



**Michigan
Technological
University**

Michigan Technological University
Digital Commons @ Michigan Tech

Dissertations, Master's Theses and Master's Reports

2020

VEHICLE DYNAMICS MODELING FOR AUTONOMOUS DRIFTING AND CLOTHOID BASED WAYPOINT INTERPOLATION

Shubham Vatsa

Copyright 2020 Shubham Vatsa

Follow this and additional works at: <https://digitalcommons.mtu.edu/etdr>



Part of the [Navigation, Guidance, Control, and Dynamics Commons](#)

VEHICLE DYNAMICS MODELING FOR AUTONOMOUS DRIFTING AND
CLOTHOID BASED WAYPOINT INTERPOLATION

By

Shubham Vatsa

A REPORT

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

In Mechanical Engineering

MICHIGAN TECHNOLOGICAL UNIVERSITY

2020

© 2020 Shubham Vatsa

This report has been approved in partial fulfillment of the requirements for the Degree of MASTER OF SCIENCE in Mechanical Engineering.

Department of Mechanical Engineering - Engineering Mechanics

Report Advisor: *Dr. Jeffrey D. Naber*

Committee Member: *Dr. Darrell L. Robinette*

Committee Member: *Dr. Steven Ma*

Department Chair: *Dr. William W. Predebon*

Contents

List of Figures	ix
List of Tables	xiii
Acknowledgments	xv
List of Abbreviations	xvii
Abstract	xix
1 Introduction	1
1.1 Taxonomy of Autonomy	3
1.2 Decision-Making Hierarchy	4
1.3 Report Overview	7
1.4 Objective	8
1.5 Report Outline	10
I Clothoid curves	11
2 Clothoid equations generation	12

2.1	Vehicle Kinematics Model	13
2.2	Clothoids	15
2.3	Problem formulation	19
3	Results	23
3.1	General cases	23
3.2	Waypoint path tracking	27
4	Future work	29
 II Vehicle Dynamics Modeling		 33
5	Motivation and Background	34
5.1	Motivation	34
5.2	Vehicle Dynamics	36
5.2.1	Axes system and definitions	36
5.2.2	Vehicle parameters	38
5.3	Drift modeling	39
6	Modeling	41
6.1	Vehicle modeling	41
6.2	Tire modeling	44
6.2.1	Tire Brush Model	45
6.2.2	Combined slip tire model	47

7	Simulation Environment model	49
7.1	Complete drifting model	49
7.2	Tire model block	51
7.2.1	Tire slip angle block	51
7.2.2	Front tire block	52
7.3	States block	53
8	Results	55
8.1	Drift equilibrium analysis	56
8.1.1	Inference of Equilibrium analysis	58
8.2	Controller design considerations	60
9	Future work	61
10	Conclusion	63
	References	65
A	Tire parameter determination	69
A.1	Pacejka model vs Neural Networks	69
A.2	IMU sensor calibration, and scale and bias matrix	71

List of Figures

1.1	The autonomous Chevy Tahoe, 2007 DARPA Urban Challenge winner, Carnegie Mellon University[1][2]	2
1.2	Taxonomy of Autonomy[3]	5
1.3	Illustration of hierarchy in decision-making process[4]	6
1.4	Autonomous test vehicle AV-2	9
2.1	Bicycle model [5]	14
2.2	Clothoid segment. If the curvature is considered to be a signed quantity it forms a double spiral	16
2.3	Clothoid as a solution to the elasticity problem [6]	18
3.1	Transition curves	24
3.2	Path tracking over an intersection	25
3.3	Path tracking over curved paths	26
3.4	Path planning along 8 test cases[7]	27
3.5	Clothoids on APSRC test path	28

4.1	Clothoids on APSRC test path with areas of improvement in the path length	30
4.2	Optimized Clothoid interpolation on APSRC test path. It can be noticed that a trial-and-error optimization has produces good results when compared to Fig 4.1	31
5.1	VW Polo R WRC drifting [8]	35
5.2	Coordinate system to define vehicle motion [9] [10]	36
5.3	Walking analogy of tire slip[11]	38
5.4	Drift equilibrium analysis using phase portraits. The three equilibria are denoted by red dots[9]. 'r' is the yaw rate and ' β ' is sideslip angle	40
6.1	Schematic of a bicycle model[9].	42
6.2	Tire brush model components and assumptions taken in the model [9]	46
6.3	Schematic of friction circle constraint in the (x,y) plane[12].	48
7.1	Complete drift model in Simulink	50
7.2	Vehicle dynamics block containing tire model and vehicle states	50
7.3	Tire model block	51
7.4	Rear tire slip angle block	51
7.5	Front tire slip angle block	52
7.6	Front tire block	52
7.7	Vehicle states block	53

7.8	Beta state block	53
7.9	Yaw rate state block	54
7.10	Velocity state block	54
8.1	Equilibrium yaw rate vs steering angle at $U_x = 4\text{m/s}$ [9][13]	56
8.2	Equilibrium sideslip versus steering angle at $U_x = 4\text{m/s}$ [9][13]	57
8.3	Equilibrium longitudinal drive force versus steering angle at $U_x = 4\text{m/s}$ [9][13]	57
A.1	Pacejka model vs machine learning model	70

List of Tables

5.1 Vehicle parameters of AV-2 vehicle 38

Acknowledgments

First of all, I would like to thank my advisor, Dr. Jeffrey D. Naber, for allowing me to work under his guidance. He has supported me throughout my research with exceptional understanding from providing the workspace and equipment to timely reviewing my work and giving suggestions for improvement.

I would extend my gratitude to the committee members Dr. Darrell Robinette and Dr. Steven Ma, for taking out time out of their busy schedules to participate in my report defense and providing valuable counsel.

Further, I would pay my special thanks to Dr. Ahammad Basha Dudekula, who introduced me to the realm of autonomous vehicles and helped me develop a considerable interest in them. He has been the first point of contact for technical support and experimentation. Also, I would thank Shuvodeep Bhattacharjya, MS Mechanical, for being a supportive teammate.

Lastly, I would thank my friends and family; without their encouragement, I would not have finished my work.

List of Abbreviations

ODER	Object and Event Detection
ODD	Operational Design Domain
LIDAR	Light Detection and Ranging
RADAR	Radio Detection and Ranging
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
ACC	Adaptive Cruise Control
LKA	Lane Keeping Assist
DARPA	Defense Advanced Research Projects Agency
ABS	Anti-lock Braking System
ESC	Electronic Stability Control
CG	Center of Gravity
RWD	Rear Wheel Drive
AWD	All Wheel Drive
MPC	Model Predictive Controller

Abstract

The advent of autonomous vehicles necessitates a redefinition of road safety regulations, considering a controller can possess better driving skills than an average person. The work presented here partly focuses on a vehicle dynamics model development to help imitate and control vehicle drifting maneuvers. As we see, a professional driver drifting through the traffic while keeping the car safe, it can be utilized to avoid accidents at high speeds, if required. Although drifting can produce higher yaw rates than the regular driving regime, these control capabilities have not yet been exploited in the current automotive control systems. Therefore, this report focuses on developing a vehicle dynamics model to simulate the drifting of an autonomous vehicle that utilizes this high yaw rate property. Drifting control capabilities will increase the vehicle's ability to avoid collisions scenarios where higher than typical yaw rates are required.

The other part of the report uses vehicle kinematics equations to generate feasible path planning algorithms for any autonomous vehicle. If we constraint these vehicle kinematics equations for linearly varying curvature of the path, they are called Clothoid curves. This linearly varying curvature is analogous to having a continuous lateral acceleration on the vehicle while cornering, i.e., avoiding jerks. Here, Clothoids are used for the interpolation of waypoints along the path. A mission planner defines

waypoints a few meters apart from each other; then, GPS coordinates are used to check whether the vehicle is following these waypoints correctly. Therefore, waypoint interpolation is required for continuous feed of track coordinates to the controller to make faster corrections for the cross-track error and not to wait every time for a low rate GPS signal. Also, the Clothoid curves are used in road construction, thus also provides the ability to create a road model along with a vehicle model which can further be used for better state estimation.

Chapter 1

Introduction

For the last few decades, we have noticed a steady increase in research in driverless technology. Indeed, a fleet of reliable autonomous vehicles can revolutionize mobility, road safety while dramatically reducing the cost of driving. Therefore, it is among the most talked-about technologies in both the industry and academia today. The recent advancements in sensor technologies and increase computing capabilities have fueled these developments, together with the perceived transportation and societal benefits: In 2018, there were 33,654 fatalities with 1.9 million injuries in the United States as per the NHTSA, 2018 crash data findings[14]. Autonomous vehicles have the potential to drastically reduce the number of crash incidences attributed to driver errors. They will provide a means of personal transport to people with physical and visual impairment, pets, and children. Along with that, autonomous vehicles would

also facilitate more productive commute time while reducing the driving stress due to daily transit [4].



Figure 1.1: The autonomous Chevy Tahoe, 2007 DARPA Urban Challenge winner, Carnegie Mellon University[1][2]

An Autonomous Vehicle is a robot carrying passengers and navigating through the traffic. The above picture in Fig 1.1 is from the 2007 DARPA Urban Challenge, it appropriately describes the extent of sensors required to make a car autonomous. The on-board computer in a Self-Driving car performs the driving task governed by several algorithms performing a decision-making task, which takes inputs from the perception sensors and give output to the vehicle controllers. The driving task has three major components on a high level:

1. **Perception:** It includes the perception of the environment surrounding the vehicle. It is a crucial step to determine the precise vehicle location and identify driving threats and safety issues along the drivable path.

2. **Planning:** This step takes the input from the perception sensors for environmental awareness and user for the destination requirements to layout a map that is required to follow.
3. **Vehicle motion control:** At the planning step, the path map is also broken down into longitudinal and lateral motion inputs. Now, the controller is required to follow this reference input and guide the vehicle along the desired path.

In this chapter, we will first discuss the vehicle taxonomy in a bit more detail. Then, we will breakdown the decision-making task at the systems level and will describe the hierarchy of the decision-making process. Then, we will go through an overview of the research conducted for this report.

1.1 Taxonomy of Autonomy

Although Autonomy seems a technology of the future, almost every vehicle on the road exhibits some level of Autonomy. The levels of Automation in an autonomous vehicle is based on the driving task and operating limits of the on-board computer.

Let's define a few terms first:

† **Longitudinal Control:** Acceleration and braking control.

† **Lateral Control:** Steering control.

† **OEDR:** Object and Event Detection and Response, it is the ability of the vehicle to detect and react to the immediate events that can affect the driving plan. It includes driver alert systems and responses based on surroundings generally using short-range RADARs.

† **ODD:** Operational Design Domain, it constitutes the functional limits under which the system is designed to operate. It includes environmental characteristics, time of day, roadways, etc. under which a system can operate reliably.

† **Fallback:** It is defined as handling of the vehicle under emergencies. The system here could guide the car to a minimum risk condition if the driver did not intervene on time.

Based on the extent of driving control distributed between the car computer and human driver SAE has structured a Taxonomy of Autonomy in SAE J3016 standard as briefly summarized in Fig 1.2. The extent of autonomy is scaled from 0 to 5, where Level 0 represents no automation, and Level 5 is for full automation [15].

1.2 Decision-Making Hierarchy

The decision-making process in a driverless car is hierarchical, as shown in Fig 1.3. First, a route is planned considering the road network from the start point to the










	Level 1	Level 2	Level 3	Level 4	Level 5
Lateral Control					
Longitudinal Control	OR 	AND			
OEDR					
Fallback					
ODD	Limited				∞
Examples	ACC LKA	GM Super Cruise	Audi A8 Sedan	Waymo Car	...

Figure 1.2: Taxonomy of Autonomy[3]

destination, which is pre-fed into the system. The shortest route is determined by minimizing the cost function with edges weights along the path. This mission planner followed by behavioral planning that navigates through the selected course considering driving conventions and rules while interacting with other traffic participants. When the behavioral layer generates motion specifications, then the motion planner translates those specifications into a trajectory or reference path that is fed to a low-level controller. It ensures the trajectory is feasible to track by the vehicle, comfortable to the passengers, and free from the obstacles detected by the perception sensors. At the last step, the reference trajectory is executed using a feedback controller. A control system also reacts to the error accumulation and generates suitable actuators inputs to the steering, throttle, and brakes. This report deals with the route planning and motion planning steps of the decision making. A detailed outline of the report is provided in the next section.

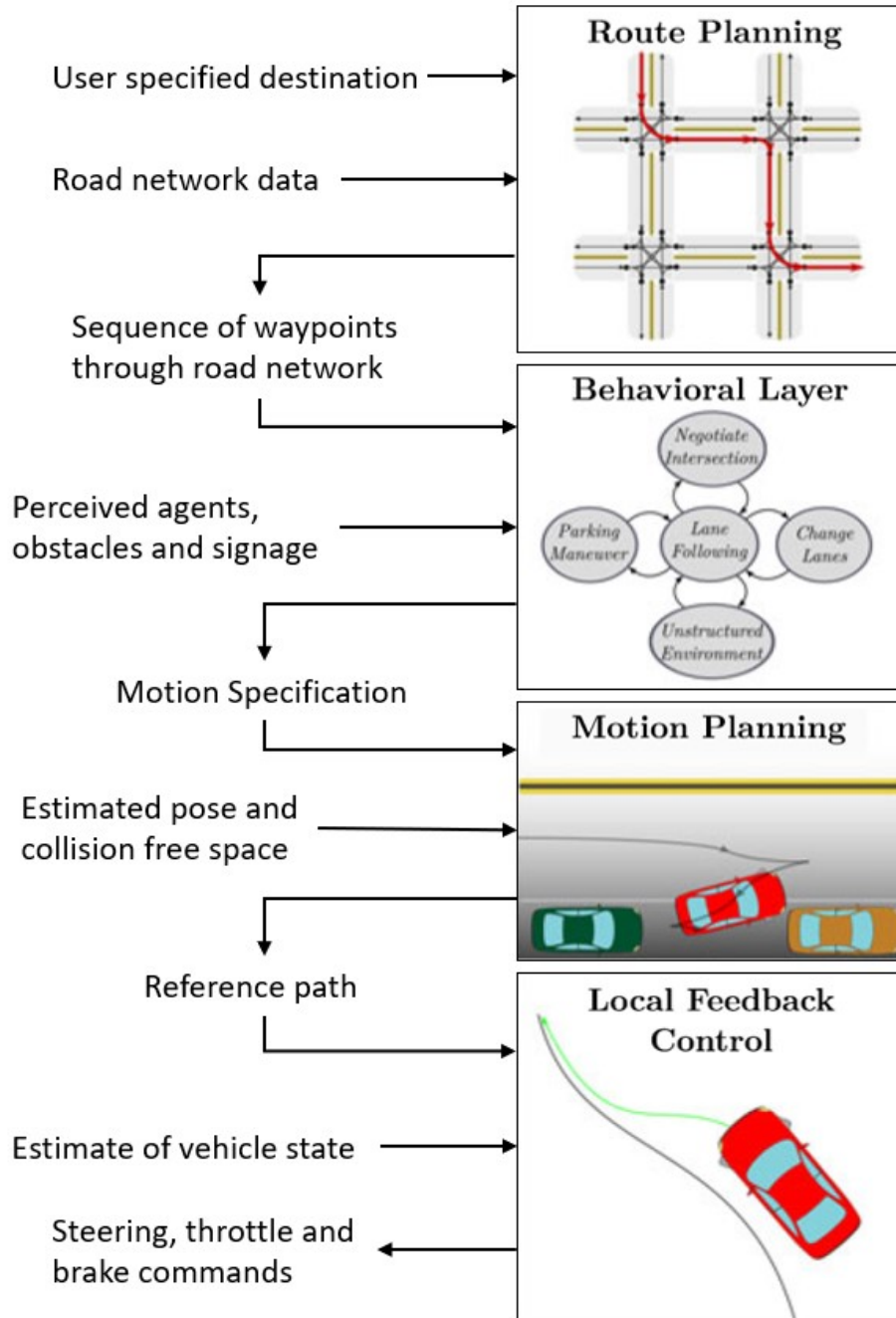


Figure 1.3: Illustration of hierarchy in decision-making process[4]

1.3 Report Overview

Building an autonomous vehicle is a broad subject and requires multidisciplinary engineering collaboration. Although the inherent governing algorithms are inspired by robotics, animations, and video games, automotive engineering plays a salient role while optimizing these algorithms. Similarly, the research presented here utilizes the core physical properties of an automobile to provide inherent constraints to the autonomous vehicle control algorithms. On a high level, this report uses vehicle kinematics equations and vehicle dynamics equations separately for different applications. Hence, it is divided into two parts:

1. Development of path planning algorithms using vehicle kinematics (Clothoids).
2. Vehicle dynamics modeling for high-speed vehicle motion control and autonomous drifting.

