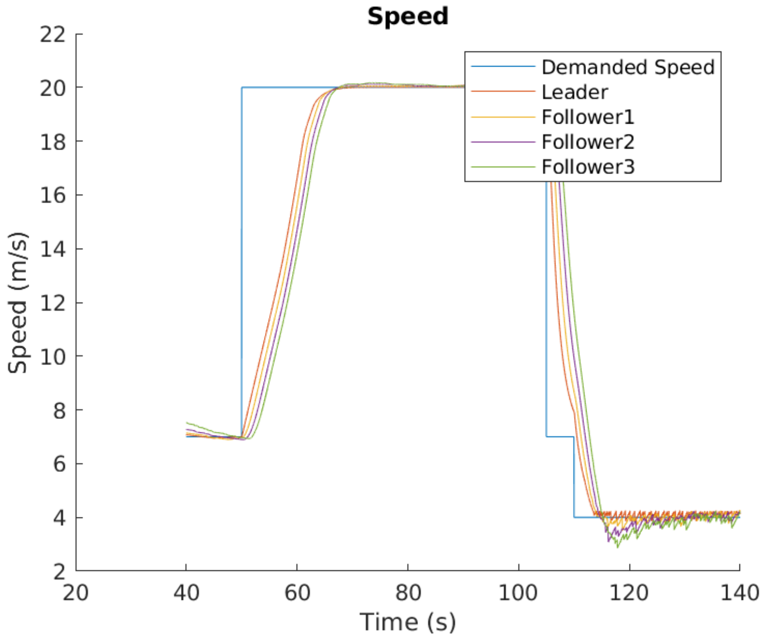


Figure 3.11: Leading vehicle V2V - Speed

Table 3.4: Maximum Absolute Lateral Error (m) wrt Lead Vehicle Trajectory

V2V Communication	**Preceding Vehicle Info**	**Lead Vehicle Info**
Follower1	0.05	0.05
Follower2	0.11	0.05
Follower3	0.37	0.05

In case of using the lead vehicle's information, the results clearly show that all the follower vehicles have a lateral error that doesn't exceed a particular value and is consistent throughout the platoon. This is due to the fact that by sending the same trajectory information to all the vehicles in the platoon, the lateral dynamics of a follower vehicle is decoupled with the preceding vehicles (except for the lead vehicle) and hence result in a stable platoon with non-increasing lateral errors down the platoon.

4. ROBUSTNESS STUDY

In this chapter, a study of how robust the above designed controller is for varying parameters is shown. The question addressed here is as follows: if the system is still stable with the chosen gains for a 30% difference in mass and moment of inertia.

4.1 Why the need for robustness study?

When the lateral controller is designed, an implicit assumption that the values of the parameters are going to remain constant is made. It would not be an viable assumption to make in the real world due to the addition of passengers and cargo onto the vehicle. This would definitely change a couple of parameter values taken constant for the vehicle, especially the mass m and the moment of inertia I. An assumption that the values a, b which are affected by the position of the center of gravity, is assumed to be negligible.

Building from the previous discussions about finding k_e, k_θ, k_ω, now given the gain values, what are the different mass values until which we can use these gain values?

Combine all the additional masses and represent the change in car parameters using the following two values

- m_p - Total additional mass
- r_p - Radius of Gyration of total mass

The car parameters are modified as follows:

- $m = m + m_p$
- $I = I + m_p r_p^2$

4.2 D-Decomposition Approach

Since the parameters being dealt with (m_p, r_p) are coupled, it would not be feasible to use the earlier method of identifying the boundary conditions - where the gain parameters were decoupled.

Therefore, the concept of resultant polynomials is used to eliminate ω to generate the imaginary boundaries.

The closed loop characteristic polynomial is defined as follows:

Closed Loop Characteristic Polynomial

$$\Delta(s) = A_6 s^6 + A_5 s^5 + A_4 s^4 + A_3 s^3 + A_2 s^2 + A_1 s + A_0 \tag{4.1}$$

