Autonomous Driving: Baseline Autonomy

by

Assylbek Dakibay

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Mechanical and Mechatronics Engineering

Waterloo, Ontario, Canada, 2017

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

In near future Autonomous driving will affect every aspect of transportation and offer a significant boost in mobility for everyone. Autonomous driving techniques and modules must be chosen according to the task the platform is developed for. Slow speed driving on campus or highway driving in poor weather conditions, may require different sets of sensors, vehicle models and as a result different software architecture. Some of the main modules that an autonomous driving system needs are the vehicle state estimator and vehicle controller. The development of these two modules relies heavily on the robustness of the vehicle model chosen and the task at hand.

University of Waterloo decided to join the Autonomous Driving research by participating in the project, which required development and implementation of the autonomous driving demo sequence for Consumer Electronics Show in 2017. Since the demo sequence was to be performed at slow speeds and, because certain vehicle parameters were not available at the time, a kinematic vehicle model was used in implementation of the core autonomous driving modules: state estimation and control. These modules are implemented on a full scale autonomous driving platform and were designed based on the needs and requirements of the demo sequence. The implementation results show that the chosen vehicle model enables the state estimator to fuse incoming sensor data and allows the controller to track the desired path and velocity profile.

For further deployment of the autonomous driving platform for research in urban and highway driving an aggressive driving framework was proposed that is based on dynamic vehicle model and incorporates the tire forces in the generation of the speed profile and keeps the vehicle at the limits of adhesion. The aggressive driving controller can be utilized for emergency evasive maneuvers at low road friction conditions. The controller was tested on a high fidelity simulation software for a double lane change emergency maneuver. The results showed that the aggressive driving framework proposed can be successfully incorporated into the autonomous driving architecture and can perform position and velocity tracking at maximum possible speed.

Acknowledgements

I would like to thank my supervisor Steven Waslander for continuous support and guidance and giving me the opportunity to join this incredible field of research. I would also like to thank my labmates in the Waterloo Autonomous Vehicles lab for creating a friendly and supporting work environment.

Dedication

In the name of Allah, the most compassionate, the most merciful!

This work I dedicate to my Lord, to Whom I am thankful for everything I achieved and learned, my parents, who have always been supportive and encouraging and are my role models and my siblings, who have always believed in me and helped me all the way.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Reliable and safe transportation is one of the essential and basic ones in a today's quickly developing and adapting world. Services and commodities that can save even a few minutes of our time are gaining traction due to a fast pace of our lifestyles. Being able to work, read or check emails during a commute to work or while on a road trip, while an autonomous vehicle is driving us to our destination,would introduce an enormous increase in productivity in our day-to-day tasks. Furthermore, introducing the driverless taxis, buses or trucks can significantly boost economical savings and allow people with disabilities and the elderly to have the mobility that a car with a personal driver can offer. Reducing the amount of human driving with robust, safety driven software with multilayer security features is, therefore, a major goal for top universities and research and development departments of major car manufacturers.

In 2016 University of Waterloo decided to join the global effort to build a fully functional autonomous vehicle by first implementing the baseline autonomy demo for CES 2017 in collaboration with Renesas and QNX. The demo comprises of an autonomous vehicle following a predefined path, stopping at the traffic light and stop sign, yielding to an approaching vehicle at the intersection and following the vehicle until the closest intersection. To implement the demo sequence multiple essential modules must be first completed and tested. Two of the most basic modules the demo sequence will need are a state estimation and vehicle control. The state estimation allows the vehicle to know its own location by fusing all of the available sensor data and determining maximum a posteriori estimate of the true vehicle state. Developing the estimator for reliable state estimation requires modeling the vehicle state evolution and the measurement system.

The estimator must take into consideration the vehicle's kinematic and/or dynamic model and fuse the measurement with the expected state evolution. Depending on the purpose of the autonomous vehicle, the target driving speeds and scenarios different vehicle models can be used to derive robust vehicle state estimation. Simple vehicle kinematic models are typically used to implement a state estimator for lower speed driving, where the geometry of vehicle motion is used to predict the future states and correlate it with the wheel odometry, inertial sensor and Global Positioning System (GPS) measurements as was done in [3]. Estimation for high speeds, when the dynamic effects are more prevalent, requires the model to represent the forces acting on the vehicle. Since the main source of forces that act on the vehicle are from the tire - ground interaction, it becomes important to estimate the tire forces in order to robustly observe the vehicle states at high speeds. The estimator structure proposed in [4] utilizes the suspension travel sensors to identify the normal forces at each tire and by fusing them with the inertial measurement system (IMU) outputs produces full vehicle state estimates. The method is heavily reliant on vehicle's suspension parameters and utilizes a cascade design, such that normal forces are estimated separately from the rest of the states. Another approach at estimating dynamic vehicle parameters is proposed in [8]. The method in [8] uses the alignment torque measurement to derive the friction coefficient, normal load and vehicle sideslip angle by means of utilizing the effect of pneumatic trail of the tire. Using the alignment torque for estimating multiple vehicle states, however, requires the nonlinear least squares optimization to be performed at each step to extract the best estimate that fits the model behavior and state history the best. In this thesis a simple kinematic model based state estimator is implemented for the low speed demo sequence. The implemented state estimator uses the measurements that are provided by GPS, wheel odometry and IMU to generate the optimal position, heading and velocity that fit the states predicted by kinematic vehicle model. Given reliable state estimation, a vehicle control module can be defined to track a desired reference trajectory.

A vehicle control module's goal is to calculate the desired actuation signals to the steering wheel, gas and brake based on the position available from the state estimation and the desired path and velocity profile, provided by a local motion planner module. By calculating the position and heading error between the path and the vehicle a steering control command can be generated to minimize the error and keep the vehicle on the path. Similarly, an error between a tracked point on the velocity profile and the vehicle velocity can be used to generate an appropriate longitudinal control command.

Once the vehicle controller and state estimator is developed for a baseline autonomy solution, the vehicle controller must be further improved for higher speeds and should be able to incorporate advanced safety features such as emergency evasive maneuvers for obstacle avoidance. Autonomous driving must be flexible enough for lower speed urban driving as well as highway driving. Both urban driving and highway driving can be quickly affected by adverse weather conditions and, therefore, require the controller to operate at the limits of tire friction. As the friction limits drops the amount of available force from the tire/asphalt interaction saturates and the controller has to take into account maximum allowable longitudinal and lateral acceleration forces. Performing emergency evasive maneuvers to avoid a collision in challenging weather conditions, for example, requires that the vehicle controller performs the motion without losing stability and control while tracking the path that avoids the obstacle, the task that can be performed by an aggressive driving controller for autonomous racing. Existing aggressive driving controllers, focused on driving at the limits of friction for racing applications, can be adopted for autonomous driving applications, enriching the capabilities from path tracking based on simple kinematic model to stable path tracking based on the dynamic vehicle model at the limits of tire friction.

This work performs an extensive review of the existing estimation and vehicle control techniques and describes the best choice for implementing in the demo that satisfies the constraints. After identifing the most suitable vehicle model that matches the available vehicle parameters, the most suitable state estomator and vehicle control modules are derived and implemented. The results during the autonomous driving demo at CES 2017 showed that the implemented estimator and the controller design performed well and achieved the desired goal successfully. Althought the autonomous driving demo was a success the controller design needs to be further improved for urban and highway driving to include vehicle dynamics effects due to tire-road interaction.

Since driving in adverse weather conditions makes the regular controllers based on linear tire models or geometry of the bicycle model unsuitable for safe execution of the path and speed profile tracking task, an aggressive driving controller framework is introduced that can be integrated into an autonomous driving architecture. A speed profile generation method is introduced that uses the friction circle to keep the vehicle at the safe speed with regards to the curvature profile. Further, the aggressive driving controller is implemented with the generated speed profile that performs path tracking along the path with a double lane change maneuver.

This thesis proceeds as follows, the test platform, available sensors and actuation is described in Chapter 2. Chapter 3 covers the vehicle state estimator. Chapter 4 describes the vehicle control framework that best fits the CES 2017 demo time line and platform capabilities and the results of the implemented controller structure. Chapter 5 covers the aggressive driving controller implementation for emergency evasive maneuvers, for highway driving scenario.