For the first part, the underlying motivation is to have autonomous vehicles navigate independently to the desired goal. The most common problems that are needed to be addressed are path planning and motion control along the path. At any point, the vehicle should be able to generate drivable paths, and corresponding steering and throttle command, while satisfying physical vehicle limitations. Generally, the imposed path constraints are curvature smoothness, actuator limits, and position and

heading constraints at the start and the endpoints. For this purpose, Clothoids curves are used; the main advantage is the vehicle kinematics consideration, which ensures the drivability of the path and curvature smoothness.

The second part of the research develops an extensive vehicle dynamics model with a non-linear tire model intended to simulate drifting maneuvers. Even if drifting is not considered as a general use case, the effects of approximation in vehicle dynamics can produce an error in the vehicle position and orientation predictions anywhere from 3% to 9% [9]. As autonomous vehicles are getting more complex, and these errors can be crucial for vehicle controls. Although not a part of this report, vehicle roll characteristics can produce unaccounted lateral load transfer and hence can interfere in lateral acceleration measurements from the IMU sensor. With this motivation, a vehicle dynamics mathematical model is created for this research; it also utilizes a non-linear tire model along with tire slip and vehicle sideslip.

1.4 Objective

The main intention of this report is to improve the capability of our test vehicle AV-2 as shown in the Fig 1.4. While testing our vehicle, we laid out several waypoints about 5 m apart. Now, our car, similar to a typical autonomous vehicle navigates using an IMU sensor and only uses GPS data to correct its path whenever GPS data

is available. Therefore, this discrete nature of the waypoints and lower update rates of the GPS sensor make it challenging to obtain a GPS data point exactly at those waypoints, such that the correction step can be done with enough accuracy. Hence, an interpolation curve between those waypoints is required to get a reasonably continuous set of waypoints. Therefore, for this purpose, Clothoid curves are used because they take vehicle kinematics constraints into account and give a feasible driving path with smooth curvature changes.



Figure 1.4: Autonomous test vehicle AV-2

Further, the vehicle dynamics modeling part in this report is an attempt to provide a feed-forward estimator of the vehicle states during drifting. The drifting modeling and analysis is inspired from the works of Hindiyeh [9], Kaba [13], Gonzales [16], Zhang [17] and Jelavic [18]. A combination of the techniques described in the above mentioned literature are used to reproduce a drift analysis for our test vehicle, the vehicle parameters are given in Table 5.1.

1.5 Report Outline

As the report is divided into two parts, part 1 contains a path planning algorithm using Clothoid curves. Chapter 2 further describes the motivation for using Clothoids, fundamental mathematics involved in generating Clothoids curves, and problem formulation. Whereas, chapter 3 discusses the results and characteristics of the Clothoids. Chapter 4 discusses some limitations to the current analysis and future work. In part 2, chapter 5 briefly touches the concept of vehicle dynamics and develops a drifting model. Chapter 6 describes the vehicle and tire model, and chapter 7 describes the corresponding Simulink environment model. Then, chapter 8 contains the results of the equilibrium analysis of the drifts and generation of vehicle model inputs required for controlled drifting. Finally, chapter 9 lays out possible future work, and chapter 10 concludes the report. Also, as a word of caution, both parts are worked upon separately, therefore, some independence is taken while naming the symbols. Thus, both the parts can have a different symbol for the same quantity. The naming conventions are carefully explained in the first chapter of each part.

Part I

Clothoid curves

Chapter 2

Clothoid equations generation

This chapter discusses the mathematical concepts and physical interpretations of the Clothoid curve. It also contains derivations and constraints on the Clothoids from vehicle kinematics equations.

An autonomous vehicle navigates by relying hugely upon the and IMU sensor and perception sensors (LIDAR and Stereo Camera are used for lane detection). However, these perception sensors, although reliable when fused appropriately, have lower update rates of 5 - 20Hz and are generally affected by environmental agents. Therefore, state estimation algorithms use only an IMU sensor for navigation and make a correction whenever a LIDAR or GNSS data is available. Therefore, path planning becomes crucial as the vehicle navigates mostly through its INS; a good path planning

algorithm could make INS-only navigation more dependable. The Clothoids curves are based on the vehicle kinematic equations and are used in road constructions as well. Therefore, a Clothoid based path planning algorithm can generate vehicle steering and throttle map throughout the path, which can be used as an input to the open-loop navigation if required.

2.1 Vehicle Kinematics Model

The construction of an autonomous vehicle, or a robot in general, is critical while determining its optimal path. Two-dimensional operation of a robot often have 3 degrees of freedom. Longitudinal translation along the x-axis, lateral translation along the y-axis, and rotation along the z-axis, which is perpendicular to the XY-plane. A hypothetical robot can rotate on the spot with no limits of degrees of rotation and can move sideways with no restrictions. Whereas a car-like autonomous vehicle has restricted maneuverability and greatly dependent on the previous orientation of the vehicle. Although a vehicle also has 3 degrees of freedom, it has only two controls, linear and angular control. This limitation does not allow a vehicle to make sharp turns as a robot. Thus a vehicle cannot trace a path with non-differentiable characteristics; hence this makes the trajectory equations non-integrable; therefore, it is called a non-holonomic system. This non-holonomic nature adds further complexity to the path planning algorithm since even if a vehicle is controllable, it cannot perform

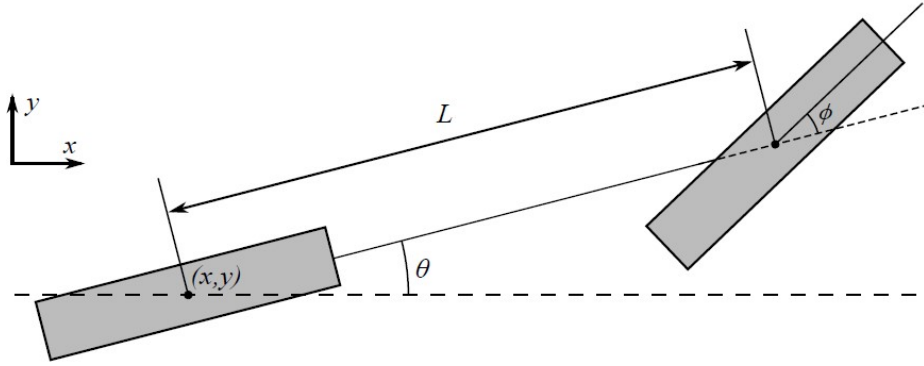


Figure 2.1: Bicycle model [5]

any arbitrary motion.

In this report, we will use a bicycle model; this model uses a single tire at each axle with stiffness and force capability of both tires in an axle “lumped” into one. It represents the vehicle kinematics appropriately in a simple model while neglecting vehicle dynamics characteristics. A bicycle model is schematically shown in the Fig 2.1. In this model, the steering angle is given as ϕ , θ is the heading angle, the rear wheel is fixed to the vehicle heading, and L is the wheelbase of the car. Another rationale for using the vehicle kinematics model is that the roads and traffic rules are designed to avoid vehicle drifting. Although the speed at some highways could be high enough to simulate the dynamic characteristics of the vehicle, the change in the yaw rate is limited to avoid a car going out of control. Therefore, almost all paved roads can be formulated using only vehicle kinematic equations.

Let $v(t)$ be the velocity at any time t , x and y be position coordinates, $\phi(t)$ be the steering angle and $\theta(t)$ be the vehicle heading angle at any time t . Therefore, the

system can be broken down into the equations as shown below:

$$\begin{aligned}
 \frac{dx(t)}{dt} &= v(t) \cos \theta(t) \\
 \frac{dy(t)}{dt} &= v(t) \sin \theta(t) \\
 \frac{d\theta(t)}{dt} &= \frac{v(t)}{L} \tan \phi(t)
 \end{aligned} \tag{2.1}$$

The Equation 2.1 is in the time-domain, however, for our purpose we require these equations in space-domain because we will be working on curvature constraint. This can be achieved easily by making following substitution $v(t).dt = ds$, assuming the $v(t) \neq 0$ and $v(t)$ is continuous i.e. mathematically analogous to the trajectory being a one-to-one function and differentiable at all the points. Equation 2.1 becomes:

$$\begin{aligned}
 \frac{dx(s)}{ds} &= \cos \theta(s) \\
 \frac{dy(s)}{ds} &= \sin \theta(s) \\
 \frac{d\theta(s)}{ds} &= \frac{1}{L} \tan \phi(s)
 \end{aligned} \tag{2.2}$$

2.2 Clothoids

A Clothoid curve is also called as Euler spiral or Cornu spiral. From Equation 2.2, let $\kappa(s) = \frac{d\theta(s)}{ds}$, where $\kappa(s)$ is the curvature as the function of path travelled. If we

limit $\kappa(s)$ as a strictly linear function of s , i.e.

$$\kappa(s) = cs + \kappa_0 \quad (2.3)$$

Then the Equation 2.3 represents a Clothoid, where c is a proportional constant. A Clothoid segment is shown in Fig 2.2, we can notice that as the path length is increasing, on both the sides, the curvature is also increasing. As discussed earlier

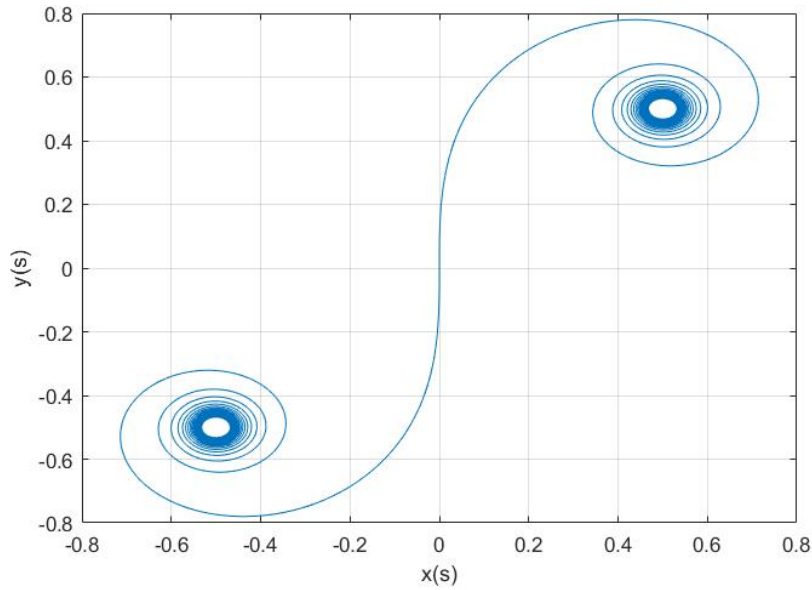


Figure 2.2: Clothoid segment. If the curvature is considered to be a signed quantity it forms a double spiral

as well, this linear characteristic in curvature is the reason why this curve describes vehicle path appropriately. It is analogous to the turning of steering wheel with a constant angular velocity. The curvature is defined as the derivative of the tangent of the steering angle [7]. Now, after making the substitutions the Clothoid equations

can be written as:

$$\begin{aligned}x(s) &= x_0 + \int_0^s \cos(\theta(s))ds \\y(s) &= y_0 + \int_0^s \sin(\theta(s))ds \\ \theta(s) &= \theta_0 + \int_0^s \kappa(s)ds\end{aligned}\tag{2.4}$$

Further, this integral representation in Equation 2.4 is called Fresnel integrals. The profusion of names indicates the fact that the linear varying curvature to arclength property has been utilized independently for different applications. Chronologically, the first discovery of the Clothoids was done by Jacob Bernoulli in 1694. The problem he was solved using Clothoids was in elasticity, the problem was: *If a cantilever spring is loaded with a mass at its free end forms a straight line. What shape it will take once the spring is unloaded?*[6]. A Fig 2.3 provides a pictorial discription of the problem stated above. Later, Augustin Fresnel in 1818, derived the well-known Fresnel integrals while working on his research in light diffraction in a single slit experiment. In an attempt to integrate all the small wavefronts to calculate the light intensity at the screen Fresnel derived the same equations. Further history behind the Clthoids can be found in [6] in more detail.