where

$$\begin{aligned}
A_6 &= \tau_2 (I + m_p r_p)(m + m_p) \\
A_5 &= \tau_1 (I + m_p r_p)(m + m_p) + \tau_2 \frac{((I + m_p r_p) + (m + m_p)a^2)C_f + ((I + m_p r_p) + (m + m_p)b^2)C_r}{V_0} \\
A_4 &= \tau_1 \frac{((I + m_p r_p) + (m + m_p)a^2)C_f + ((I + m_p r_p) + (m + m_p)b^2)C_r}{V_0} \\
&\quad + \tau_2 (\frac{(a+b)^2 C_f C_r}{V_0^2} - (m + m_p)(aC_f - bC_r)) + (m + m_p)(I + m_p r_p) \\
A_3 &= \tau_1 (\frac{(a+b)^2 C_f C_r}{V_0^2} - (m + m_p)(aC_f - bC_r)) \\
&\quad + \frac{((I + m_p r_p) + (m + m_p)a^2)C_f + ((I + m_p r_p) + (m + m_p)b^2)C_r}{V_0} + k_\omega C_f (m + m_p) a \\
A_2 &= \frac{(a+b)^2 C_f C_r}{V_0^2} + k_e C_f (I + m_p r_p) + k_\theta C_f (m + m_p) a - (m + m_p)(aC_f - bC_r) + \frac{k_\omega C_f C_r (a+b)}{V_0} \\
A_1 &= \frac{C_f C_r (a+b)}{V_0} (b k_e + k_\theta) \\
A_0 &= k_e C_f C_r (a+b)
\end{aligned}$$

The gain values k_e, k_θ, k_ω are fixed.

The Characteristic Polynomial must be Hurwitz

The coefficients of all the terms in the characteristic polynomial must be of the same sign, which is a necessary condition for a polynomial to be Hurwitz. Since $A_1 > 0$, the rest of the terms must be positive.

This gives rise to certain inequalities that must be satisfied by the gain values in order for the system to be necessarily stable.

$$A_6 > 0$$

$$A_5 > 0$$

$$A_4 > 0$$

$$A_3 > 0$$

$$A_2 > 0$$

Even-Odd Decomposition

$$\Delta(s) = P_e(s^2) + sP_o(s^2) \tag{4.2}$$

where

$$P_e(s^2) = A_6 s^6 + A_4 s^4 + A_2 s^2 + A_0$$
$$P_o(s^2) = A_5 s^4 + A_3 s^2 + A_1$$

Using the same D-Decompositon approach discussed earlier, it is noticed that the real roots boundary do not contribute to a non-trivial solution. Therefore, looking at the imaginary roots boundary after doing the even-odd decomposition:

For $\omega > 0$, we have

$$\text{Let } \lambda = \omega^2$$
$$P_e(\lambda) = -A_6\lambda^3 + A_4\lambda^2 - A_2\lambda + A_0$$
$$P_o(\lambda) = A_5\lambda^2 - A_3\lambda + A_1 \tag{4.3}$$

To solve the equations simultaneously, the idea that the resultant of the two polynomials must be 0, since they have a common root, is used.

Definition: Resultant of two polynomials $f(x) = a_n x^n + ... + a_1 x + a_0, \quad g(x) = b_m x^m + ... + b_1 x + b_0$ is defined by

$$Res(f, g, x) = a_n^m b_m^n \prod_{i,j} (\alpha_i - \beta_j)$$

where $f(\alpha_i) = 0$ for $1 \leq i \leq n$, and $g(\beta_j) = 0$ for $1 \leq j \leq m$

The resultant of two polynomials is equal to zero if and only if they have a root in common.

Definition: Sylvester Matrix is a matrix associated to two univariate polynomials whole entries are the coefficients of the polynomials. The determinant of the matrix is the resultant of the two polynomials.

The two univariate polynomials, where $\lambda = \omega^2$

$$f(\lambda) = -A_6\lambda^3 + A_4\lambda^2 - A_2\lambda + A_0 = 0$$
$$g(\lambda) = A_5\lambda^2 - A_3\lambda + A_1 = 0$$

Sylvester Matrix:

$$Syl(f, g, \lambda) = \begin{bmatrix} A_6 & 0 & A_5 & 0 & 0 \\ A_4 & A_6 & A_3 & A_5 & 0 \\ A_2 & A_4 & A_1 & A_3 & A_5 \\ A_0 & A_2 & 0 & A_1 & A_3 \\ 0 & A_0 & 0 & 0 & A_1 \end{bmatrix}$$

Therefore, the $det(Syl(f, g, \lambda)) = 0$ will provide an equation exclusively in $(\mathbf{m_p}, \mathbf{r_p})$ which can then be plotted.