Chapter 2

Autonomoose - Platform

The University of Waterloo autonomous driving research platform, known as the Autonomoose, is a 2015 Lincoln MKZ, retrofitted with a drive-by-wire system by Dataspeed Inc. The drive-by-wire system gives the opportunity to control the vehicle without the driver in the loop. The vehicle is a front wheel drive and has a hybrid powertrain with an electrical motor and 2 liter gas engine with a continuously variable transmission.

2.1 Sensor Suite

The platfrom as seen in Figure 2.1 is equipped with Novatel Span GPS system and Novatel Inertial Measurement Unit. The GPS system provides the position accuracy of about 5 cm horizontally and vertically at 20 Hz. The IMU provides the heading angle accuracy of 0.08 degrees published at 100 Hz. The accuracy of the positioning system allows for reliable and robust localization for autonomous driving applications. The CAN-ROS interface additionally provides the encoder readings from all 4 wheels and, therefore, allows for direct vehicle speed measurement once the effective wheel radius is measured or estimated.

The encoder readings from the steering system provide the current steering wheel angle and if divided by the steering column gear ratio - it provides the average steering angle of the front tires. Additionally, the Dataspeeed modules provide the reports on current gear state, braking torque and throttle position. The vehicle comes equipped with additional on-board automotive grade GPS and IMU units. Dataspeed modules provide the readings from the GPS and IMU as standard ROS messages. The Dataspeed modules provide a gear report at 50 Hz, IMU report at 100 Hz and GPS report at 1 Hz. The vehicle is equipped with 3D laser scanner for detection the obstacles and mapping the environment. Additionally vehicle to vehicle (V2V) and vehicle to infrastructure (V2I) communication radio systems are used on the platform to be able to identify the vehicles nearby and get

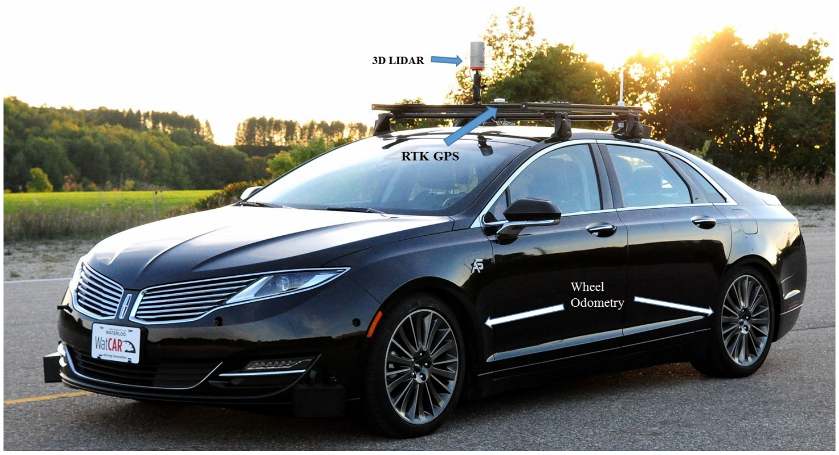

Figure 2.1: Autonomoose Platform

the information such as traffic light signals and road signs and their locations along the path.

2.2 Actuation

The actuation is also provided via ROS interface by means of publishing desired actuation commands as ROS topics. Custom ROS node provided by Dataspeed subscribes to actuation commands and converts them into CAN signals that actuate the vehicle. Available actuation commands are: throttle pedal percentage, brake torque, gear command, steering wheel angle and steering wheel angular speed. All of the actuation commands are sent to CAN at 50 Hz rate. The brake pedal command can also be published as a percentage and Dataspeed's ROS node can convert the percentage into braking torque using brake pedal to brake torque map, in case the control law doesn't utilize the output torque as input. The throttle command, however, can not be converted to desired torque since no engine-motor torque conversion was provided. The steering wheel command receives the values between ± 8.2 rads. The angular speed of the steering wheel ranges from 0 to 8.7 rad/s. The angular speed value of 0 corresponds to a maximum achievable angular speed as set by Dataspeed.

2.2.1 Vehicle's Known Parameters

Due to the complexity of testing required to fully characterize the parameters of a complete vehicle model and tight deadline for the CES demo completion, the process of vehicle system identification was delayed until after the demo and is still underway on the Autonomoose vehicle. We developed a model of the vehicle based on published values for the Lincoln MKZ, which includes the following paramaters, listed in Table 2.1.

Table 2.1: Vehicle parameters as found through online resources

Parameter Name	**Value**	**Units**	**Description**
Weight	1627	kg	Curb Weight of Vehicle
Wheelbase	2.84	m	Distance between front and rear axles
Max Steering Wheel Angle	+/-8.203	rad	Clockwise - negative, counter-clockwise - positive
Steering Wheel Angular speed	0/8.7	rad/s	0 speed corresponds to default - maximum achievable speed
Effective Wheel Radius	0.33	m	The effective wheel radius that is smaller due to tire deflection under the vehicle weight
Steering Column Gear Ratio	1:16	-	Relation between steering wheel angle and front axle average steering angle

Chapter 3

Vehicle State Estimator

3.1 Vehicle Modeling

In order to get baseline autonomy the position and orientation measurements must be filtered for noise. Depending on the measurement and vehicle model the measurement denoising and vehicle state estimation can be performed in various ways. Basic kinematic vehicle model for slow speeds can be described as below, using notation given in [14].

$$\begin{aligned} \dot{X} &= v\cos(\phi+\beta) \\ \dot{Y} &= v\sin(\phi+\beta) \\ \dot{\phi} &= \frac{v\cos(\beta)}{L_{wb}}(\tan(\delta)) \end{aligned} \tag{3.1}$$

The vehicle position, $(X, Y) \in \mathbb{R}^2$ is updated based on vehicle's ground speed v, yaw angle ϕ and slip angle β. In Equation (3.1), $\dot{X}$ and $\dot{Y}$ are velocities in global X and Y coordinates, $\dot{\phi}$ is the heading rate and L_{wb} is the wheelbase. The wheelbase L_{wb} is a sum of a distance from front axle to the centre of gravity, a, and a distance from the rear axle to the centre of gravity, b. The slip angle, β, can be calculated from sine law from based on the geometry shown in Figure 3.1 and derivation from [14]. From the triangle that is formed by point O, CG and centre of front tire in Figure 3.1 a sine law can be represented as:

$$\frac{\sin(\delta-\beta)}{a} = \frac{\sin(\pi/2-\delta)}{R} \tag{3.2}$$

similarly, using the adjacent triangle that connects points O, CG and the centre of the rear axle we get:

$$\frac{\sin(\beta)}{b} = \frac{\pi/2}{R} \tag{3.3}$$

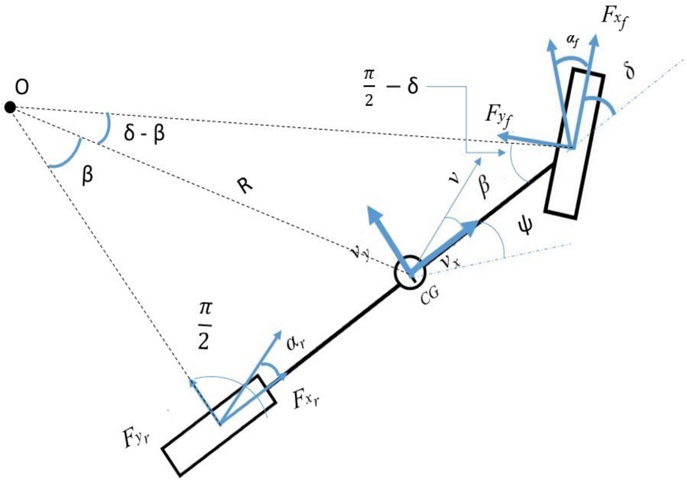

Figure 3.1: Bicycle Model

By expanding the summation of the arguments in the sine function in Equation (3.2) the following expression can be derived:

$$\frac{\sin(\delta - \beta)}{a} = \frac{\sin(\delta)\cos(\beta) - \sin(\beta)\cos(\delta)}{a} = \frac{\cos(\delta)}{R} \tag{3.4}$$