Today, Clothoids are used widely used to generate transitional curves between straight lines and circular arcs, and between two circular arcs with different radii. It has

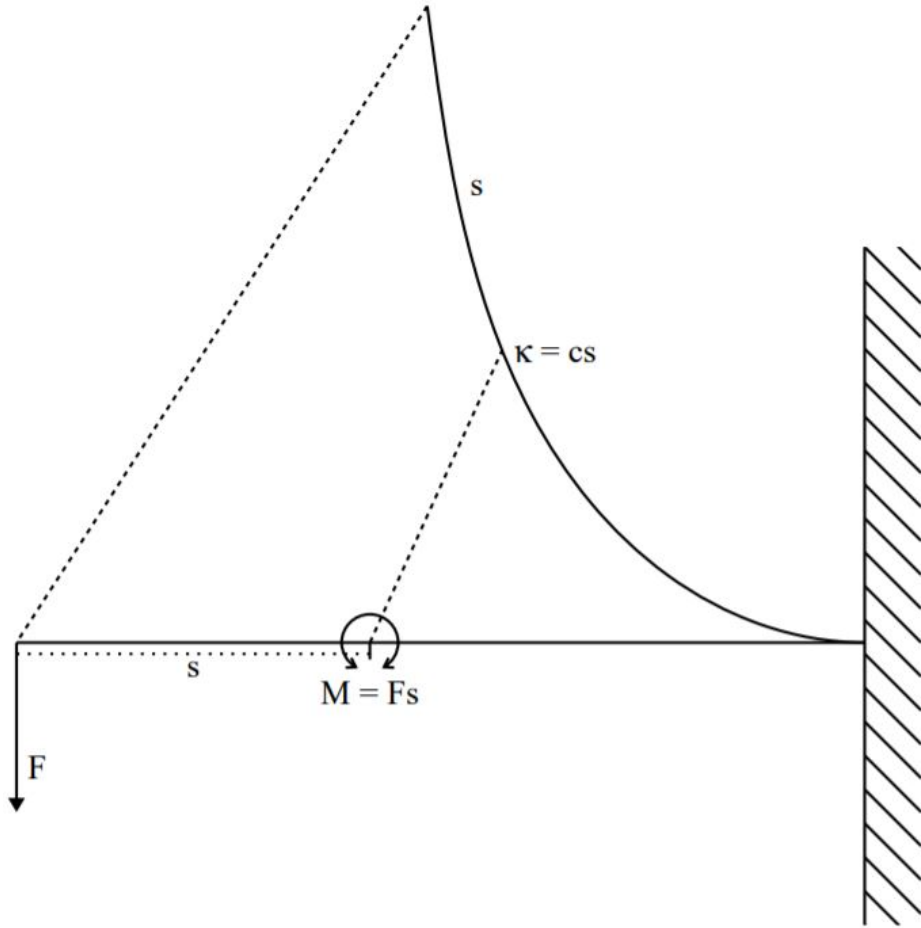


Figure 2.3: Clothoid as a solution to the elasticity problem [6]

applications in laying railway tracks, road construction, computer graphics, and animations. Clothoid based transitional curves are appealing since a smooth change in curvature leads to a continuous change in the lateral forces experienced by the passengers in the vehicle. Besides that, it prevents the vehicle from going out of control by limiting lateral forces and hence reducing the risk of accidents.

2.3 Problem formulation

We use G^1 Hermite interpolation for waypoint navigation, which is the interpolation of two points with a given tangent of direction in a plane. If the curvature is also assigned to the given points, then the interpolation will be called G^2 interpolation [7].

The problem which is addresses here is to find the Clothoid curve segment that joins two points. Let us consider (x_0, y_0) be the starting point and (x_1, y_1) be the end point, with θ_0 and θ_1 be the heading angle at the start and end respectively. First, we will derive the $\theta(s)$ from the Equation 2.4 using substitution of $\kappa(s) = cs + \kappa_0$.

$$\begin{aligned}\theta(s) &= \theta_0 + \int_0^s (cs + \kappa_0) ds \\ &= \theta_0 + \frac{1}{2}cs^2 + \kappa_0s \\ \implies \theta(s) &= \frac{1}{2}cs^2 + \kappa_0s + \theta_0\end{aligned}\tag{2.5}$$

Now, using Equation 2.5 with τ as the independent variable to avoid confusion between the variable and the integration limit. $x(s)$ and $y(s)$ can be written as:

$$\begin{aligned}x(s) &= x_0 + \int_0^s \cos\left(\frac{1}{2}c\tau^2 + \kappa_0\tau + \theta_0\right) d\tau \\ y(s) &= y_0 + \int_0^s \sin\left(\frac{1}{2}c\tau^2 + \kappa_0\tau + \theta_0\right) d\tau\end{aligned}\tag{2.6}$$

Let the path contains N waypoints which will require N-1 clothoid segments. Then Equation 2.7 represents the x and y coordinates at the i^{th} point of the path.

$$\begin{aligned} x_{i+1} &= x_i + \int_{s_i}^{s_{i+1}} \cos\left(\frac{1}{2}c(\tau - s_i)^2 + \kappa_i(\tau - s_i) + \theta_i\right) d\tau \\ y_{i+1} &= y_i + \int_{s_i}^{s_{i+1}} \sin\left(\frac{1}{2}c(\tau - s_i)^2 + \kappa_i(\tau - s_i) + \theta_i\right) d\tau \end{aligned} \quad (2.7)$$

Now, since we have derived a general case, we will now discuss the problem in hand which is to find a curve from the Equation 2.7 that satisfies:

$$\begin{aligned} x(0) &= x_0 \\ x(s) &= x_1 \\ y(0) &= y_0 \\ y(s) &= y_1 \\ (x(\dot{0}), y(\dot{0})) &= (\cos(\theta_0), \sin(\theta_0)) \\ (x(\dot{s}), y(\dot{s})) &= (\cos(\theta_1), \sin(\theta_1)) \end{aligned} \quad (2.8)$$

Where s is the total length of the segment. Now, at $i = 0$:

$$\begin{aligned} x_1 &= x_0 + \int_0^s \cos\left(\frac{1}{2}c\tau^2 + \kappa_0\tau + \theta_0\right) d\tau \\ y_1 &= y_0 + \int_0^s \sin\left(\frac{1}{2}c\tau^2 + \kappa_0\tau + \theta_0\right) d\tau \end{aligned} \quad (2.9)$$

After rearranging Equation 2.9 we get:

$$\begin{aligned}x_1 - x_0 - \int_0^s \cos\left(\frac{1}{2}c\tau^2 + \kappa_0\tau + \theta_0\right)d\tau &= 0 \\y_1 - y_0 - \int_0^s \sin\left(\frac{1}{2}c\tau^2 + \kappa_0\tau + \theta_0\right)d\tau &= 0\end{aligned}\quad (2.10)$$

Now, the Equation 2.10 is needed to be solved to obtain the required path segment. However, there are 2 equations and 3 unknowns namely, s , c and κ_0 , hence the equation cannot be solved as it is. There is another problem with solving those equations is with the term $c\tau^2$, if we closely notice that τ^2 when integrated will become the path length s . Therefore, $c\tau^2$ makes the equation non-homogeneous which is further difficult to solve. Thus, to solve the equations we will introduce some substitutions, before that, let's assume an angle ϕ such that $\phi_0 = \theta_0 - \phi$ and $\phi_1 = \theta_1 - \phi$. We will also put:

$$x_1 - x_0 = \Delta x, \quad y_1 - y_0 = \Delta y, \quad A = \frac{1}{2}c\tau^2, \quad \delta = \theta_1 - \theta_0 \quad (2.11)$$

Now, using Equation 2.11 and 2.5, we can write $\kappa_0\tau = \delta - A$. Further, we will also convert Δx and Δy into polar coordinates as: $\Delta x = r \cdot \cos(\phi)$, $\Delta y = r \cdot \sin(\phi)$ and following trigonometric identities:

$$\cos(p - q) = \cos p \cos q + \sin p \sin q, \quad \sin(p - q) = \sin p \cos q - \cos p \sin q \quad (2.12)$$

Now, using Equation 2.10 we will define $f(s,A)$ and $g(A)$:

$$\begin{aligned}
f(s, A) &= [\Delta x - \int_0^s \cos(A + (\delta - A) + \theta_0)d\tau]. \cos \phi + [\Delta y - \int_0^s (\sin(A + (\delta - A) + \theta_0)d\tau)]. \sin \phi \\
&= [r \cos^2 \phi - \int_0^s (\cos(A + (\delta - A) + \theta_0). \cos \phi)d\tau] \\
&\quad + [r \sin^2 \phi - \int_0^s \sin(A + (\delta - A) + \theta_0). \sin \phi)d\tau] \\
&= r + \int_0^s \cos(A + (\delta - A) + \phi_0)d\tau \\
&= \sqrt{\Delta x^2 + \Delta y^2} + \int_0^s \cos(A + (\delta - A) + \phi_0)d\tau \tag{2.13}
\end{aligned}$$

Similarly $g(A)$ which is a function only in A can be derived as shown in below.

Equation 2.15 is a change of limit to 1 of the Fresnel integral, the complete derivation is given in [7].

$$g(A) = \frac{1}{s} \int_0^s \sin(A + (\delta - A) + \phi_0)d\tau \tag{2.14}$$

$$= \int_0^1 \sin(A + (\delta - A) + \phi_0)d\tau \tag{2.15}$$

Now, the interpolation problem can be solved by only solving $g(A)$ and the unknowns can be calculated by reverse substitution[7].

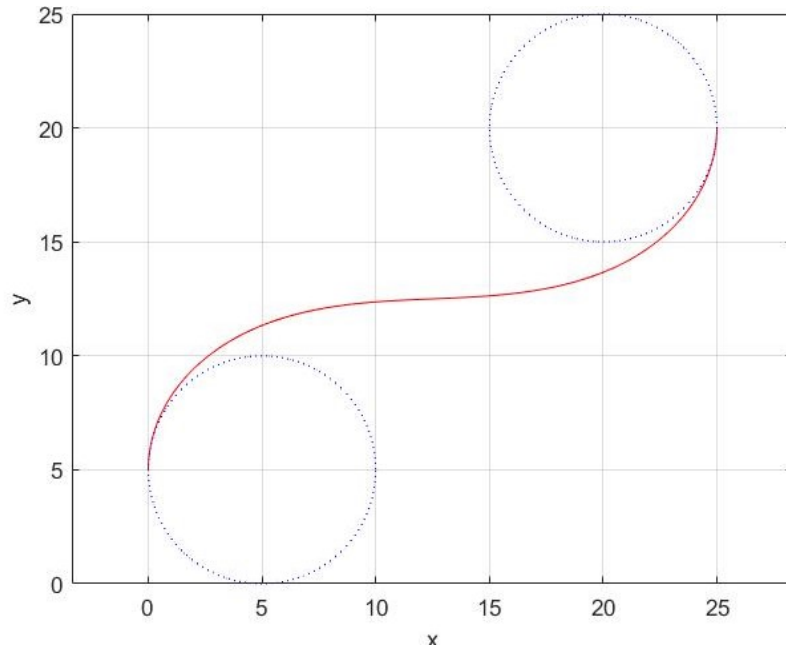
Chapter 3

Results

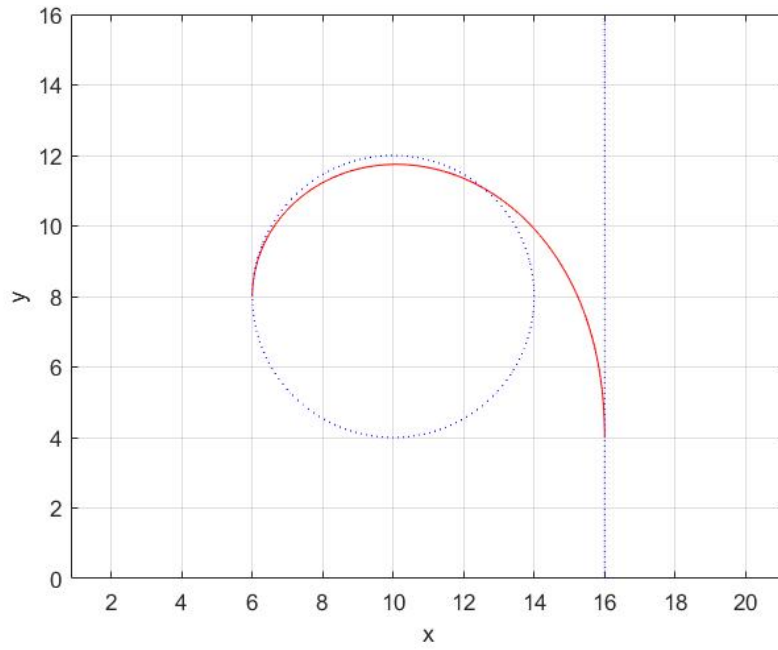
In this chapter, we will first discuss the general cases to develop a better understanding of the Clothoids working by using transition using common shapes. Then, we will present the interpolated path for our test vehicle.

3.1 General cases

The aim behind using general cases is to establish that the generated transition curves produce a smooth path with linearly varying curvature. We know that a circle has constant curvature, i.e. $(\frac{1}{Radius})$, therefore, if the vehicle is to take a turn from a straight path, its curvature will suddenly rise from 0 to $(\frac{1}{Radius})$.



(a) Transition curve connecting two circular paths



(b) Transition curve connecting straight path to circular path

Figure 3.1: Transition curves

In Fig 3.1, the red plot is the transition curve. Fig 3.1(a) shows how the Clothoid changes curvature on two different circular paths. In Fig 3.1(b) We can notice the curve matches the curvature of that circle only at the end and how the clothoid makes a smooth transition from the 0 curvature to the curvature of the circle. Further, these plots were extended to the simulated road environment, using Matlab, as shown in Fig 3.2 for 90° intersection. For this purpose, the heading angles at the start and endpoints are taken as straight-line heading to the nearest point. However, for the point at the intersection, the heading is considered as half the intersection angle.