4.3 Results

Plotting the resultant polynomial defined earlier for a given set of gain values $(k_e, k_\theta, k_\omega) = (1.2, 1, 0.5)$, the following plot is obtained

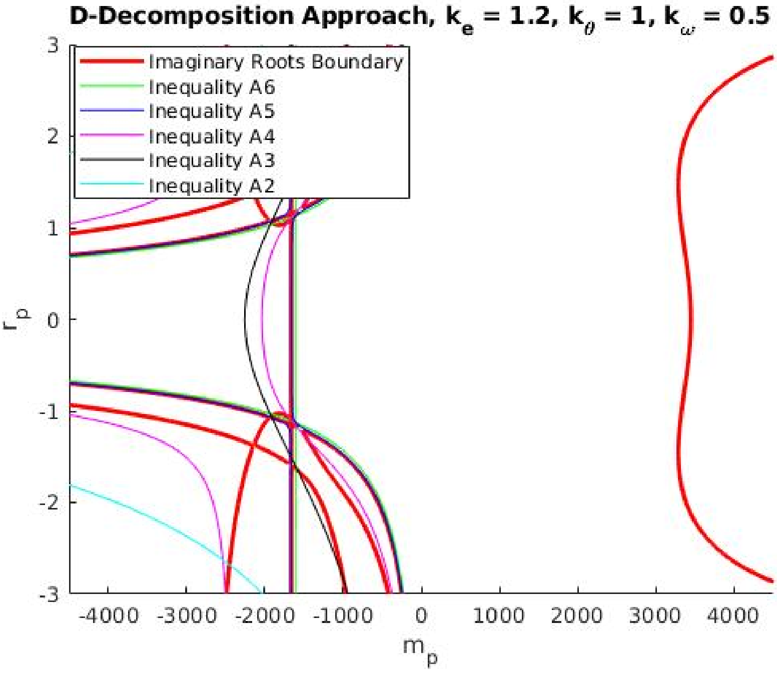


Figure 4.1: Mass - Radius D Decomposition

As shown, the plot is split into two feasible regions which would be further sampled to see if either one of them are stable.

In most cases, it has been observed that if the sample is deep enough in the stable region which identifying the gains, most often a 30% variation in mass wouldn't affect the stability of the system.

However, if we take the scenario where the gains are chosen close to the boundary, as shown below

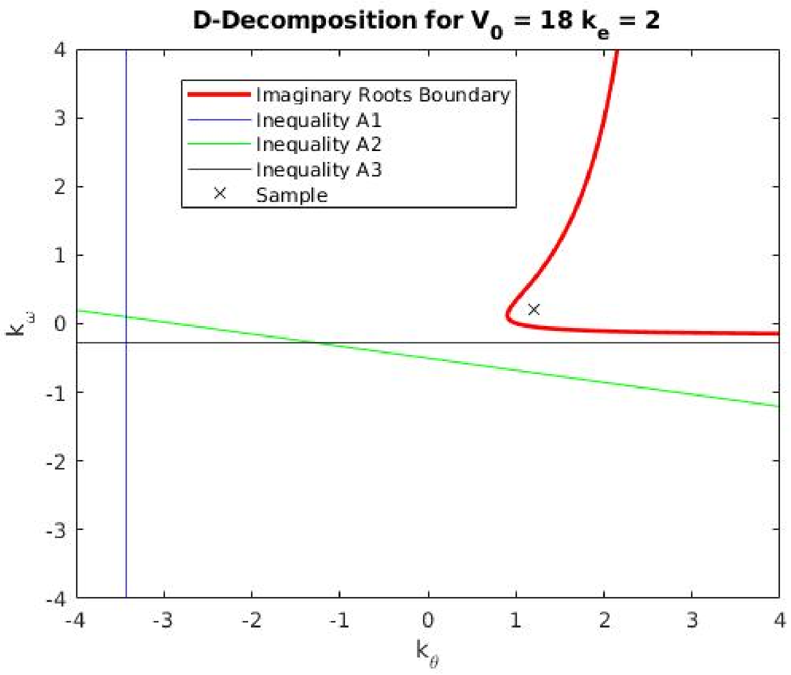


Figure 4.2: Gain Parameter Space

The sample gain values are taken quite close to the boundary. Now, if there is an addition of over $500kg$ of cargo and passengers, the car would become unstable at the given velocity of $18m/s$.