The Equation (3.3) can be simplified further to be:

$$\sin(\beta) = \frac{b}{R} \tag{3.5}$$

Multiplying the first expression in Equation (3.4) with $\frac{a}{\cos(\delta)}$, adding Equation (3.5) results in the following relation:

$$\frac{\tan(\delta)\cos(\beta)}{b + a} = \frac{1}{R} \tag{3.6}$$

Assuming the yaw rate $\dot{\phi}$ is equal to angular velocity of the vehicle at low speeds and assuming no slip, yaw rate $\dot{\phi}$ can be expressed as:

$$\dot{\phi} = \frac{v}{R} = \frac{v\cos(\beta)}{a + b}(\tan(\delta)) \tag{3.7}$$

Using Equations (3.4) and (3.6) slip angle can be derived as:

$$\beta = \tan^{-1}\left(\frac{b\tan(\delta))}{a + b}\right) \tag{3.8}$$

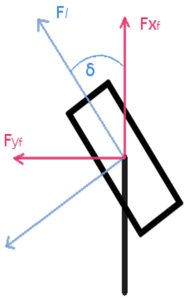

Figure 3.2: Bicycle Model

The forces acting on a center of gravity of the entire vehicle mass m_v can be summarized in Equation (3.9) using a bicycle model in Figure 3.1:

$$\begin{aligned} F_x &= m_v v_y \dot{\phi} + 2F_{x_f} + 2F_{x_r} \\ F_y &= -m_v v_x \dot{\phi} + 2F_{y_f} + 2F_{y_r} \\ I_{zz}\ddot{\phi} &= 2aF_{y_f} - 2bF_{y_r} \end{aligned} \tag{3.9}$$

The total forces acting in the X and Y directions in the vehicle's frame are F_x and F_y, meanwhile the forces F_{x_f}, F_{x_r} are the force components on the front and rear tires in the X direction, whereas F_{y_f}, F_{y_r} are force components acting on a tire in the Y direction. The velocities v_x and v_y are the velocity components of the vehicle in the vehicle frame, whereas I_{zz} is the vehicle's moment of inertia about Z axis in the vehicle frame and $\ddot{\phi}$ is the angular acceleration about axis Z. The front tire forces as seen in Figure 3.2 can be expressed in terms of longitudinal and lateral components:

$$\begin{aligned} F_{x_f} &= F_l \sin(\delta) + F_c \cos(\delta) \\ \\ F_{y_f} &= F_l \cos(\delta) - F_c \sin(\delta) \end{aligned} \tag{3.10}$$

and for the rear tires the lateral and longitudinal forces overlap with the forces in the X and Y directions.

$$\begin{aligned} F_{y_r} &= F_c \\ F_{x_r} &= F_l \end{aligned} \tag{3.11}$$

The longitudinal F_c and lateral F_l forces in the linear tire region can be described as:

$$\begin{aligned} F_c &= C_c \alpha \\ F_l &= C_l \lambda \end{aligned} \tag{3.12}$$

C_l and C_r are the longitudinal and lateral tire stiffness respectively, α and λ are a slip angle and a slip ratio of individual tire. The slip ratio can be calculated by using the wheel rotational speed ω, longitudinal tire speed v_x in the tire frame and effective wheel radius r_{eff}:

$$\lambda = \frac{r_{eff}\omega - v_x}{r_{eff}\omega} \tag{3.13}$$

and slip angle α_{tire} can be found from the ratio of the lateral and longitudinal velocities of the tire:

$$\alpha = \frac{v_y}{v_x} \tag{3.14}$$

From the bicycle model the front and rear tire slip angles can be related to vehicle kinematics as shown below.

$$\begin{aligned} \alpha_f &= \beta + \frac{a\dot{\phi}}{v_x} - \delta \\ \alpha_r &= \beta - \frac{b\dot{\phi}}{v_x} \end{aligned} \tag{3.15}$$

3.2 Existing Work On Vehicle State Estimation

The existing work on the vehicle state estimation can be summarized in two main categories:

- Kinematic parameter estimation
- Dynamic parameter estimation

3.2.1 Kinematic Parameter Estimation

The main vehicle kinematic states such as position, heading and velocity can be estimated using differential GPS system, on board IMU and wheel odometry with an Extended Kalman Filter (EKF). In [3] the GPS, INS and wheel encoders were used for pose and yaw angle estimation and for identifying the effective wheel radii R^l and R^r. The sideslip β was assumed to be zero.

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\phi} \\ \dot{R}^r \\ \dot{R}^l \end{bmatrix} = \begin{bmatrix} -V_g sin(\phi) \\ V_g cos(\phi) \\ \frac{1}{a}r - \frac{b}{a} \\ 0 \\ 0 \end{bmatrix} \tag{3.16}$$

The measurements from the GPS are the X, Y position in NEU (North, East, Up) frame, heading, ground speed yaw rate. When the GPS is not available the wheel encoder readings can be used for identifying the yaw rate from Equation (3.17) using the rear track width E_r assuming a front wheel drive vehicle:

$$\dot{\phi} = \frac{V^r}{E_r/2} - \frac{V^l}{E_r/2} \tag{3.17}$$

The wheel speeds V^r and V^l are estimated using the wheel radius estimates R^l and R^r and rotational speeds from wheel encoders ω_r and ω_l.

$$\begin{aligned} V^r &= \omega_r R^r \\ V^l &= \omega_l R^l \end{aligned} \tag{3.18}$$

Since wheel encoders provide the measurements at a higher rate compared to GPS, the yaw rate $\dot{\phi}$ estimated from wheel encoders can be integrated and added to the last yaw measurement from GPS. The front wheel drive vehicle can provide reliable wheel speed measurements except for the cases, when the brakes are actuated and the longitudinal slip is present on all four wheels - yaw rate measurements will be corrupted with non-Gaussian noise that can only be estimated by extending the vehicle model. Hence an INS or IMU system must be integrated to identify the vehicle dynamic parameters such as longitudinal slip ratio λ and, for more reliable position estimates, the sideslip angle β.

3.2.2 Dynamic Parameter Estimation

Multiple vehicle models exist in the literature and, accordingly, different state estimation strategies have been developed that utilize the benefits of each modeling technique. In [4] four wheel vehicle model (FWVM) has been used for state estimation of vehicle dynamic properties such as sideslip angle and tire forces. The FWVM has some additional parameters compared to bicycle model, such as front and rear track width E_f, E_r. The estimation process can be summarized in two blocks:

- estimating the normal forces on 4 tires and accelerations of cg
- estimating individual tire forces and vehicle side slip angle

The first block utilizes the measurements from the suspension sensors and IMU and provides the normal force estimates on the tires. The measurements from the suspension sensors are used to identify the load variations of m^s_{ij} at each wheel using quarter car model and a linear spring approximation with k_s being the spring coefficient, and $\triangle_{ij}$ is the relative spring displacement, whereas index i represents the front/rear axle and index j represents the left or right wheel respectively. The change in load $\triangle m_{ij}$ at each corner can be calculated as:

$$\triangle m^s_{ij} = \frac{k_s \triangle_{ij}}{g} \tag{3.19}$$

The load at each wheel is estimated using the manufacturer provided initial load at rest and the variation in load m_{ij}^s. A linear Kalman filter is then utilized to estimate the lateral load transfer $\triangle F_{z_l}$ using the lateral acceleration a_y and steering angle δ as shown in Equation (3.20). The mass at each corner, m_{ij}, is estimated from the suspension travel in the previous block by adding mass variation, $\triangle m_{ij}$, to the mass at rest, m_{ij}^s. The parameters k_f and k_r, are the front and rear roll bar stiffnesses respectively, h_f and h_r are heights to the front and rear roll centers, m_s is, sprung mass and sum of mass, m_{ij}^s at each corner.

$$\triangle F_{z_l} = (m_{fl} + m_{rl} - m_{fr} - m_{rr})g - 2\left((\frac{k_f}{E_f} + \frac{k_r}{E_r} \right)\theta - 2\frac{m_s a_y}{l}\left(\frac{bh_f}{E_f} + \frac{ah_r}{E_r}\right) \tag{3.20}$$