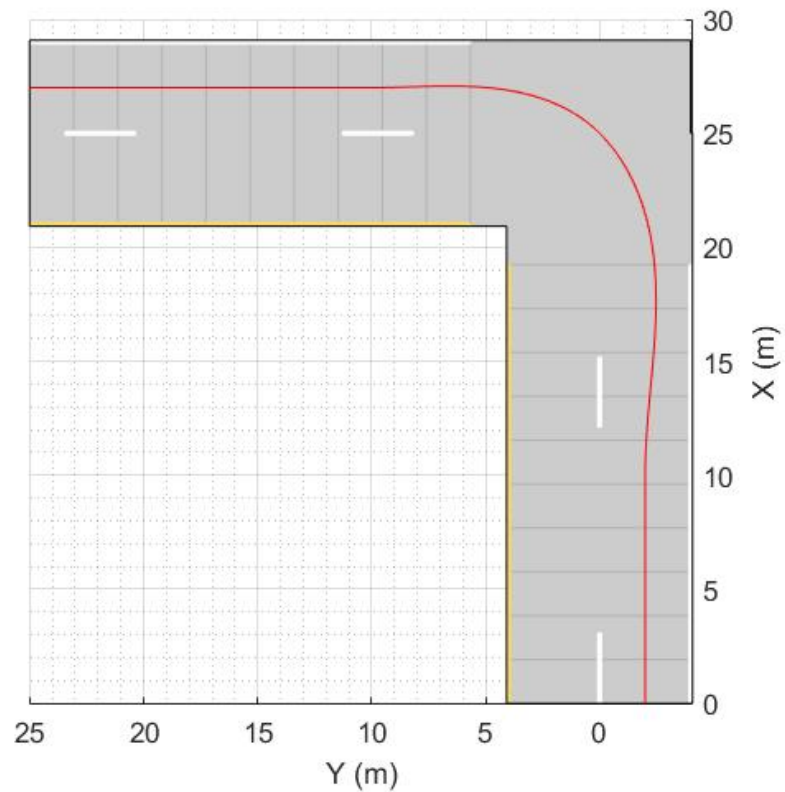
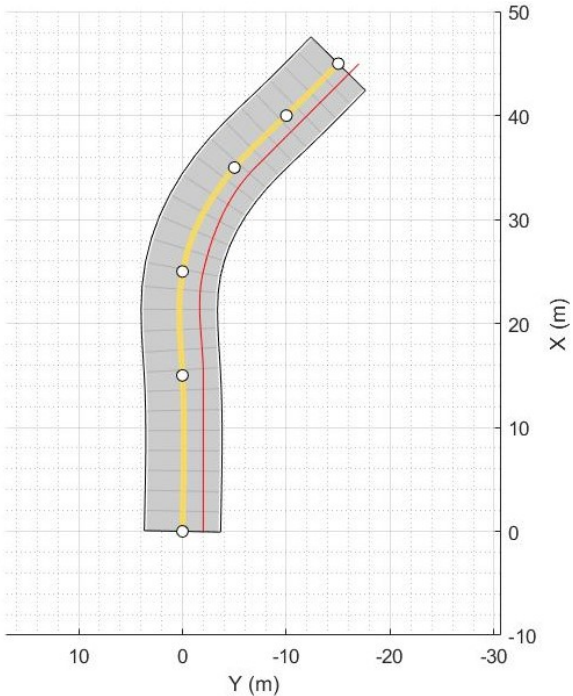
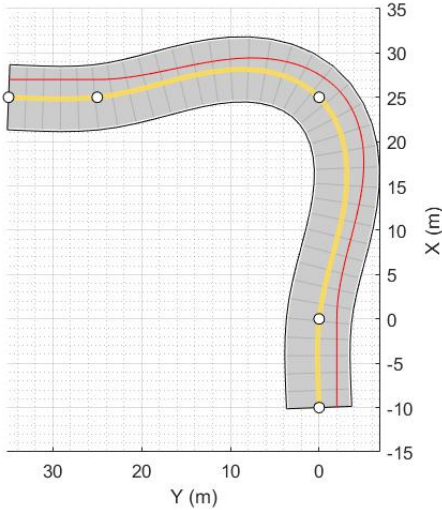


Figure 3.2: Path tracking over an intersection

Similarly, the following paths are traced in the Fig 3.3



(a) Curved path with 45° turn



(b) Curved path with 90° turn

Figure 3.3: Path tracking over curved paths

3.2 Waypoint path tracking

Before, moving to the test path tracking further analysis is carried out to check the versatility of the Clothoids. For the next test various headings variations were planned out as shown in Fig 3.4.

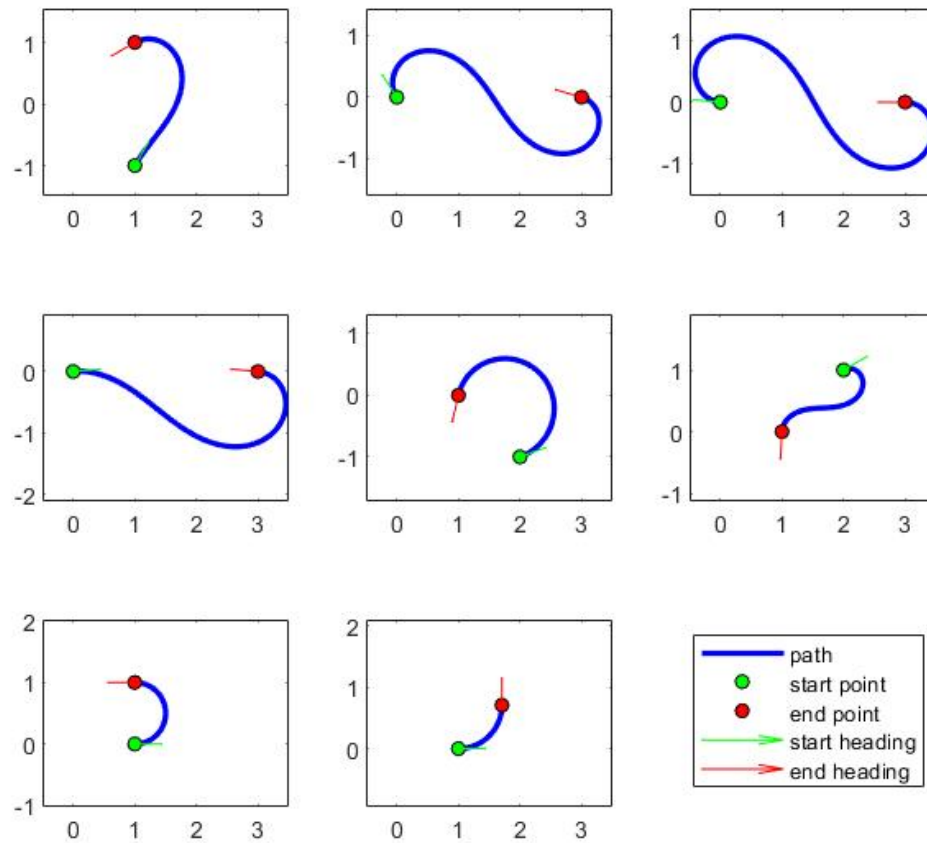


Figure 3.4: Path planning along 8 test cases[7]

Now, we can extend our curves to the test track we build at the APSRC site, as shown in Fig 3.5. The Clothoid interpolation provides waypoints at any update rate at will. This ensures that at every location on the path, the controller will have a waypoint to measure its positional error. This becomes further crucial because GPS data, which is used to measure the positional error, works at a lower frequency. Since waypoints are preset, therefore, are used to correct the cross-track error using the GPS signal. More number of waypoints ensure that whenever a GPS signal is received, the nearest waypoint can be used to correct its location and prevents the controller from making approximations.

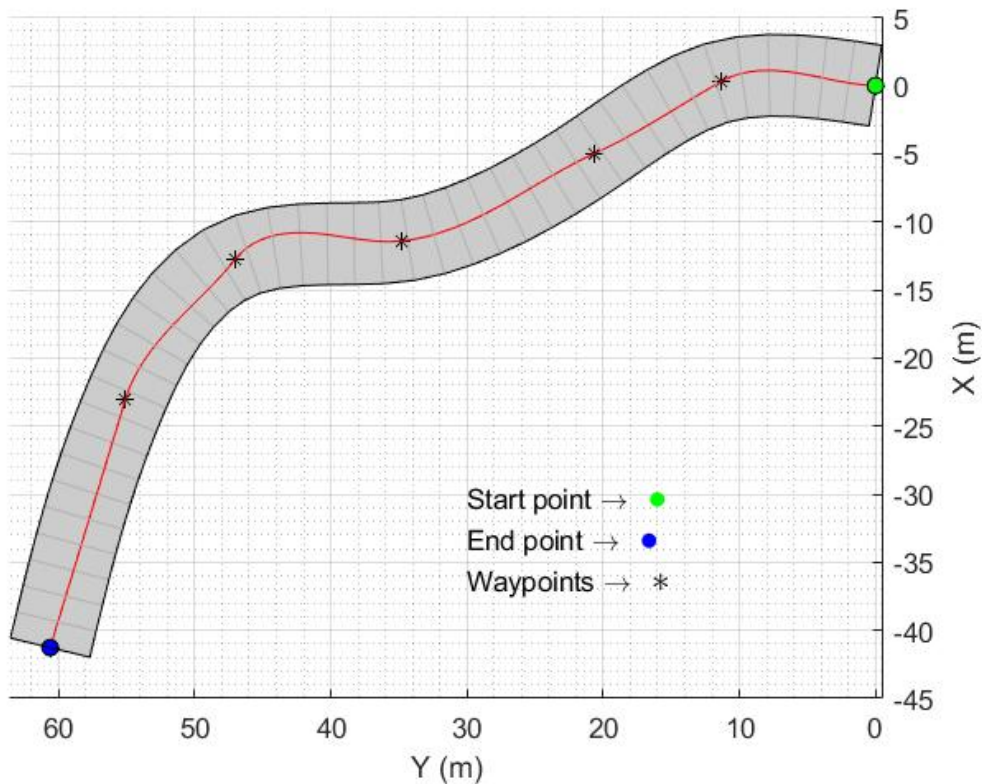


Figure 3.5: Clothoids on APSRC test path

Chapter 4

Future work

Although Clothoids provide a smooth transition between waypoints and path curvature, it may not lead to the shortest path. It is because of the hard constraint of the heading angle at the endpoint. It becomes noticeable after careful inspection and comparison with the road environment. In Fig 4.1, the orange boxes over the curve show such two instances, where to maintain endpoint heading, the path takes bigger turns.

Therefore, the future work on the topic would be to optimize the path for the shortest length with a soft constraint on the endpoint heading angle. A trial-and-error optimization is performed in our case to make the curve smoother, as shown in Fig 4.2. It was possible because of the lower number waypoints in the parent problem.

However, a generalized application optimization is required.

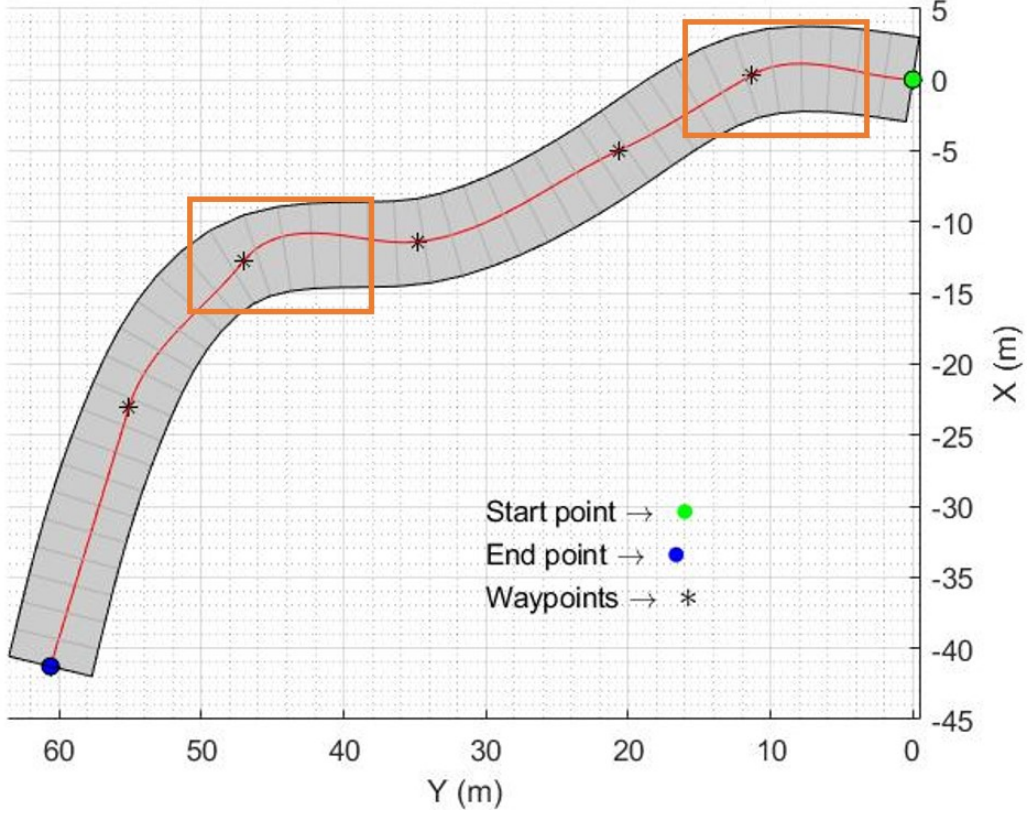


Figure 4.1: Clothoids on APSRC test path with areas of improvement in the path length

The optimization problem can be formulated as shown in Equation 4.1.

$$\begin{aligned}
 &\Rightarrow \min \sum_{i=1}^s \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \\
 s.t. \quad &x_{i+1} - x_i = \int_{s_i}^{s_{i+1}} \cos\left(\frac{1}{2}c(\tau - s_i)^2 + \kappa_i(\tau - s_i) + \theta_i\right) d\tau \\
 &y_{i+1} - y_i = \int_{s_i}^{s_{i+1}} \sin\left(\frac{1}{2}c(\tau - s_i)^2 + \kappa_i(\tau - s_i) + \theta_i\right) d\tau \\
 &\theta(s) \leq \frac{1}{2}cs^2 + \kappa_0s + \theta_0
 \end{aligned} \tag{4.1}$$

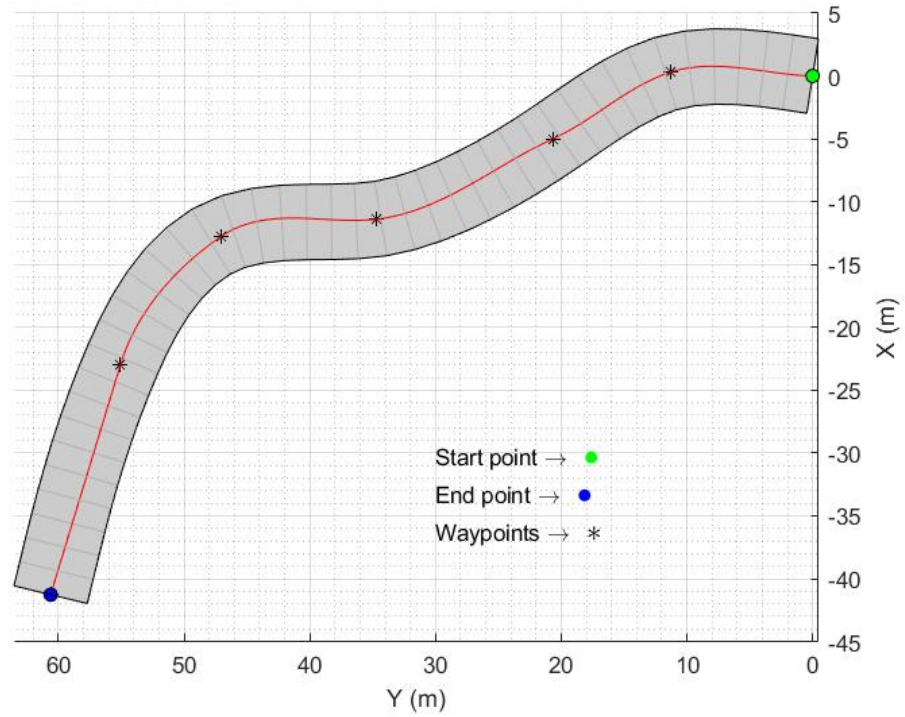


Figure 4.2: Optimized Clothoid interpolation on APSRC test path. It can be noticed that a trial-and-error optimization has produces good results when compared to Fig 4.1

Further, An LTV-MPC (Linear Time-Varying Model Predictive Controller) based approach is made in [19] on Clthoids for local path planning.