Shown below is the Mass-Radius D Decomposition

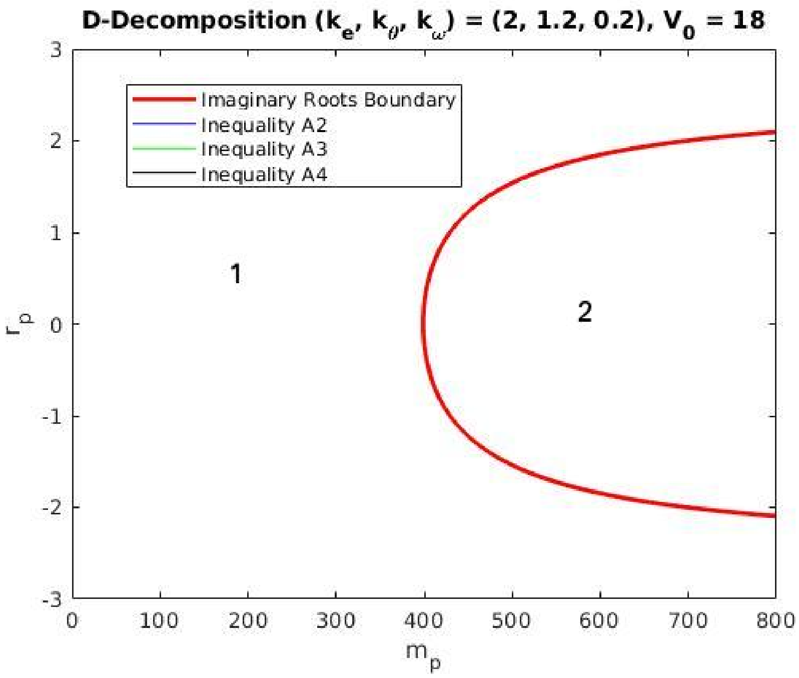


Figure 4.3: Mass Radius Parameter Space

As shown, the unstable region is reached when there is an addition of $500kg$. One way to mitigate this is to use the same D-Decomposition approach, but have the longitudinal velocity V_0 as the parameter with the same gain values and the new mass and moment of inertia (radius of gyration) values.

Additional Mass	**Max Velocity**
0	20.13
250	18.70
500	17.57

The trend of how the maximum stable velocity changes, as we add more mass to the vehicle is shown above. Therefore, if faced with the scenario of driving with the same gains,but with more than expected increase in mass, the vehicle can still perform in a stable manner albeit at reduced speeds.

5. SUMMARY AND CONCLUSIONS

To summarize,

- A combined lateral and longitudinal controller that is string stable is designed for the given vehicle parameters. The stabilizing gains for the lateral controller are identified using a D-Decomposition approach.

- Assuming a homogeneous vehicle platoon, we look at 2 different Vehicle-to-Vehicle architectures that are designed for an urban environment. The vehicles are assumed to be having access to the reference trajectory and the relative lateral and heading errors are computed with respect to the reference trajectory.

- Different simulators were tried out, and in the end, it was decided that the TORCS simulator in conjunction with Simulink would be the one that would be used, due to its ease in prototyping. A more complete simulation can be tested with Gazebo, before testing it in the actual vehicle.

- Simulation results show that the lead vehicle information is necessary to maintain the stability of the platoon. This helps in decoupling the lateral dynamics from one vehicle to another. As a result, proving individual stability in the lateral sense is sufficient to show that the string is stable.

- An analytical method for determining if variation in a few parameters due to the addition of passengers and cargo would affect the stability of the vehicle was presented. The characteristic polynomial had the required parameters coupled, therefore a different approach utilizing the resultant of two polynomials was presented.

- Based on these variations, if it is noticed that the vehicle might exhibit unstable behaviour, a limit on the maximum allowable longitudinal speed is computed.

5.1 Future Work

- The simulation could be further improved by taking into account the delays in communication and actuation lags. Although, due to the buffer nature of the V2V communication a effect on the lateral stability isn't expected, longitudinal stability is definitely expected to be affected.

- Sensor models could be attached which add in Gaussian noise to the sensor values, thereby making the simulation high-fidelity.

- One important aspect that would definitely affect the stability is packet losses in communication. This is one study that could lead to interesting results

- While it is proved that the system is stable, currently a limit on the maximum lateral error iss not proved yet. There has been recent works trying to use the advantages of a fixed structure controller (for proving stability) and provide new ways to restrict the maximum lateral error

REFERENCES

[1] Z. Shiller and S. Sundar, "Emergency lane-change maneuvers of autonomous vehicles," *ASME. J. Dyn. Sys., Meas., Control.*, pp. 37–44, 1998.