The roll angle, θ, can be estimated using relative spring displacements $\triangle_{ij}$, lateral acceleration of the entire vehicle, a_{yv}, total vehicle mass, m_v, height of the cg, h, and tire stiffness, k_t.

$$\theta = \frac{\triangle_{fl} - \triangle_{fr} + \triangle_{rl} - \triangle_{rr}}{(2e_f)} - \frac{m_v a_{ym} h}{k_t} \tag{3.21}$$

Additionally the load shift due to longitudinal acceleration for front wheels $F_{z,fi}$ and rear wheels $F_{z,ri}$ is estimated as follows:

$$\begin{aligned} F_{z,fi} &= \frac{1}{2}m_v(\frac{b}{L_{wb}}g - \frac{h}{l}a_x) \pm m_v(\frac{b}{L_{wb}} + \frac{h}{l}a_x)\frac{h}{E_f g}a_y \\ F_{z,ri} &= \frac{1}{2}m_v(\frac{b}{L_{wb}}g - \frac{h}{l}a_x) \pm m_v(\frac{a}{L_{wb}} + \frac{h}{l}a_x)\frac{h}{E_r g}a_y \end{aligned} \tag{3.22}$$

Using the four wheel vehicle model shown in Figure 3.3 and using a tire model $F_{ij} = \Upsilon(F_{z,ij}, \alpha_{ij}, \zeta_{ij})$, the state vector $X = [\dot{\phi}, V_g, \beta, F_{y_{lr}}, F_{y_{rr}}, F_{y_{fl}}, F_{y_{fr}}, F_{x_f}]^T$ can be expressed as follows:

$$\begin{aligned} \dot{V}_g &= \frac{1}{m_v}(F_{x_f}\cos(\beta - \delta) + F_{y_{fl}}\sin(\beta - \delta) + (F_{y_{rl}}) + F_{y_{rr}})\sin(\beta - \delta)) \\ \ddot{\phi} &= \frac{1}{I_{zz}}\left[\begin{array}{l} a(F_{y_{fl}}\cos(\delta) + F_{y_{fr}}\cos(\delta) + F_{x_f}\sin(\delta)) - \\ b(F_{y_{21}} + F_{y_{22}}) + \frac{E_f}{2}(F_{y_{fl}}\sin(\delta) - F_{y_{fr}}\sin(\delta)) \end{array}\right] \\ \dot{\beta} &= \frac{1}{m_v v}\left[\begin{array}{l} F_{x_f}\sin(\beta - \delta) + F_{y_{fl}}\cos(\beta - \delta) \\ +(F_{y_{fr}}) + F_{y_{fl}})\cos(\beta - \delta) \end{array}\right] \\ a_y &= \frac{1}{m_v}\left[F_{y_{fl}}\cos(\delta) + F_{y_{fr}}\cos(\delta)(F_{y_{rl}} + F_{y_{rr}}) + F_{x_f}\sin(\delta)\right] \\ a_x &= \frac{1}{m_v}\left[-F_{y_{fl}}\sin(\delta) - F_{y_{fr}}\sin(\delta) + F_{x_f}\sin(\delta)\right] \end{aligned} \tag{3.23}$$

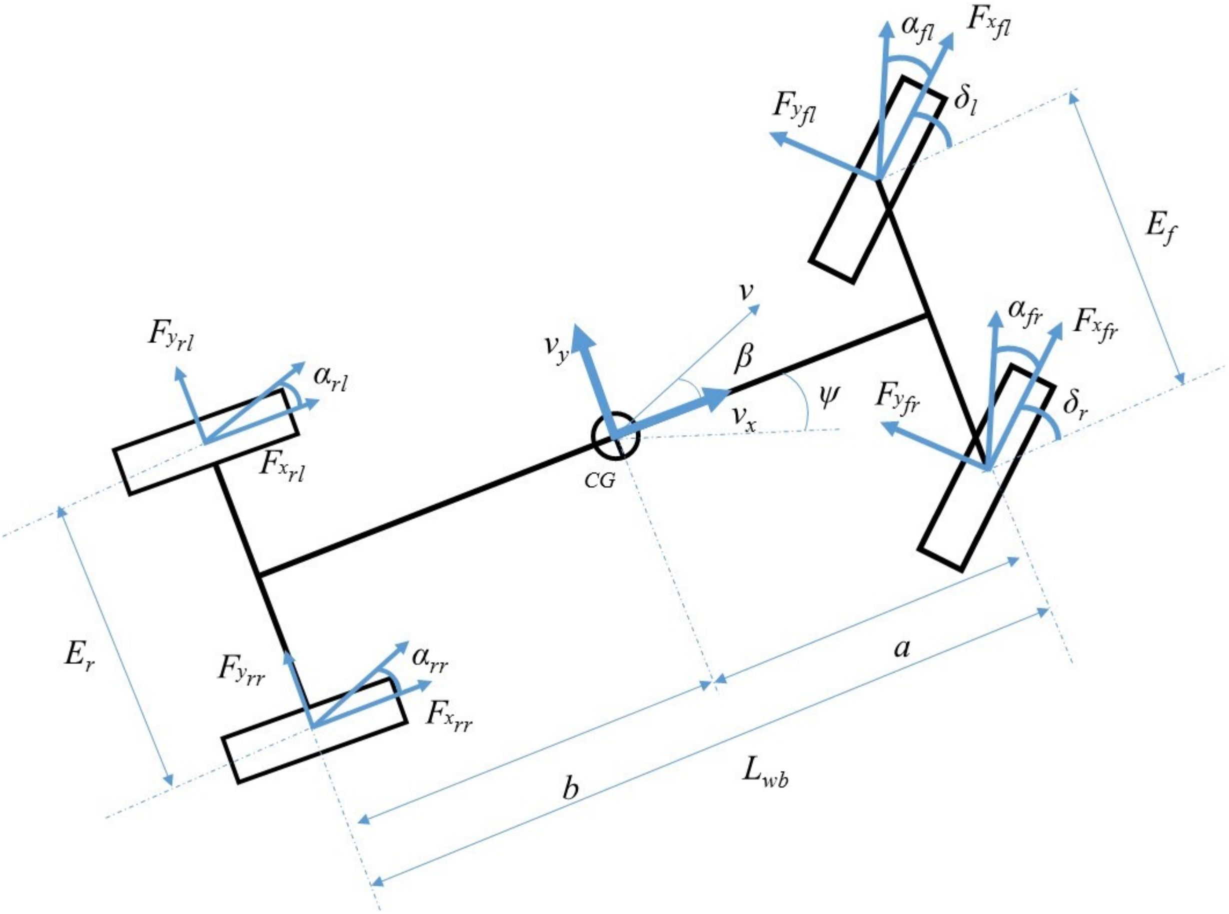

Figure 3.3: FWVM model

The measurement vector is $Y = \left[\dot{\phi}, V, a_x, a_y\right]$. The ground velocity V was measured using wheel encoders, yaw rate and accelerations are calculated using IMU readings. This method offers good results at estimating vehicle's dynamics characteristics defined by state vector X, but requires knowledge of the chassi parameters such as k_f front roll bar stiffness, k_r rear roll bar stiffness, k_t tire stiffness, k_s suspension spring stiffness, h_f front roll center height, h_r rear roll center height, h cg height. Although these parameters can be obtained using in-depth system identification and special modelling tools, the tight deadline and high cost of the modelling tools make it not a feasible method for given task.