Part II

Vehicle Dynamics Modeling

Chapter 5

Motivation and Background

In this chapter, we discuss the motivation behind this research. We will also develop a background on the ways to model drift in a vehicle.

5.1 Motivation

The underlying motivation for this research is the utilization of the high yaw rate property of the drifting into vehicle safety. Conventionally, drifting, or operation of the vehicle in the tire saturation region is avoided for better control. Many technologies have been implemented into today's vehicles such as ABS and ESC as critical safety systems, with the intention to prevent the vehicle from drifting.



Figure 5.1: VW Polo R WRC drifting [8]

Whereas for an average driver tire saturation poses a driving hazard, a professional driver, in rally racing, can navigate through tight corners while drifting at very high speeds, Fig 5.1, and keep the vehicle safe throughout. Actually, at those speeds, drifting is probably the safest way to make the turn because applying hard brakes every time can lead to overheating, and eventually, Brake fade, which is another safety hazard. Therefore, pro-racing drivers go through intense training and hours of driving to be able to control this drifting. Since an autonomous vehicle controller can be programmed to handle such maneuvers with the precision of masters, this will make the roads much safer even at higher speeds.

Therefore, in this research, a vehicle dynamics model is developed, and drifting analysis is carried to provide initial inputs to the drift controller design.

5.2 Vehicle Dynamics

We will first introduce the axes system used in this research. Then we will touch some basic definitions, followed by test vehicle parameters.

5.2.1 Axes system and definitions

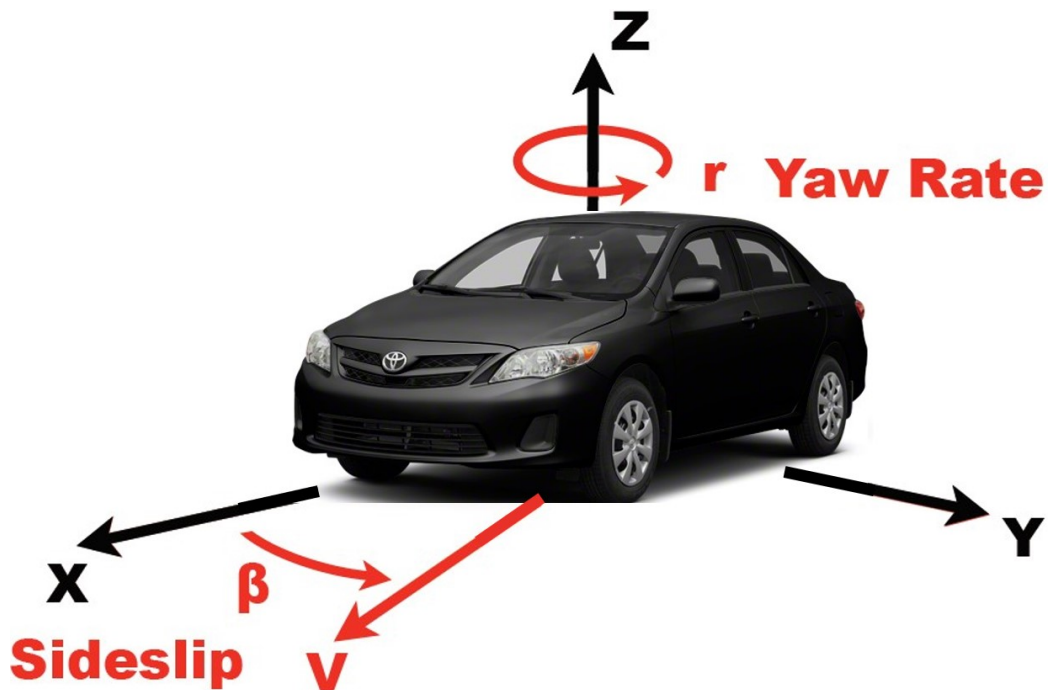


Figure 5.2: Coordinate system to define vehicle motion [9] [10]

Fig 5.2 represents the Body-fixed Coordinate system used in this research to define vehicle motion. This system complies with the SAE standard described in SAE J670e [20]. Here, the x-axis is defined as parallel to the road and longitudinal direction of the vehicle, with a positive side being the forward direction. The y-axis is perpendicular to the x-axis and parallel to the ground, with a positive direction on the left of the vehicle. And, the z-axis is perpendicular to both the axes and positive above the ground.

† **Sideslip angle:** It is the angle between the longitudinal direction of the vehicle and velocity vector, which is the instantaneous vehicle velocity.

† **Tire slip angle:** It analogically similar to the sideslip angle, however, related to the tires. It represents the angle between vehicle velocity and the heading direction of the wheel. A walking analogy is given in the Fig 5.3.

† **Tire Saturation:** It is the operational region of the tire where it cannot produce lateral force (or longitudinal force) after the saturation limit i.e., $\mu.F_z$.

† **Yaw:** It is the rotation of the vehicle about the z-axis.

† **Counter-steering:** steering the wheel opposite to the intended direction.

There are other characteristics of the tire and vehicle which are not considered here for the sake of simplicity, such as Camper angle, Caster angle, Roll, Pitch, etc. Further,

the vehicle is considered to be a rigid body; therefore, the effects due to the suspension system are ignored.

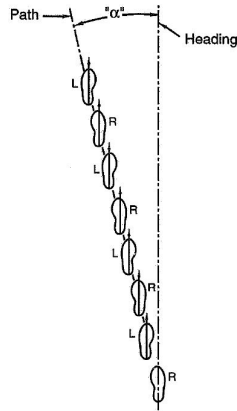


Figure 5.3: Walking analogy of tire slip[11]

5.2.2 Vehicle parameters

The vehicle parameters are the quantities that remain constant at any time.

Table 5.1
Vehicle parameters of AV-2 vehicle

Parameter	Description	Value
m	Mass	14.28 kg
I_z	Yaw inertia	4.712 kg.m ²
a	CG from front axle	0.29 m
b	CG from rear axle	0.25 m
h	CG height	0.167 m
μ	Tire friction coefficient	0.55
$C_{\alpha F}$	Front tire stiffness Coefficient	100 N/rad
$C_{\alpha R}$	Rear tire stiffness Coefficient	150 N/rad

The yaw inertia of the vehicle is calculated by taking the mass at the individual wheels. Now, the car is assumed to be a rigid body symmetric about both the planar axes with 4 point masses (tires) at each corner. Then, using the parallel axis theorem and moment of inertia formula, rotational inertia at the z-axis is calculated. Also, the CG height is calculated by keeping one axle of the vehicle at a known height and taking the weights on all the four wheels, then solving the moment equation at the center of the axle, which is at the ground.

5.3 Drift modeling

A vehicle is said to be drifting when its tires are in the saturated state and the vehicle is operating at the handling limits. Tire saturation means the tires are generating maximum force available from friction. If the rear tires saturate before the front tires, the vehicle exhibits fishtailing, as shown in the figure below. It occurs because the rear wheels are not able to produce the required turning torque. In contrast, front wheels are still able to provide necessary lateral acceleration, therefore the front wheels are closer to the turning center. If this fishtailing increases, the rear wheel will be moved even further, and the car will make a 180° turn. Conversely, if the front wheels lose traction, then the vehicle will go into limit understeer and will move out of the corner from the outer turning edge.

Intuitively, whenever a car drift, the driver can sustain the drift following some precious operations such as acceleration, steering, and countersteering. Therefore, drift can be defined as a steady-state operation, and modeled as an equilibrium with rear tire saturation. The Fig 5.4 shows that the stable equilibrium is between two unstable equilibrium.

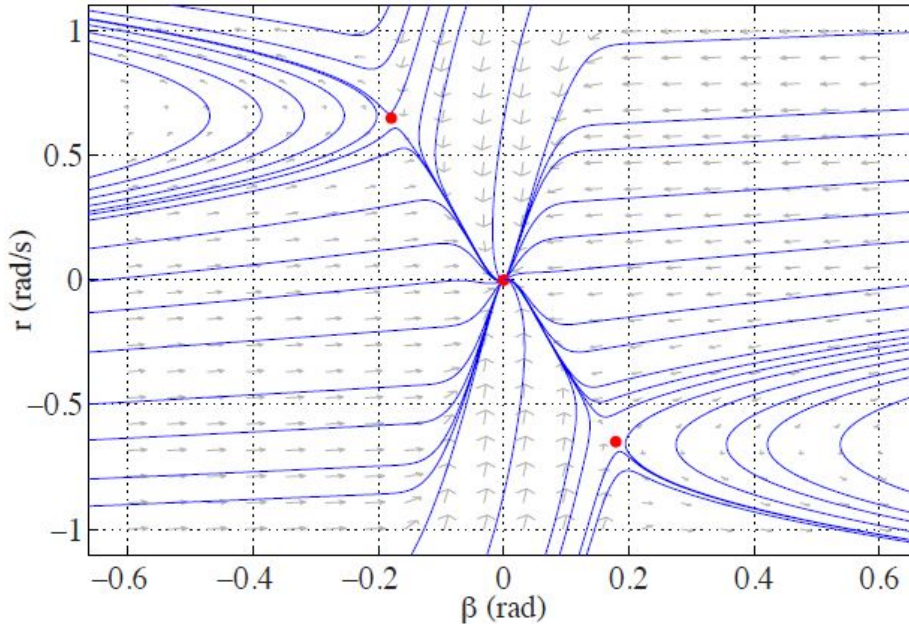


Figure 5.4: Drift equilibrium analysis using phase portraits. The three equilibria are denoted by red dots[9]. 'r' is the yaw rate and 'β' is sideslip angle

Both the unstable equilibrium are high-sideslip region, which is the region where high countersteer is required to sustain the drift. With high-sideslip, the velocity component in the lateral direction also increases; if the vehicle is accelerating, this velocity vector produces high yaw moments. High yaw moments and countersteering make those areas unstable equilibrium for drifting.

Chapter 6

Modeling

In this chapter, we will layout the mathematical and physical foundations while modeling the different states. We will define the states for our vehicle system and develop a relationship for the tire model.

6.1 Vehicle modeling

Let us discuss some intuitive understanding of the drifting maneuver. To achieve drifting, a driver initially drives the vehicle to attain a velocity, then he turns the car with significant steering input and increases the throttle abruptly. Doing this creates a large slip angle at the rear wheel, and the rear tire force saturates. Due to the

saturation, rear-wheel cannot produce more lateral force to keep the rear axle closer to the corner instead moves to the outer end of the turn, creating high sideslip angle and yaw.

From this initial analysis, we can conclude that a drifting vehicle requires two inputs: steering angle and rear longitudinal force. Therefore, a three-state model is used here instead of a regular two-state model. A three-state model allows for two inputs, although it increases another unknown that accounted fairly easily as given in 6.3.

To define the vehicle state, we have used a bicycle model, as shown in Fig 6.1. Since the bicycle model is planar, it assumes no effects from the roll and pitch motion of the vehicle. However, when the tire parameters are determined empirically, the impact of the lateral road transfer due to rolling can be captured.

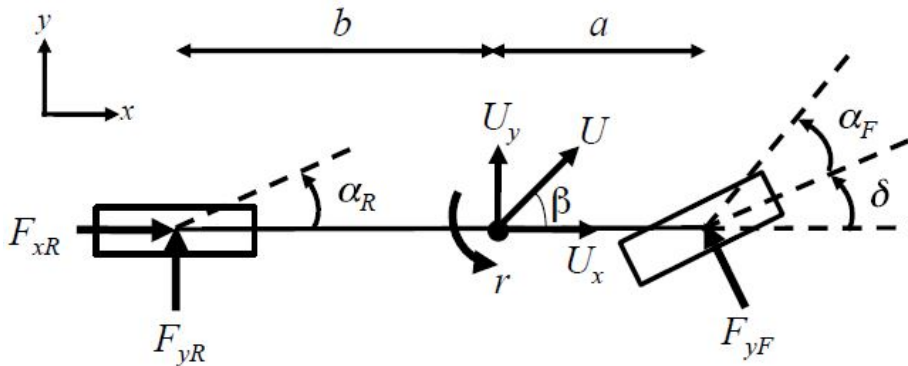


Figure 6.1: Schematic of a bicycle model[9].

We are using a 3-state version of the bicycle model, where the state are U_x , U_y and r for longitudinal velocity, lateral velocity, at the CG and yaw rate respectively. We can

easily determine these states by drawing a Free-Body Diagram and balancing forces in x and y directions, and moment in z-direction as shown in Equation 6.1.

$$\begin{aligned}
 ma_y &= F_{yF} \cos \delta + F_{yR} \\
 I_z \dot{r} &= aF_{yF} \cos \delta + bF_{yR} \\
 ma_x &= F_{xR} - F_{yF} \sin \delta
 \end{aligned} \tag{6.1}$$

There are two categories of equations used in the modeling: conservation laws and constitutive relations. Conservative laws are the properties that are conserved in the systems i.e. force balance and energy balance equations. Whereas, Constitutive relations are the relationship between the different states, for example: velocity acceleration equations. Both set of equations are required to derive the state equations. Therefore, conservation laws or the balance equations are given in Equation 6.1. And, constitutive equations given in Equation 6.2.

$$\begin{aligned}
 a_y &= \dot{U}_y + rU_x \\
 a_x &= \dot{U}_x - rU_y \\
 \beta &= \arctan \frac{U_y}{U_x} \approx \frac{U_y}{U_x}
 \end{aligned} \tag{6.2}$$

Using, Equations 6.1 and 6.2 we can derive the 3-state model as follows:

$$\begin{aligned}
 \dot{\beta} &= \frac{F_{yF} + F_{yR}}{mU_x} - r \\
 \dot{r} &= \frac{aF_{yF} - bF_{yR}}{I_z} \\
 \dot{U}_x &= \frac{F_{xR} - F_{yF} \sin \delta}{m} + rU_x\beta
 \end{aligned} \tag{6.3}$$

The reason for using sideslip angle β as a state in the model instead of lateral velocity is again for intuitive understanding. Having longitudinal and lateral velocity does not give a clear understanding of the vehicle orientation. Whereas, sideslip angle along yaw rate and longitudinal velocity provides descriptive understanding the vehicle orientation and also about its future position.