[2] A. Papadoulis, M. Quddus, and M. Imprialou, "Evaluating the safety impact of connected and autonomous vehicles on motorways," *Accident Analysis Prevention*, vol. 124, pp. 12 – 22, 2019.

[3] S. Darbha, S. Konduri, and P. R. Pagilla, "Benefits of v2v communication for autonomous and connected vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 5, pp. 1954–1963, 2019.

[4] D. Swaroop and J. K. Hedrick, "String stability of interconnected systems," *IEEE transactions on automatic control*, vol. 41, no. 3, pp. 349–357, 1996.

[5] O. McAree and S. M. Veres, "Lateral control of vehicle platoons with on-board sensing and inter-vehicle communication," *2016 European Control Conference (ECC)*, pp. 2465–2470, 2016.

[6] S. Solyom, A. Idelchi, and B. B. Salamah, "Lateral control of vehicle platoons," *2013 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 4561–4565, 2013.

[7] R. Rajamani, S. B. Choi, B. Law, J. Hedrick, R. Prohaska, and P. Kretz, "Design and experimental implementation of longitudinal control for a platoon of automated vehicles," *J. Dyn. Sys., Meas., Control*, vol. 122, no. 3, pp. 470–476, 2000.

[8] M. Liu, S. Rathinam, and S. Darbha, "Lateral control of an autonomous car with limited preview information," *2019 18th European Control Conference (ECC)*, pp. 3192–3197, 2019.

[9] M. Liu, S. Rathinam, and S. Darbha, "Lateral control of a convoy of vehicles with a limited preview information," *Proc. SPIE 11413, Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications II, 114131M (19 May 2020)*, 2020.

[10] J.-J. E. Slotine, W. Li, *et al.*, *Applied nonlinear control*, vol. 199. Prentice hall Englewood Cliffs, NJ, 1991.

[11] D. Swaroop and S. M. Yoon, "The design of a controller for a following vehicle in an emergency lane change maneuver," *UC Berkeley: California Partners for Advanced Transportation Technology. Retrieved from https://escholarship.org/uc/item/20v299zx*, 1999.

[12] B. W. Eric Espié, Christophe Guionneau, "The open source racing car simulator," *http://torcs.sourceforge.net/*.

[13] B. Wymann, "Torcs robot tutorial," *http://www.berniw.org/tutorials/robot/tutorial.html*.

[14] E. Onieva, D. A. Pelta, J. Alonso, V. Milanes, and J. Perez, "A modular parametric architecture for the torcs racing engine," *2009 IEEE Symposium on Computational Intelligence and Games*, pp. 256–262, 2009.

[15] J. Muñoz, G. Gutierrez, and A. Sanchis, "A human-like torcs controller for the simulated car racing championship," *Proceedings of the 2010 IEEE Conference on Computational Intelligence and Games*, pp. 473–480, 2010.

[16] D. Loiacono, A. Prete, P. L. Lanzi, and L. Cardamone, "Learning to overtake in torcs using simple reinforcement learning," *IEEE Congress on Evolutionary Computation*, pp. 1–8, 2010.

[17] Owen, "Torcslink," *https://github.com/VerifiableAutonomy/TORCSLink*.

[18] D. Swaroop and J. K. Hedrick, "Constant spacing strategies for platooning in automated highway systems," *ASME. J. Dyn. Sys., Meas., Control. September 1999; 121(3)*, pp. 462–470, 1999.

[19] L. Xiao, S. Darbha, and F. Gao, "Stability of string of adaptive cruise control vehicles with parasitic delays and lags," *2008 11th International IEEE Conference on Intelligent Transportation Systems*, pp. 1101–1106, 2008.

[20] M. Liu, *Lateral Control for Autonomous and Connected Following Vehicles with Limited Preview Information*. PhD thesis, 2020.

[21] I. L. De Guevara, J. Muñoz, O. De Cózar, and E. B. Blázquez, “Robust fitting of circle arcs,” *Journal of Mathematical Imaging and Vision*, vol. 40, no. 2, pp. 147–161, 2011.

APPENDIX A

LINK TO SIMULINK MODELS

The Simulink models can be found in the following link:

https://drive.google.com/drive/folders/
1PsYqav2BooAqn_W43pMs0SfP_pAX2yhF?usp=sharing