Another approach is to estimate the vehicle dynamic characteristics using the steering torque measurements and the effect of pneumatic trail as was done in [8]. The method utilizes the parabolic pressure distribution on the tire and no longitudinal forces. The lateral force of the entire front axle can be characterized by sum of the lateral forces on two tires.

$$F_y = \begin{array}{ll} -C_c \tan\alpha + \dfrac{C_c^2}{3}|\tan(\alpha)|\tan(\alpha)I_f - \dfrac{C_c^3}{27}\tan^3(\alpha)I_f^2 & if|\alpha| \leq \alpha_S L \\ -\dfrac{1}{I_f}sgn(\alpha) & else \end{array} \tag{3.24}$$

The inverse normal front axle load $I_f = \dfrac{1}{\mu F_z}$ is main parameter to be estimated in this method. Using the relation between tire slip angle and the vehicle kinematics expressed in Equation (3.15) the front and rear axle dynamics can be summed as:

$$\begin{aligned} \dot{\beta} &= \frac{1}{M_v}(F_{yf} + F_{yr}) - \dot{\phi} \\ \ddot{\phi} &= \frac{1}{I_z}(aF_{yf} - bF_{yr}) \end{aligned} \tag{3.25}$$

Differentiating Equations (3.15) and using Equation (3.25) the front axle slip angle α_f can be found from integrating expression below:

$$\dot{\hat{\alpha}}_f = \left(\frac{1}{mv_x} + \frac{a^2}{I_z z v_x}\right)F_{yf} + \left(\frac{1}{mv_x} - \frac{ab}{I_{zz}v_x}\right)F_{yr} - \dot{\phi} - \dot{\delta} \tag{3.26}$$

The δ is the steering wheel reading, v_x can be measured from the wheel encoders and $\dot{\phi}$ can be measured using the on board IMU. Using the Equation (3.24) The vertical load on the front axle is found using the pneumatic trail equation (3.27) and the aligning torque τ_a expressed using the pneumatic tire model and the pneumatic trail t_p.

$$t_p = \begin{array}{ll} t_{PO} - \dfrac{t_{PO}C_r}{3}I_f|\tan(\alpha)|, & \text{if } |\alpha| \leq \alpha_{sl} \\ 0, & \text{otherwise} \end{array} \tag{3.27}$$

$$\tau_a = -(t_m + t_p)F_y \tag{3.28}$$

The mechanical trail t_m can be found from suspension kinematics as described from [12]. Using the estimates of the inverse front load I_f load and the nominal value the road friction coefficient can be found as well.

$$\hat{\mu} = \frac{1}{I_f F_{z_f}} \tag{3.29}$$

The method described in [7] can be implemented without any additional sensor that are not already on the car. The only disadvantage is the use of the front lateral stiffness coefficient C_{cf}, which is unknown for the demo vehicle, and the assumption of no longitudinal forces on the tire. The longitudinal forces can be included by using a different tire model such as Dougoff tire model as was done in [20].

The integration of GPS and Inertial Navigation Sensor (INS) for estimation of vehicle dynamic properties has been proposed in [1]. The GPS readings were used for estimation of lateral and longitudinal speeds v_x, v_y, INS measurements used for estimating the lateral acceleration and yaw rate $\dot{\phi}$. The lateral acceleration is integrated to estimate the v_y lateral velocity between the GPS velocity measurements. By numerically differentiating the yaw rate $\dot{\phi}$ and using the sideslip angle β, front and rear slip angles from Equation (3.15) the cornering stiffness can be continuously estimated as well by solving the system of Equations (3.9) and assuming linear tire forces in Equation (3.12). The only limitation being the unreliability of the GPS measurements due to possibility of loss of signal.

3.2.3 Summary

Successful implementation of baseline autonomy depends on the ability to localize the vehicle and track its kinematic and dynamic parameters. The estimation methods vary by the models they utilize, sensors used and tracking performances and limitations. The method described in 3.2.1 proposed by [3] shows promising results for relatively simple approach and doesn't require any additional sensors to be installed on the car. The only disadvantage is that the longitudinal forces are neglected and possible occurrence of wheel slip can affect the estimation of the yaw rate and as a result yaw angle - which can lead jitter by automated steering controller. The assumption of no side slip limits the range of speeds this methods can be used for due to tire dynamics that is not accounted for. Additional estimators can be implemented as mentioned in 3.2.2 that can be used to improve the planar kinematics method proposed by [3] as observed measurements and improve the yaw and position estimates at higher speeds and accelerations. The only limitation being the vehicle characteristics that are not shared by the manufacturer such as: tire curve, various suspension characteristics as required by method in [4] or tire cornering stiffness coefficient and front suspension kinematic parameters to estimate the pneumatic trail and extract the dynamic parameters from it as described in [7]. The most feasible among the dynamic parameter estimation methods is the one described in [1], that allows for estimation of the side slip angle and cornering stiffness continuously. Although this method is great for estimation cornering stiffness with limited sensor suite, the loss of GPS signal can be detrimental to the estimation of sideslip angle, which depending on the steering controller and driving speeds, can result in large estimation errors. The combination of GPS-INS based estimators [1] and pneumatic trail based observers as in [8] or in [20] can be implemented, where the cornering stiffness, sideslip angle and tire slip angles can estimated using GPS and INS. When the GPS measurements are not available the pneumatic trail method can be used for estimation of the slip angle, friction limit and tire lateral forces. The CES 2017 demo was planned to take place in a small demo track and due safety concerns was limited to low speeds. The dynamic properties such as side slip and tire forces were not expected to have a significant effect on the vehicle kinematics at low speeds, additionally the on board INS on ”Skyline” has much higher accuracy compared to the inertial sensors used in [4],[1] and [3]. The loss of GPS signal, therefore, would have less effect on the estimation of the position and yaw angle, since the INS unit can provide the pose estimates from internal filter that integrates the accelerometer and gyroscope readings. It was decided, therefore, to implement an EKF estimator based on [3] with modifications due to additional sensors.

3.3 Implementation

As was mentioned in Chapter 2 the GPS-INS unit installed on the ”Skyline” has top grade accuracy for several minutes due to fiber-optic gyroscopes when the GPS signal is lost. The yaw rate and yaw angle integrated from it can be estimated using the INS instead of rear wheel encoders in the absence of GPS measurements, the vehicle speed is to be estimated using the wheel encoders at higher rate rather than using the GPS as was done in [3]. The reason for using INS for yaw rate instead of wheel speeds is that INS has a higher accuracy and the longitudinal slip due to braking can significantly skew the estimates of yaw angle and hence affect the pose estimates as well.

The model in Equation (3.1) can be simplified and expressed as discrete system $X_t = f(X, U)$ as shown below for the EKF estimation process in Equation (3.30), but the input vector here is ignored for a simpler model.

$$\begin{bmatrix} X_t \\ Y_t \\ \phi_t \\ v_{x_t} \\ \dot{\phi}_t \\ a_{x_t} \\ a_{y_t} \end{bmatrix} = \begin{bmatrix} X_{t-dt} & + & v_{x_{t-dt}} \cos(\phi_{t-dt}) dt \\ Y_{t-dt} & + & v_{x_{t-dt}} \sin(\phi_{t-dt}) dt \\ \phi_{t-dt} & + & \dot{\phi}_{t-dt} dt \\ v_{x_{t-dt}} & + & a_{x_{t-dt}} dt \\ \dot{\phi}_{t-dt} & & \\ a_{x_{t-dt}} & & \\ a_{y_{t-dt}} & & \end{bmatrix} \tag{3.30}$$

The yaw angle ϕ is zero at the East X coordinate according to East North Up (ENU) convention. When the velocity v_x estimates are not available the IMU acceleration a_x is added to the previous velocity readings in the prediction step. Similarly the previous yaw rate estimate is used for numeric integration of the yaw angle in the prediction step, since all the IMU readings are published at the same rate.