6.2 Tire modeling

Modeling of a tire is considerably challenging, owing to their complex behavior. Tires are heterogeneous due to the variety of constituents used in their construction, such as steel wires, rubber, and polymeric fibers like Kevlar and nylon. Further, different tires have different construction, rubber compounds, composition, sizes, and tread pattern. This complexity, along with the variations in tire properties due to changes in temperature, environment, treads wear, and inflation pressure, possess immense difficulty in building a model that will apply to all the tires with sufficient accuracy.

Therefore, there are different approaches for tire modeling, such as finite element analysis [21] and empirical models like Pacejka Model or "Magic Formula" [22]. Both models can represent a highly accurate tire model. However, these model very complicated and thus restricts physical intuition. Therefore, for our purpose, a slightly modified version of the relatively more uncomplicated model is used.

6.2.1 Tire Brush Model

In this work a variant of tire brush model is used as proposed in [23]. As shown in the Fig 6.2, it assumes a rigid tire ring with a rigid carcass and a rectangular contact patch of length $2l$. The net lateral generated by the tire depends on the cornering force demand and total available force from friction. Cornering force demand is dictated by the tire slip angle α . Let us assume $q_z(x)$ and $q_y(x)$ be the vertical force and demanded lateral per unit length on the contact patch, respectively. For the model used here, the pressure distribution is taken a parabolic across the contact patch, therefore, $q_z(x)$ and $q_y(x)$ are given in Equation 6.4, where 'x' is the length from the start for the contact patch and $C_{\alpha y}$ is lateral stiffness.

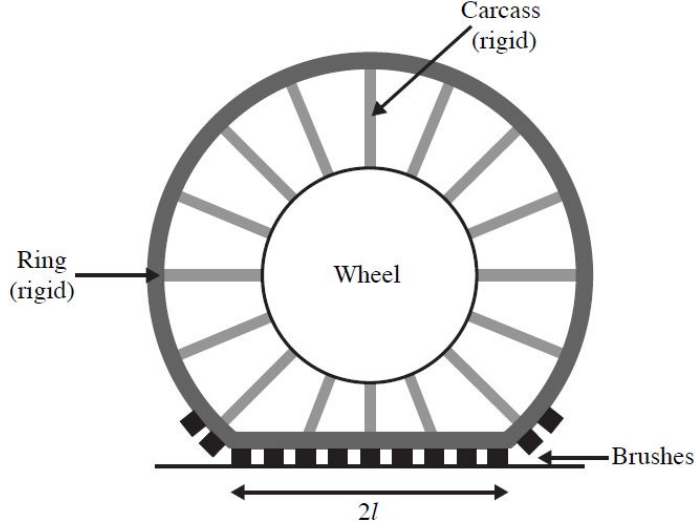


Figure 6.2: Tire brush model components and assumptions taken in the model [9]

Now, by integrating $q_y(x)$ over the length of the contact patch total F_y can be obtained [9]. The expression is shown in Equation 6.5.

$$q_z(x) = \frac{3F_z}{4l} \frac{l^2 - x^2}{l^2}$$

$$q_y(x) = -C_{\alpha y}(2l - x) \tan \alpha \quad (6.4)$$

$$F_y = \begin{cases} -C_\alpha \tan \alpha + \frac{C_\alpha^2}{3\mu F_z} |\tan \alpha| \tan \alpha - \frac{C_\alpha^3}{27\mu^2 F_z^2} |\tan \alpha|^3 & |\alpha| \leq \alpha_{sl} \\ -\mu F_z \operatorname{sgn}(\alpha) & |\alpha| < \alpha_{sl} \end{cases} \quad (6.5)$$

Where, $\alpha_{sl} = \arctan \frac{3\mu F_z}{C_\alpha}$

This model captures two major characteristics: linear dependence and tire saturation

behaviour. However, from the intuitive understanding, drifting is controlled by two inputs namely: steering and acceleration. Therefore, it becomes necessary to account for the longitudinal slip in this model which is not accounted at in the Equation 6.5.

6.2.2 Combined slip tire model

In this section, previously derived brush is extended to the combined slip model. The longitudinal wheel slip is taken as κ which is defined in Equation 6.7. Where R_w is the tire radius, ω is the wheel angular velocity, V_w is the longitudinal velocity.

$$\kappa = \frac{\omega R_w - V_w}{V_w} \quad (6.6)$$

Which gives out the longitudinal force demand to be $q_x(x)$ given in Equation 6.4

$$q_x(x) = C_{\alpha x}(2l - x) \left(\frac{\kappa}{1 + \kappa} \right) \quad (6.7)$$

Now, instead of integrating the $q_x(x)$ in the same way as in 6.5, we will utilize a more general case by using a property that the tire forces are coupled in order to obey friction constraints known as friction circle shown in Fig 6.3. The force constraint is given in Equation 6.8. Now, this constraint is used to write F_z in terms of a derating factor ξ ($0 \leq \xi \leq 1$) in Equation 6.9, where ξ is given as Equation 6.10

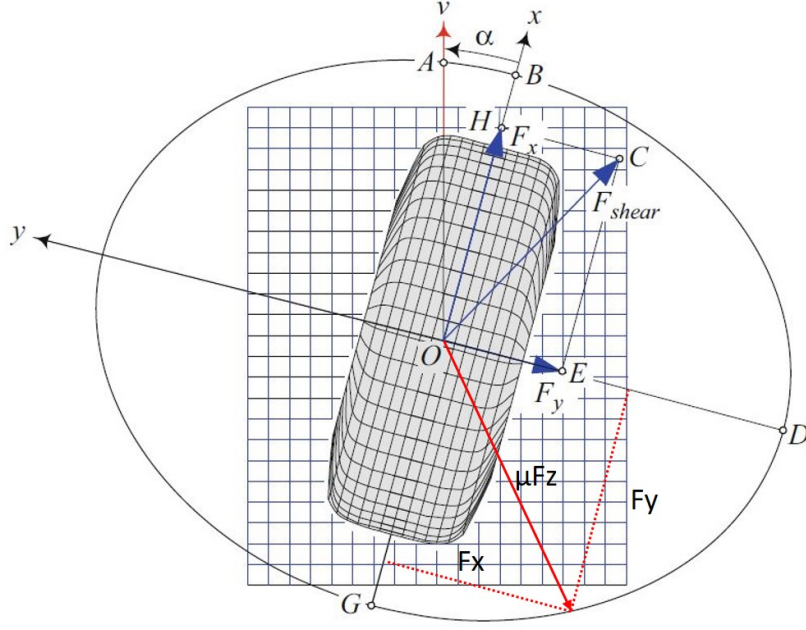


Figure 6.3: Schematic of friction circle constraint in the (x,y) plane[12].

Using the Equations in 6.8, 6.9, and 6.10 the combined tire model is derived as shown in Equation 6.11.

$$F_y^{max} = \sqrt{(\mu F_z)^2 - F_x^2} \quad (6.8)$$

$$F_y^{max} = \xi \mu F_z \quad (6.9)$$

$$\Rightarrow \xi = \frac{\sqrt{\sqrt{(\mu F_z)^2 - F_x^2}}}{\mu F_z} \quad (6.10)$$

$$F_y = \begin{cases} -C_\alpha \tan \alpha + \frac{C_\alpha^2}{3\xi \mu F_z} |\tan \alpha| \tan \alpha - \frac{C_\alpha^3}{27\xi^2 \mu^2 F_z^2} |\tan \alpha|^3 & |\alpha| \leq \alpha_{sl} \\ -\xi \mu F_z \operatorname{sgn}(\alpha) & |\alpha| < \alpha_{sl} \end{cases} \quad (6.11)$$

$$\alpha_{sl} = \arctan \frac{3\xi \mu F_z}{C_\alpha}$$

Chapter 7

Simulation Environment model

In this chapter we have build all the previously discussed equations into a Simulink model. The model also contains block of the controller which is discussed in the next chapter.

7.1 Complete drifting model

The Simulink model given in Fig 7.1 is the complete drifting model with vehicle dynamics model and controller. The switch block used at the start of the model is to switch from initial values to the simulated values after 1 time step. Further, the controller is designed in a Matlab script, and vehicle parameter block is script which

contains all the parameter values in structure variable type. Further, Fig 7.2 shows the dynamics block which contains tire model and vehicle states block.

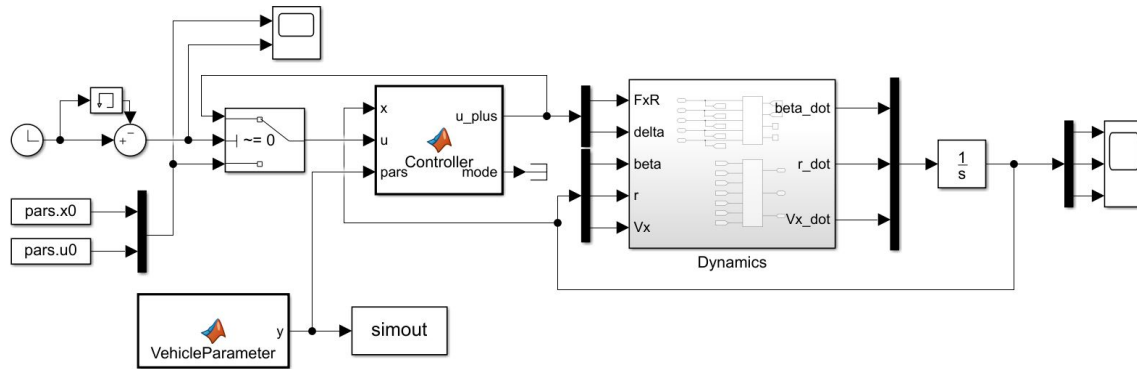


Figure 7.1: Complete drift model in Simulink

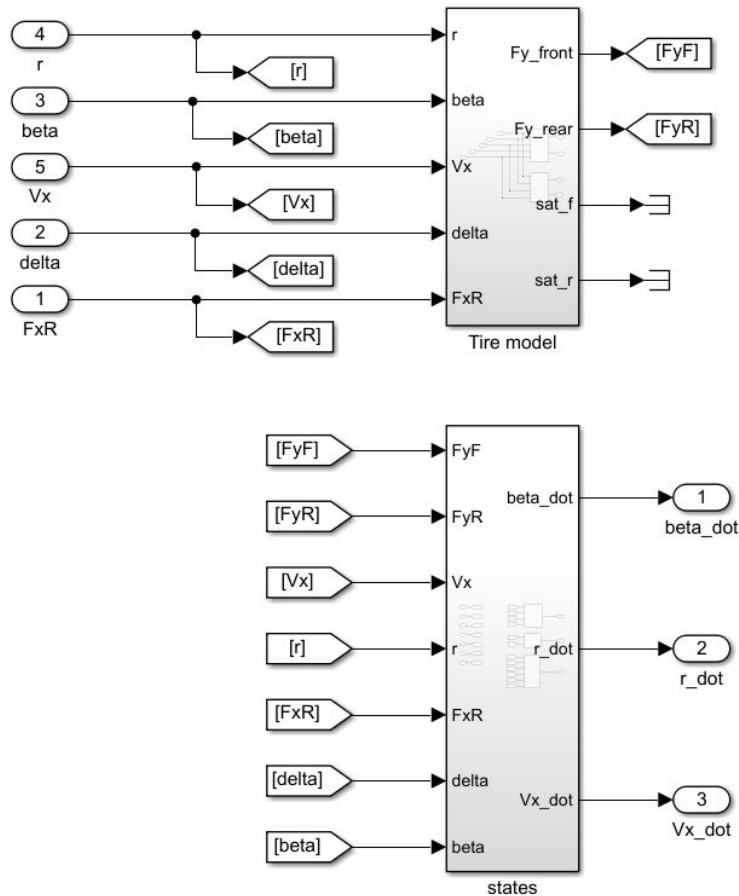


Figure 7.2: Vehicle dynamics block containing tire model and vehicle states

7.2 Tire model block

Fig 7.3 illustrates the tire model block. Different front and rear tire models are used for different slip angle calculations, and have different tire model based on ξ values.