3.3.1 Extended Kalman Filter For Autonomous Demo

Using the standard EKF algorithm 3.31:

$$
\begin{aligned}
&\text{Prediction}\\
&\ \bar{X}_t = f_{t-dt}\hat{X}_{t-dt}\\
&\ \bar{P}_t = F_{t-dt}P_{t-dt}F_{t-dt}\\
&\text{Update step}\\
&\ \hat{X}_t =\\
&\ K_t = \bar{P}_t H_t (H_t \bar{P}_t H_t + R_t)^{-1})\\
&\ P_t = (I - K_t H_t)\bar{P}_t
\end{aligned}
\tag{3.31}
$$

In the algorithm above F_t is the Jacobian of the discretized nonlinear motion model defined by function f_t estimated at the previous time instant using the previous estimate, and H_t is the Jacobian of the measurement matrix, but since states to be measured are directly what the sensors provide - the matrix H_t is identity. The matrices R and Q are the process and measurement noise matrices. The measurement matrices for different sensors are provided below:

$$
\begin{aligned}
&\text{GPS Measurement}\\
&H_{GPS} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}\\
&\text{Encoder velocity}\\
&H_{Wheels} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}\\
&\text{IMU}\\
&H_{IMU} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
\tag{3.32}
$$

The Jacobian of the motion model with respect to state vector is listed below:

$$
F_t = \begin{bmatrix}
1 & 0 & -V_{x_{t-dt}} \sin(\phi_{t-dt})dt & \cos(\phi_{t-dt})dt & 0 & 0 & 0 & 0 \\
0 & 1 & V_{x_{t-dt}} \cos(\phi_{t-dt})dt & \sin(\phi_{t-dt})dt & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & dt & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & dt & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\tag{3.33}
$$

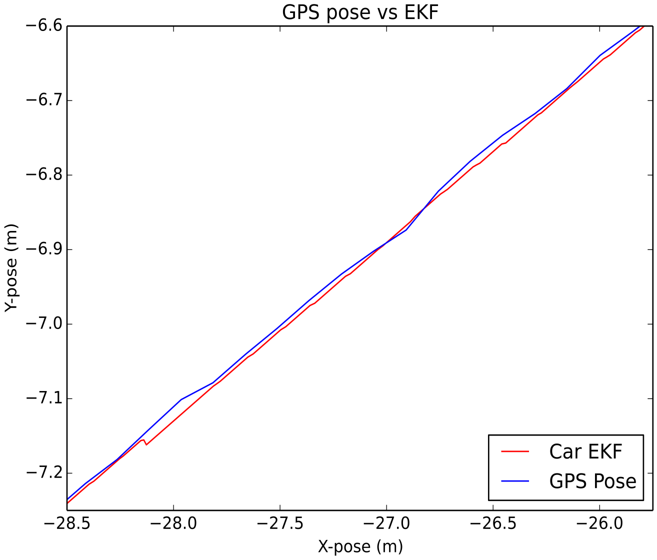

Figure 3.4: EKF vs raw GPS Measurements

The wheel encoder readings where averaged between two rear wheels and the wheel radius was assumed to be constant. The reason for not estimating the wheel radius as was done in [3] is due to unreliability of ROS provided timers that are necessary to identify the velocity from GPS readings. The GPS ROS drivers only provide position estimates and hence the ground velocity must be estimated by numeric differentiation.

3.4 EKF Results

The random noise in the GPS measurements results in readings that can indicate that vehicle is moving sideways or backwards even when the vehicle parked. The EKF utilizes the vehicle model in the prediction step to propose the next pose due to the vehicle speed and heading and fuses it with the GPS measurements. In Figure 3.4, the EKF smoothens the random noise coming from the GPS, reduces the jitter and fluctuation. Since the weights for GPS in the measurement noise matrix Q were set to be low, the EKF estimates follow the GPS measurements very closely reducing the small amount of noise that is present. The measurement of the vehicle heading from IMU were also successfully smoothened and correlated with the integrated yaw rate prediction step as can be seen from Figure 3.5. The IMU update rate is much higher than that of the GPS and therefore the oscillation of the measurements has higher frequency. The EKF smoothens the drastic IMU measurement as can be seen at 4.5-4.9 seconds on the plot, hence allowing for more smooth control signal to steering and more comfortable ride.

Similarly the velocity measurement that have noise due to encoder speed registration

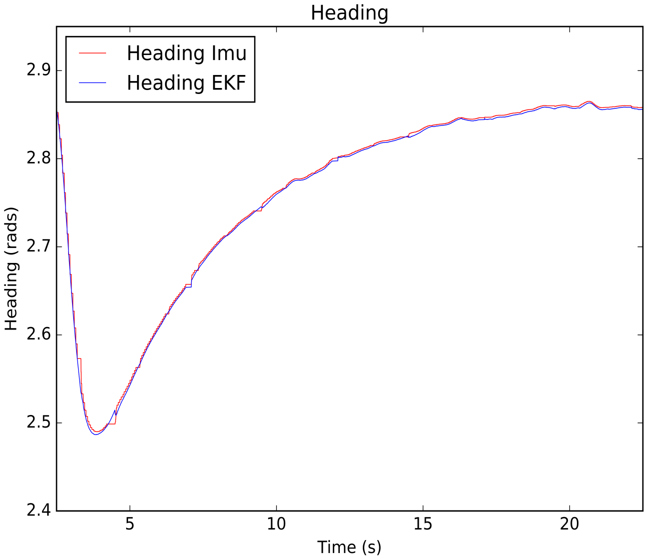

Figure 3.5: Heading Estimation (IMU vs EKF)

errors, are smoothened and drastic changes in the odometry measurements that don't corrspond to the integral of the current acceleration a_x do not skew the EKF velocity estimates.

Similarly the longitudinal and lateral accelerations and the yaw rate readings from the IMU were successfully smoothed in real time reducing the effect of random noise in the measurements as can be seen from Figure 3.7, 3.8 and 3.9. Since the model used is a constant acceleration model and since the lateral or longitudinal jerk and angular acceleration is not measured, the acceleration measurements and yaw rate are only smoothened by the EKF.

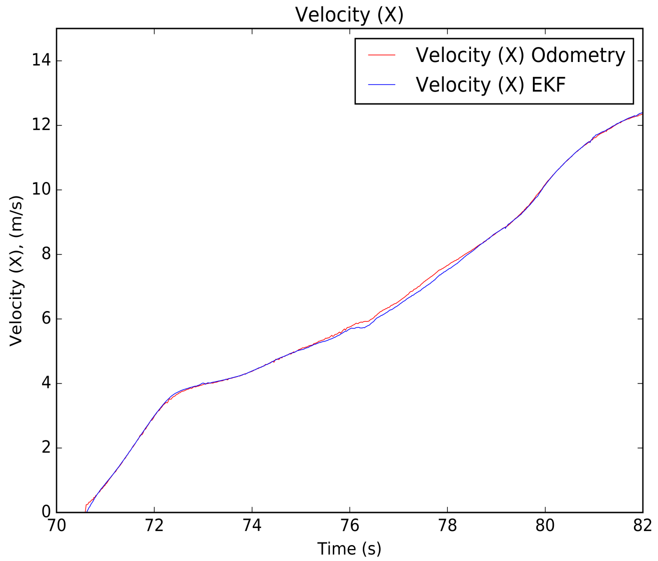

Figure 3.6: Velocity Estimation (IMU vs EKF)

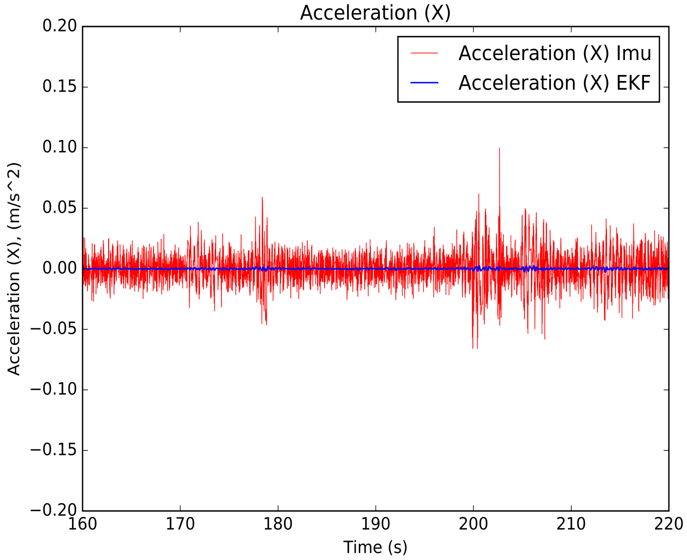

Figure 3.7: Acceleration (X) Estimation (IMU vs EKF)

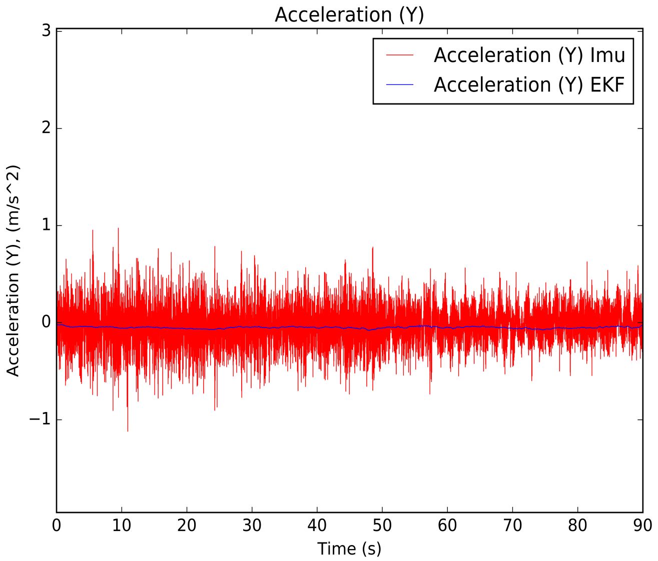

Figure 3.8: Acceleration (Y) Estimation (IMU vs EKF)

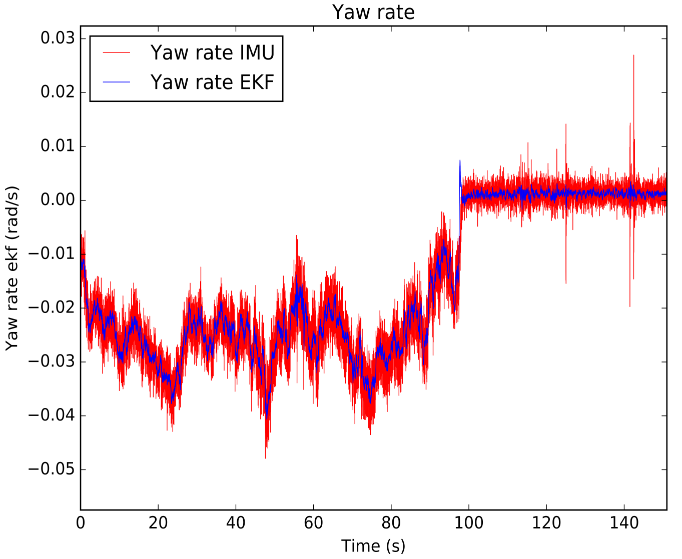

Figure 3.9: Yaw Rate Estimation (IMU vs EKF)

Chapter 4

Vehicle Control

The vehicle controller for an autonomous vehicle can be characterized with two main components:

- Steering Controller
- Longitudinal Controller

The trajectory planner node publishes a local path and an associated velocity profile as a discrete vector. At every loop the controller finds the point on the velocity profile that corresponds to the vehicle's current position. The desired velocity at the tracked point is to be tracked by the longitudinal controller. Similarly, the steering controller searches through the local path and identifies the desired point to track. Once the closest point is found the lateral and heading errors can calculated for steering controller.

4.1 Steering Control Background

The steering control methods can be essentially split into three groups:

- Geometric
- Dynamic
- Optimization based

4.1.1 Pure pursuit

The pure pursuit controller is among the simplest steering control methods that was used. The pure pursuit controller is based on the geometry of the bicycle model. The controller

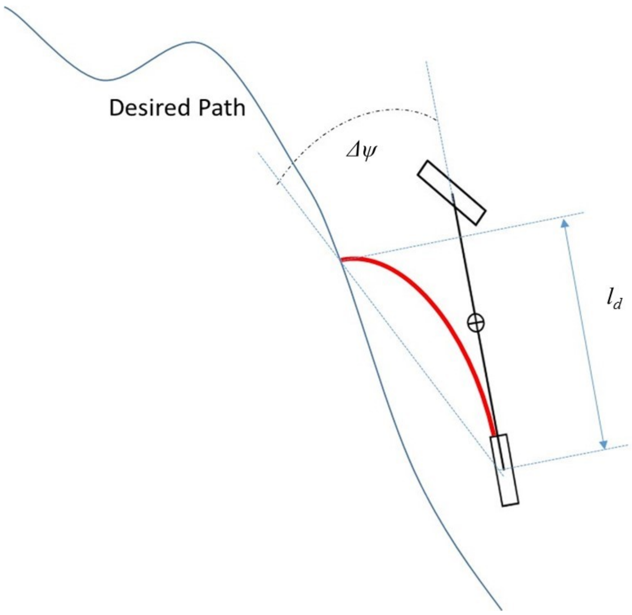

Figure 4.1: Pure Pursit Controller

focuses on the center of the rear axle and a point on the path that is l_d distance ahead of the vehicle as can be seen from Figure 4.1. The heading angle error between the vehicle model and road heading at the goal point are used to derive the control law. First, using the geometry of triangles and sine law from Figure 4.1 the following relation can be found:

$$\frac{l_d}{\sin(2\triangle\psi)} = \frac{R}{\sin(\frac{\pi}{2}) - \triangle\psi} \tag{4.1}$$

Equation (4.1) can be simplified further into Equation (4.2):

$$\kappa = \frac{2\sin(\triangle\psi)}{l_d} \tag{4.2}$$

such that κ in Equation (4.2) is curvature of the arc from the rear axle to the goal point as can be seen in Figure 4.1. Using geometry of the bicycle model from Equation (3.7) the steering angle can be calculated to be:

$$\delta(t) = \tan^{-1}(\frac{2L_{wb}\sin(\triangle\psi)}{l_d}) \tag{4.3}$$

By using the Equation (4.2) and dividing by velocity with proportional gain k_{sp} for scaled steering dynamics we get:

$$\delta(t) = \tan^{-1}\left(\frac{2L_{wb}\sin(\triangle\psi)}{k_{sp}v_x(t)}\right) \tag{4.4}$$

The velocity term in the denominator in Equation (4.4) can be very low and, therefore, result significant steering angle changes. This can be avoided by adding a constant V_s as can be seen in the following equation.

$$\delta(t) = \tan^{-1}\left(\frac{2L_{wb}\sin(\triangle\psi)}{k_{sp}v_x(t) + V_s}\right) \tag{4.5}$$

The local curvature in Equation (4.2) can be simplified using the relation of the heading error $\triangle\psi$ and lateral error at the tracked point e_{lat}.

$$\sin(\triangle\psi) = \frac{e_{lat}}{l_d}\kappa = \frac{2e_{lat}}{l_d^2} \tag{4.6}$$

The control steering law from Equation (4.4) can be expressed as:

$$\delta(t) = \tan^{-1}\left(\frac{2L_{wb}e_{lat}}{k_{st}v_x(t) + V_{const}}\right) \tag{4.7}$$

The constant k_{st} covers all the terms in Equation (4.6) except the lateral error.

4.1.2 Stanley Controller

The DARPA urban challenge has boosted the development of the autonomous driving and hence introduced multiple new methods and techniques for automated steering. Among them is the Stanley method [22] of the team that won the DARPA Urban challange. The Stanley Controller is designed for a bicycle model as seen in Figure 4.2. The steering law is based on reducing two errors: e_{lat} - cross track error and $\triangle\psi$ - heading error.

$$\delta(t) = \triangle\psi(t) + \tan^{-1}\left(\frac{k_{lat}e_{lat}(t)}{v_x(t) + k_c}\right) \tag{4.8}$$

A k_c is added at the denominator for numeric stability at low speeds. The terms under $\tan^1$ are almost identical to Pure Pursuit since the numerator in the Equation (4.4) can be represented using the lateral error. The difference is then the addition of the heading error between the heading of the vehicle and the road such that the heading of current closest point on the path directly feeds into the steering law.