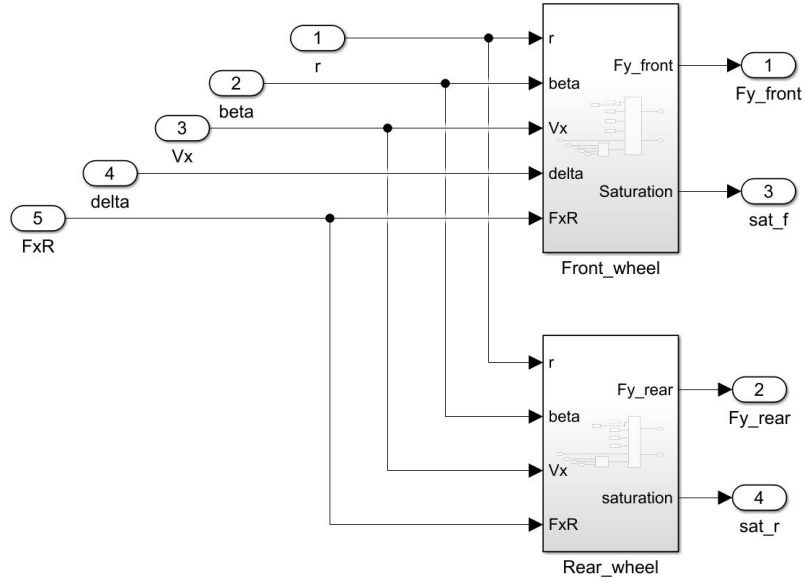


Figure 7.3: Tire model block

7.2.1 Tire slip angle block

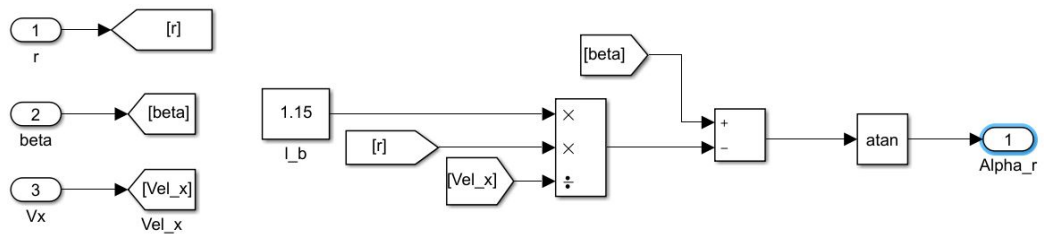


Figure 7.4: Rear tire slip angle block

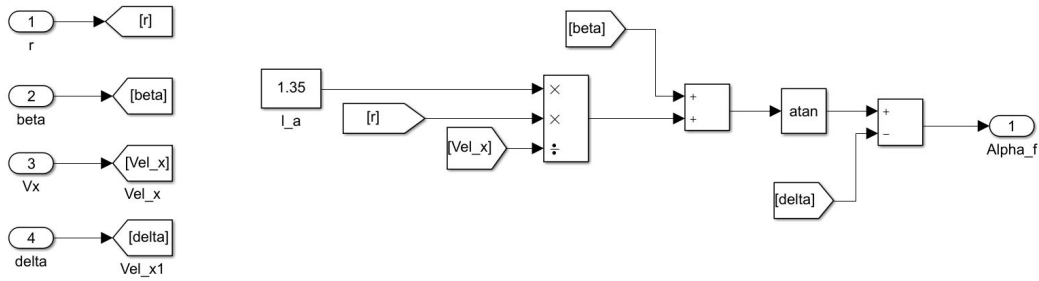


Figure 7.5: Front tire slip angle block

7.2.2 Front tire block

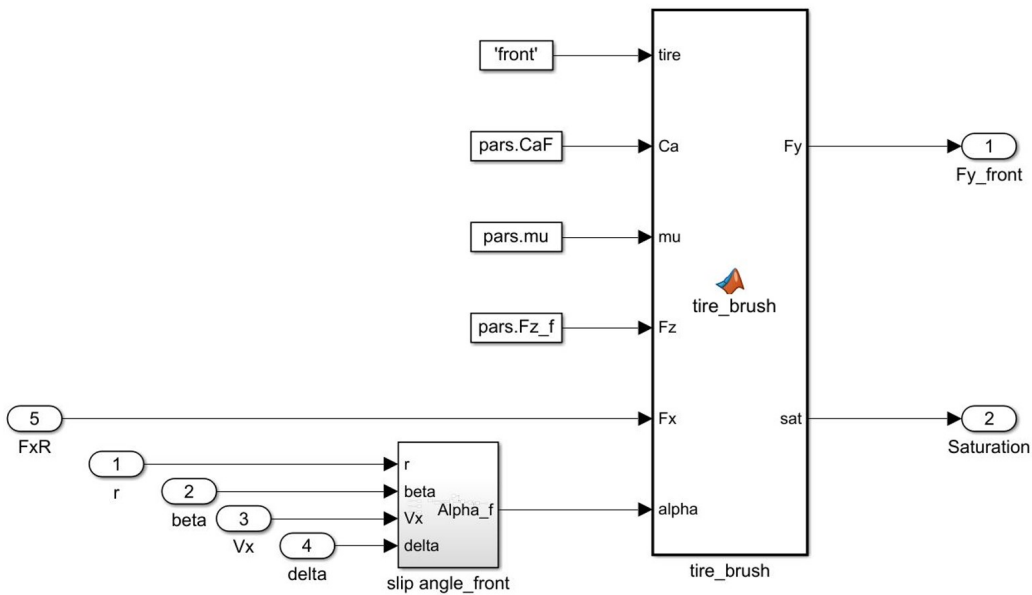


Figure 7.6: Front tire block

Fig 7.6 represents the front tire model, with force calculations are done using a Matlab script. The rear tire block is precisely similar to Fig 7.6, with the following changes: string input as 'rear' and a 'slip angle_rear' block, which is shown in Fig 7.4.

7.3 States block

The vehicle states block is shown in Fig 7.7. Individual states of $\dot{\beta}$, and \dot{U}_x are also shown in Fig 7.8, 7.9 and 7.10, respectively.

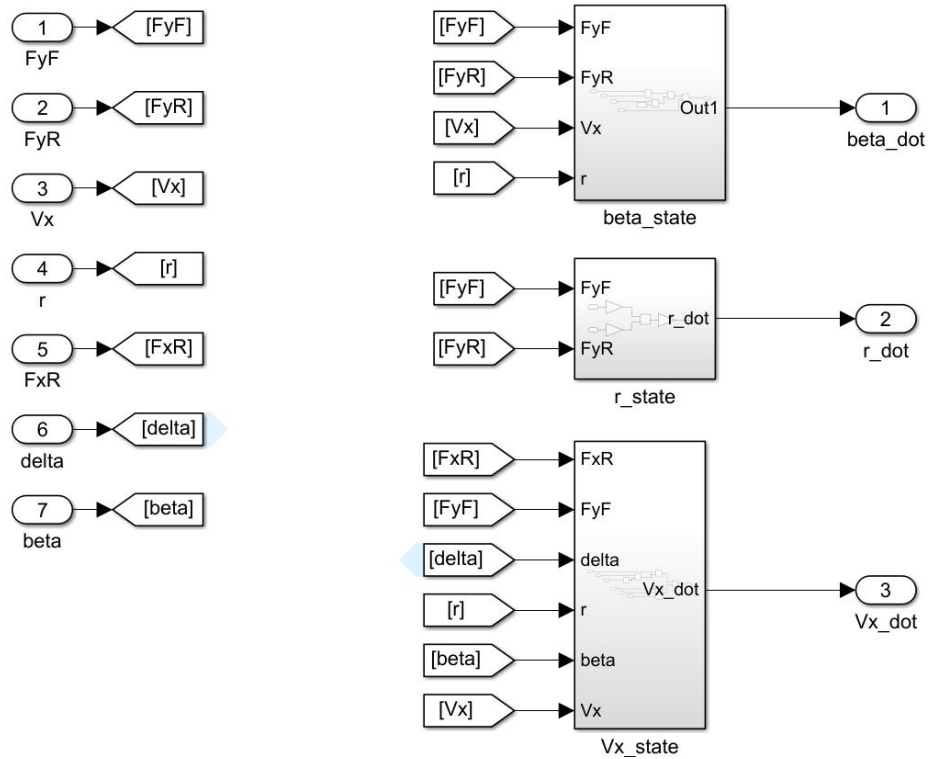


Figure 7.7: Vehicle states block

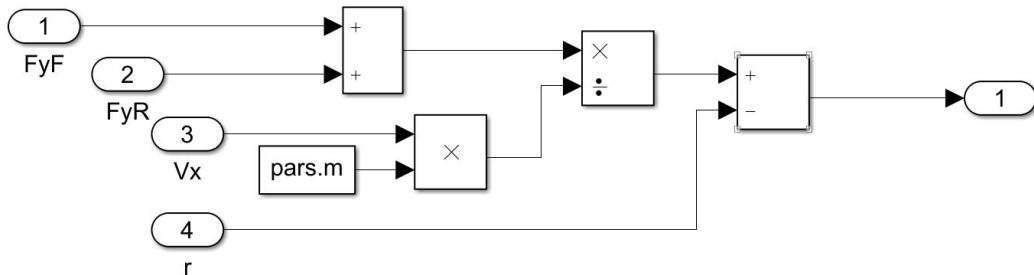


Figure 7.8: Beta state block

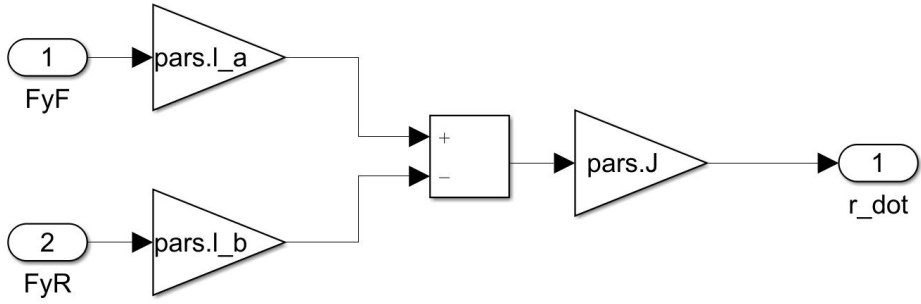


Figure 7.9: Yaw rate state block

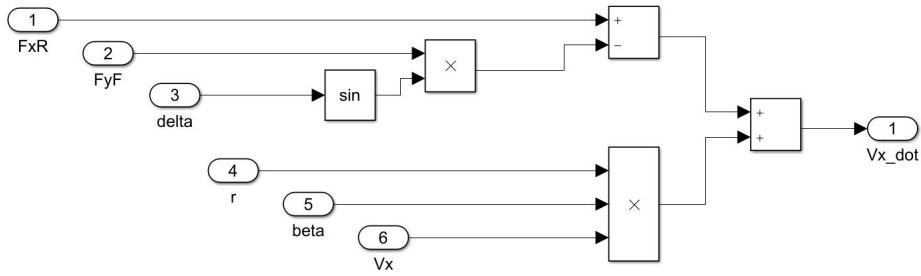


Figure 7.10: Velocity state block

Chapter 8

Results

In this chapter, we have tried to obtain the vehicle states, which correspond to drifting. Since we have discussed earlier that drifting can be defined as a stable equilibrium, therefore, there have to be equilibrium values for the states represented in the three-state model in the equation 6.3. Thus, an equilibrium is carried out to find state values necessary for stable drifting. This analysis also validates that the vehicle dynamics model can simulate drifting. The analysis was run to show the equilibrium locations of our test vehicle AV-2. However, AV-2 vehicle is assumed to be RWD and have 45-55 weight distribution.

8.1 Drift equilibrium analysis

The equilibrium occurs where $\dot{\beta} = \dot{r} = \dot{U}_x = 0$, this yields algebraic equations as shown in 8.1. Since, there are five unknowns and three equations two variables are predefined. Here, δ is varied over the operation range and $U_x = 4\text{m/s}$ [9].

$$\begin{aligned} \frac{F_{yF} + F_{yR}}{mU_x} - r &= 0 \\ \frac{aF_{yF} - bF_{yR}}{I_z} &= 0 \\ \frac{F_{xR} - F_{yF} \sin \delta}{m} + rU_x\beta &= 0 \end{aligned} \quad (8.1)$$

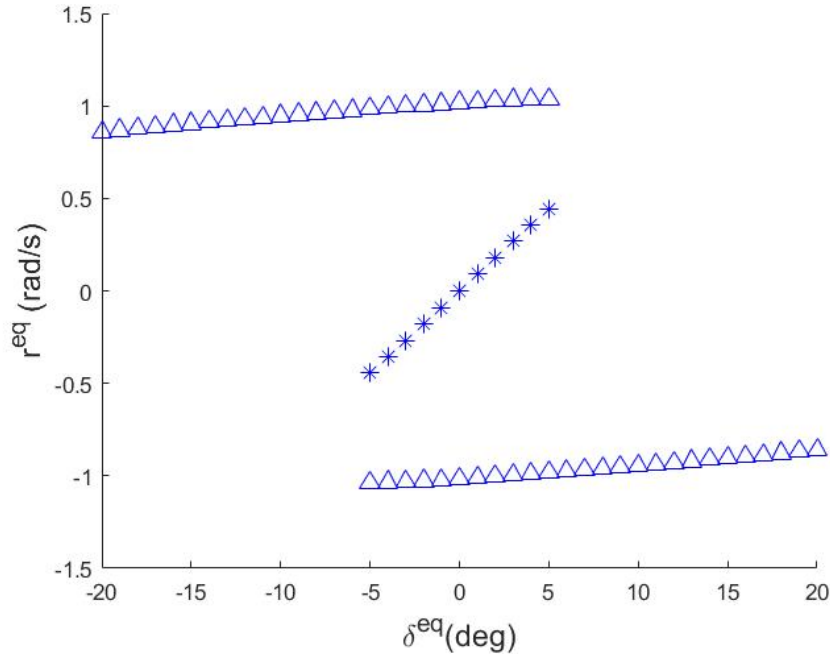


Figure 8.1: Equilibrium yaw rate vs steering angle at $U_x = 4\text{m/s}$ [9][13]

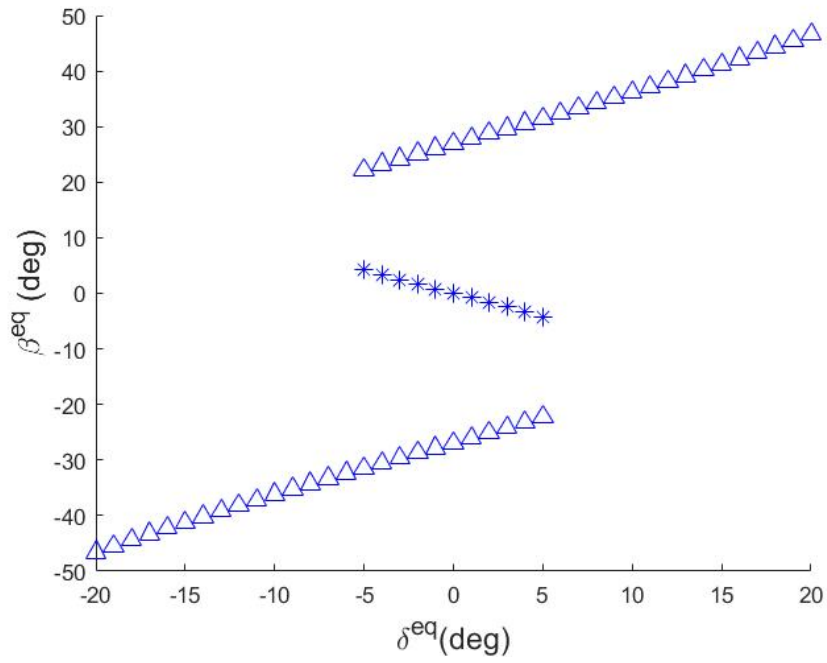


Figure 8.2: Equilibrium sideslip versus steering angle at $U_x = 4\text{m/s}$ [9][13]

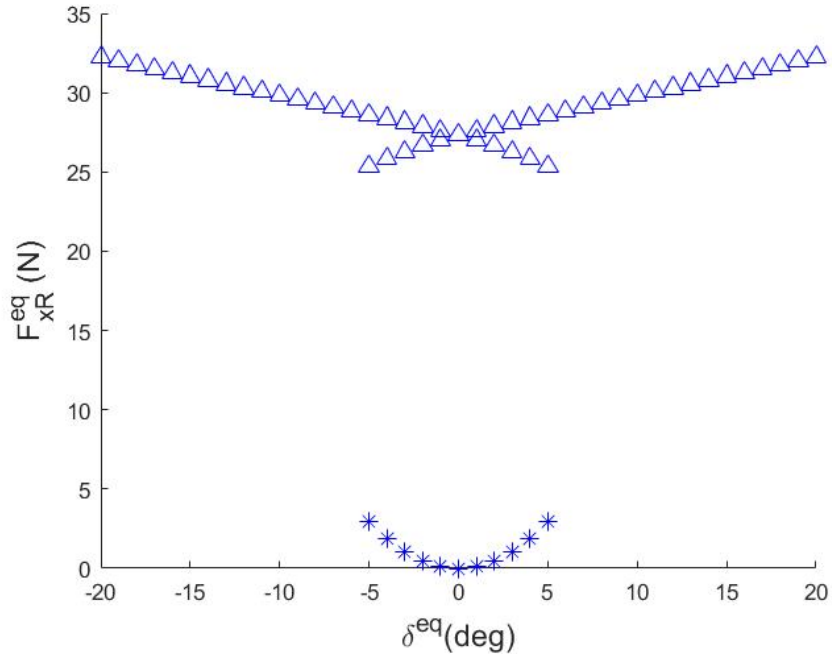


Figure 8.3: Equilibrium longitudinal drive force versus steering angle at $U_x = 4\text{m/s}$ [9][13]

In the figures 8.1, 8.3 and 8.2, symbols Δ and $*$ represent drift equilibrium points and normal cornering equilibrium points, respectively. Following inferences can be made based on this analysis:

8.1.1 Inference of Equilibrium analysis

In Fig 8.1, $*$ shows a linear dependence of yaw rate to the steering angle with positive steering angles increasing positive yaw rate. Which is per the vehicle behavior when operating at lower steering inputs (lower tire slip). Whereas, the yaw rate dependence changes at the points represented using Δ . From the trend, we can notice that a constant positive yaw rate (or negative) spans from both the negative and positive steering angles. Which shows the vehicle is required to countersteer to maintain that yaw rate.

Similarly, in Fig 8.2, sideslip angle values at the drift equilibrium are given. The sideslip increases linearly in the drifting region, accounting for countersteering, which goes in line with the yaw rate vs. steering angle curve in Fig 8.1 as it remains constant.

Furthermore, drift equilibrium points in Fig 8.3 represents the longitudinal force required at the drift equilibrium. The drifting points show the lateral force saturation, which in turn limits the increase of the longitudinal force because both the forces are constraint to friction circle. At typical cornering points, ' $*$ ' the longitudinal force is

much lower as compared to the one required for drifting. It satisfies the intuition that for drifting abrupt acceleration is required along with large steering angles.

One more thing which can be noticed here is both the inputs: longitudinal force and steering angle affect the drift equilibrium differently. Steering input, although required initially to start drifting, is not able to increase the yaw rate after a certain point, because yaw rate remains with respect to the steering angle when the steering angle and yaw rate have the same sign ($[0^\circ, 5^\circ]$ and $[-5^\circ, 0^\circ]$) (Fig 8.1).

Again, an intuitive explanation can provide a rationale for the above-stated hypothesis. When a vehicle is drifting, the rear tire force saturates before the front tires, so, till the front tire saturate, they are still able to increase the yaw rate to sustain the drift. However, after both the tires saturate, the rear longitudinal force is the only force that increases the yaw rate by creating a moment at the CG.

However, during the countersteer, yaw rates can be decreased effectively. Because by changing the direction of the wheel in the opposite direction changes the slip of the tire. Which, in turn, changes the components of the lateral force. Which evident from the Fig 8.1, where yaw rate and steering angle changes sign which yields a slight decrease in yaw rates ($[-20^\circ, -5^\circ]$ and $[5^\circ, 50^\circ]$) (Fig 8.1)

8.2 Controller design considerations

Following the discussion so far, we can layout the following considerations for the controller design for sustained drifting.

1. While maintaining a drift at β^{eq} , if β decreases, then a longitudinal force should be increased to increase the yaw rate hence β increases.
2. When β increases, countersteer should be increased to decrease β .
3. Therefore, the controller must have two-mode design: steering mode and longitudinal force mode. Along with β acting as an artificial input to the yaw controller, which then controls both the modes [9].

Chapter 9

Future work

For this part, a sustained drift controller can be designed, as discussed in section 8.2. Although the fundamental understanding of strategies is discussed in this research, a controller can be designed that takes advantage of the controllability associated with drifting. The equilibrium analysis can provide reference inputs to the controller, which can make a vehicle sustain a drift autonomously.

Also, weight transfer effects are not considered here. A more sophisticated model can be implemented, which represents the vehicle model more accurately. Also, the vehicle is assumed to be a rigid body with no roll and pitch effects. Roll and Pitch affect the lateral and longitudinal weight transfer, respectively.

Further, the model can be extended to an AWD vehicle. AWD vehicles have both

axles producing longitudinal force; hence they will have more controlling inputs. Although this can further increase the complexity of the model, an increased number of controlling parameters can make the controller more robust.

Further, this analysis can be used in a local path planner of an autonomous vehicle coupled with an MPC. A local path planner takes decisions based on the immediate surrounding. Where an MPC will predict the motion of the ego vehicle and will take a note of the drive inputs. Now, if those inputs align with any of the drifting equilibrium points, the local planner can switch to a drift controller and can avoid any unfortunate events on the road.

Chapter 10

Conclusion

Through this research, two important parts of an autonomous vehicle are discussed: mission planning and local planning. The clothoid based path planning can interpolate location and heading data of the ego vehicle throughout the path. Which can be utilized by the vehicle controller to correct the vehicle track at a higher rate. They can also be used at INS only navigation to fuse the vehicle heading data with the current heading. Whereas, the vehicle dynamics model for autonomous drifting and help the drifting controller. A drifting controller can then be used by the local planner in cases where high yaw rates are required. It takes inspiration for the rally drivers where they can navigate at very high speeds and uses tire saturation region for the advantage. An autonomous vehicle can be programmed for those maneuvers and can make our transportation faster and safer than ever.

References

- [1] L. Hower, “Self-driving vehicles: The future always takes longer to arrive,” bettereveryday.vc/preface-as-a-consumer-i-am-excited-by-the-prospect-of-autonomous-vehicles-and-the-individual-and-ebdb73fd9fa9, 2016.
- [2] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer *et al.*, “Autonomous driving in urban environments: Boss and the urban challenge,” *Journal of Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008.
- [3] S. Waslander, “Introduction to self-driving cars by university of toronto on coursera,” <https://www.coursera.org/learn/intro-self-driving-cars/home/welcome>, 2020.
- [4] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, “A survey of motion planning and control techniques for self-driving urban vehicles,” *IEEE Transactions on intelligent vehicles*, vol. 1, no. 1, pp. 33–55, 2016.

- [5] M. Lundberg, “Path planning for autonomous vehicles using clothoid based smoothing of a* generated paths and optimal control,” 2017.
- [6] R. Levien, “The euler spiral: a mathematical history,” *Rapp. tech*, 2008.
- [7] E. Bertolazzi and M. Frego, “G1 fitting with clothoids,” *Mathematical Methods in the Applied Sciences*, vol. 38, no. 5, pp. 881–897, 2015.
- [8] Volkswagen Motorsports, “Unknown territory for the volkswagen polo r wrc–premiere in poland,” <https://www.conceptcarz.com/a8037/unknown-territory-for-the-volkswagen-polo-r-wrc-%E2%80%93-premiere-in-poland.aspx>, 2014.
- [9] R. Y. Hindiyeh, “Dynamics and control of drifting in automobiles,” *PhD thesis*, *PhD Thesis*, 2013.
- [10] R. H. Toyota, “Toyota corolla,” <https://express.richmondhilltoyota.com/inventoryother-vin=2T1BU4EE4DC093464>, 2014.
- [11] W. F. Milliken, D. L. Milliken *et al.*, *Race car vehicle dynamics*. Society of Automotive Engineers Warrendale, PA, 1995, vol. 400.
- [12] R. N. Jazar, *Advanced vehicle dynamics*. Springer, 2019.
- [13] M. Kaba, “Controlled autonomous vehicle drift maneuvering,” Ph.D. dissertation, The Ohio State University, 2019.

- [14] *2018 crash data key findings*. <https://cdan.nhtsa.gov/SASStoredProcess/guest:>
National Highway Transport Safety Administration, Washington DC, USA, 2018.
- [15] SAE International, “Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles,” *SAE International, (J3016)*, 2016.
- [16] K. L. J. Gonzales, F. Zhang and F. Borrelli, “Autonomous drifting with on-board sensors,” in *13th International Symposium on Advanced Vehicle Control, AVEC16*, 2016.
- [17] F. Zhang, J. Gonzales, K. Li, and F. Borrelli, “Autonomous drift cornering with mixed open-loop and closed-loop control,” in *Proceedings IFAC World Congress*, 2017.
- [18] E. Jelavic, J. Gonzales, and F. Borrelli, “Autonomous drift parking using a switched control strategy with onboard sensors,” in *Proceedings 2017 IFAC World Congress*, 2017.
- [19] P. F. Lima, M. Trincavelli, J. Mårtensson, and B. Wahlberg, “Clothoid-based model predictive control for autonomous driving,” in *2015 European Control Conference (ECC)*. IEEE, 2015, pp. 2983–2990.
- [20] S. V. D. Committee *et al.*, “Vehicle dynamics terminology. sae j670e. warrendale, pa: Society of automotive engineers,” 1978.

- [21] M. Gipser, “Ftire: a physically based application-oriented tyre model for use with detailed mbs and finite-element suspension models,” *Vehicle system dynamics*, vol. 43, no. sup1, pp. 76–91, 2005.
- [22] E. Bakker, L. Nyborg, and H. B. Pacejka, “Tyre modelling for use in vehicle dynamics studies,” *SAE Transactions*, pp. 190–204, 1987.
- [23] R. Hadekel, “The mechanical characteristics of pneumatic tires,” *Clearinghouse Fed Sci & Tech Info*, 1952.
- [24] E. B. Craig, “Applications of model predictive control to vehicle dynamics for active safety and stability,” Ph.D. dissertation, PhD thesis, 2011.
- [25] J. J. Van Oosten and E. Bakker, “Determination of magic tyre model parameters,” *Vehicle System Dynamics*, vol. 21, no. S1, pp. 19–29, 1992.
- [26] E. Fiala, “Lateral forces on rolling pneumatic tires,” *Zeitschrift VDI*, vol. 96, no. 29, pp. 973–979, 1954.

Appendix A

Tire parameter determination

A.1 Pacejka model vs Neural Networks

The tires were tried to modeled using different approaches before finalizing on the modified tire brush model. The machine learning was tempting because it does not require any specific parameters to be determine rather a map is created with related all the inputs to the outputs. In the conventional method, the curve fitting equation needs to be calculated every time an output is asked from any inputs. Fig A.1 graphically explains the difference between two approaches. Here, A, B, c and D are pacejka parameters. Also, Neural Networks code used to determine tire parameters is given below.

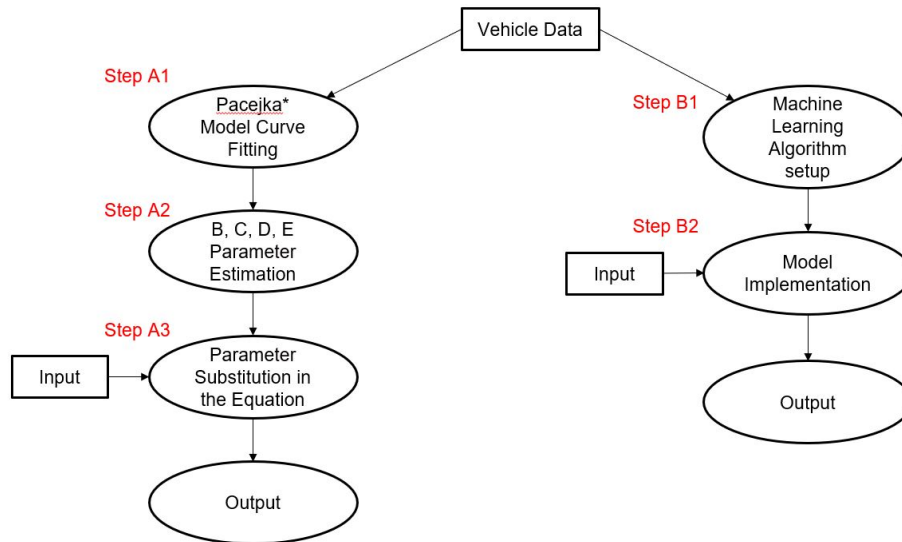


Figure A.1: Pacejka model vs machine learning model

```

data_array = swa_filtered;

%data_table = array2table(data_array, 'VariableNames', {'ax' ←
    ax', 'velo', 'swa'});

ay = ay_filtered;

net = fitnet(9);

net.divideParam.trainRatio = 70/100;

net.divideParam.valRatio = 15/100;

net.divideParam.testRatio = 15/100;

[net, tr] = train(net, data_array', ay');

data_array1 = data_array'; ay1 = ay';

test = net(data_array1(:, tr.testInd));
  
```

A.2 IMU sensor calibration, and scale and bias matrix

IMU calibrations were done at some point in the mainly to use the already gathered data for our analysis. The problem was the previous gathered data of the raw values from the IMU sensor that are needed to be converted into processed data in term of g values to be used to determine tire parameters. Therefore, following calibrations were done on the IMU to get the scale and bias matrix that can be used to process the given data. following functions were used.

```
function [conv,m1,c1] = Linear_scaling_parameters(raw,↵
    proc)
r1 = [max(raw(:,1)), max(raw(:,2)), max(raw(:,3))];
p1 = [max(proc(:,1)), max(proc(:,2)), max(proc(:,3))];
r2 = [min(raw(:,1)), min(raw(:,2)), min(raw(:,3))];
p2 = [min(proc(:,1)), min(proc(:,2)), min(proc(:,3))];
syms m c; m1 = zeros(1,3);c1 = zeros(1,3);
for i = (1:3)
    eqn = [m*r1(i) + c == p1(i), m*r2(i) + c == p2(i)];
    vars = [m c];
```

```

    [m1(i), c1(i)] = solve(eqn, vars);
end

m1 = double(m1);

m1 = [m1(1) 0 0; 0 m1(2) 0; 0 0 m1(3)];

c1 = double(c1)

conv = zeros(length(raw),3);

for ii = (1:length(raw))

    conv(ii,:) = ((m1*(raw(ii,:))) + c1')';

end

end

function [conv,m,c] = Accel_scaling(raw)

m = 1.0e-03*[0.2437, 0, 0; 0, 0.2426, 0; 0, 0, 0.2439];

c = [-0.0027, -0.0096, 0.0231];

conv = zeros(length(raw),3);

for ii = (1:length(raw))

    conv(ii,:) = ((m*(raw(ii,:))) + c')';

end

% assignin('base','m',m)

% assignin('base','c',c)

end

```