4.1.3 LQR

Optimal control techniques can also be used for steering. In [19] an approach is described for using Linear Quadratic Regulator with dynamic bicycle model with an assumption of vehicle operating in the linear tire force region with constant cornering stiffness coefficient. The bicycle model and tire forces expressed in Equations (3.9), (3.12) (3.10) and (3.11)

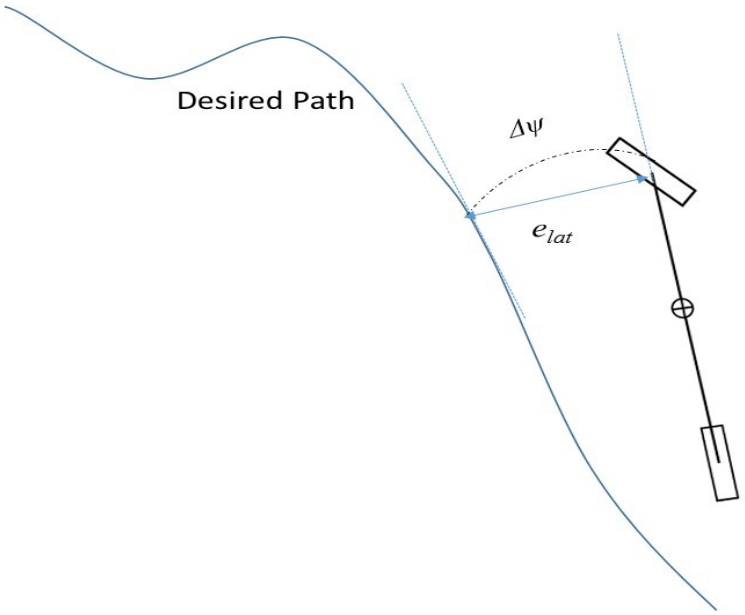

Figure 4.2: Stanley Controller

can be further summarized in the equations below (4.9) with front and rear lateral stiffness coefficients denoted as C_{cf},C_{cr}.

$$\begin{bmatrix} a_y \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} \frac{-(C_{cf}+C_{cr})}{m_v v_x} & \frac{l_r C_{cr} - l_f C_{cf}}{m_v v_x} - v_x \\ \frac{l_r C_{cr} - l_f C_{cf}}{I_{zz} v_x} & \frac{-(l_f^2 C_{cf} + l_r^2 C_{cr})}{I_{zz} v_x} \end{bmatrix} \begin{bmatrix} v_y \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} \frac{C_{cf}}{m_v} \\ \frac{l_f C_{cf}}{m_v} \end{bmatrix} \delta \tag{4.9}$$

Using the bicycle dynamic model the error dynamics of the vehicle with regard to the path model and using path yaw rate r can be represented in an affine form $\dot{X} = AX_e + B_1\delta + B_2 r$ as below.

$$\begin{aligned} \begin{bmatrix} \dot{e_{cg}} \\ \ddot{e_{cg}} \\ \dot{\theta}_e \\ \ddot{\theta}_e \end{bmatrix} &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-(C_{cf}+C_{cr})}{m_v v_x} & \frac{C_{cf}+C_{cr}}{m_v} & \frac{l_r C_{cr} - l_f C_{cf}}{m_v v_x} \\ 0 & 0 & 0 & 1 \\ 0 & \frac{l_r C_{cr} - l_f C_{cf}}{I_z z v_x} & \frac{l_f C_{cf} - l_r C_{cr}}{I_z z} & \frac{-(l_f^2 C_{cf} + l_r^2 C_{cr})}{I_z z v_x} \end{bmatrix} \begin{bmatrix} e_{cg} \\ \dot{e_{cg}} \\ \theta_e \\ \dot{\theta}_e \end{bmatrix} \\ &+ \begin{bmatrix} 0 \\ \frac{C_{cf}}{m} \\ 0 \\ \frac{l_f C_{cf}}{I_z z v_x} \end{bmatrix} \delta + \begin{bmatrix} 0 \\ \frac{l_r C_{cr} - l_f C_{cf}}{m_v v_x} - v_x \\ 0 \\ \frac{-(l_f^2 C_{cf} + l_r^2 C_{cr})}{I_z z v_x} \end{bmatrix} r(s) \end{aligned} \tag{4.10}$$

The term e_{cg} is the cross track error relative to the center of gravity of the car and θ_e is the heading error. Using the error state vector $X_e = (e_{cg}\ \dot{e}_{cg}\ \theta_e\ \dot{\theta}_e)^T$ The control law is summarized as:

$$\delta(k) = -KX_e = -k_1 e_{cg} - k_2 \dot{e}_c g - k_3 \theta_e - k_4 \dot{\theta}_e \tag{4.11}$$

The gain vector K can be found from minimizing the objective function in Equation (4.12) such that matrix Q is weight matrix for minimizing error state vector X_e and matrix R is the weight matrix to minimize the control effort δ.

$$J_{LQR} = \sum_{k=0}^{\infty} x^{\delta}(k) Q x(k) + \delta(k) R \delta(k) \tag{4.12}$$

The Riccati matrix that satisfies the objective function J_{LQR} can be sloved using discretized versions of matrix A and B denoted as A_d and B_d respectively.

$$P = A_d^T P A_d - A_d^T P B_d (R + B_d^T P B_d)^{-1} B_d^T P A_d + Q \tag{4.13}$$

The resulting Riccatti matrix P can be used to find the gain vector from the expression below.

$$K = (R + B_d^T P B_d)^{-1} B_d^T P A_d \tag{4.14}$$

4.1.4 Model Predictive Control

Another approach that utilizes the optimal control techniques is using the model predictive control method. In [2] an MPC approach was used for dynamic bicycle model and nonlinear tire model. Unlike the linear tire model introduced in Equation (3.12) , the nonlinear tire model covers the variations in coefficient of friction μ, takes into account the normal load force on the tire F_z^t and covers the region of tire curve where the force is saturated.

$$\begin{aligned} F_l &= f_l(\alpha, \lambda, \mu, F_z) \\ F_c &= f_c(\alpha, \lambda, \mu, F_z) \end{aligned} \tag{4.15}$$

The normal load for front and rear tires F_z is assumed to be constant and is derived from the load distribution relative to centre of gravity of the vehicle.

$$\begin{aligned} F_{z_f} &= \tfrac{bmg}{2(a+b)} \\ F_{z_r} &= \tfrac{amg}{2(a+b)} \end{aligned} \tag{4.16}$$

All of the vehicle dynamics Equation (3.9) - (3.15) are summarized in the differential equations below:

$$\begin{aligned} \dot{\upsilon} &= f_{s,\mu}(\upsilon, u) \\ \eta &= h(\upsilon) \end{aligned} \tag{4.17}$$

The input u is the steering angle δ and $\upsilon = [y,\ \dot{y},\ \dot{x},\ \phi,\ \dot{\phi},\ Y,\ X]^T$ is the state vector. The output map is given as follows:

$$h(\upsilon) = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \upsilon \tag{4.18}$$

The system dynamics is discretized using Euler method as follows:

$$\begin{aligned} &\upsilon(t+1) = f^{dt}_{s,\mu}(\upsilon(t), \triangle u(t)) \\ &\eta(t+1) = h(\upsilon(t)) \\ &u(k) = u(t-1) + \triangle u(t) \\ &u(t) = \delta_f(t) \end{aligned} \tag{4.19}$$

And the following cost function was used:

$$J(\upsilon(t), \triangle u_t) = \sum_{i=1}^{H_p} \left\| \hat{\eta}_{t+1,t} - \eta_{ref_{t+i,t}} \right\|^2_Q + \sum_{i=0}^{H_c-1} \left\| \triangle u_{t+i,t} \right\|^2_R \tag{4.20}$$

where $\triangle u_t = \triangle u_{t,t}, .., \triangle u_{t+H_C-1,t}$ is the vector for optimization over the prediciton horizon H_p and control horizon H_c, η_{ref} is the desired output, Q and R are the weight matrices. The given $J(\upsilon, \triangle u_t)$ is minimized subject to Equation (4.21)

$$\begin{aligned} &\upsilon_{k+1,t} = f^{dt}_{s,\mu}(\upsilon_{k,t}, \triangle u) \\ &\eta_{k,t} = h(\upsilon_{k,t}) \\ &k = t, ..., t + H_P \\ &\delta_{f,min} =< \triangle u_{k-1,t} =< \triangle\delta_{f,max} \\ &k = t, ..., t + H_C - 1 \\ &u_{k,t} = u_{k-1,t} + \triangle u_{k,t} \end{aligned} \tag{4.21}$$

The model used in the method is not linear and hence the authors used nonlinear optimization package NPSOL for optimizing the input vector at every iteration.

4.1.5 Potential Field

In [16] an automatic steering method was proposed for lanekeeping and path following based on an artificial potential field. The potential field is created by increasing the potential as the lateral distance and the heading error increases away from the center of the lane or a path. The energy represented below as V_c. The potential V_c increases as the vehicle moves away from the lane at lateral distance and deviates from the heading. The effect of lateral and heading error is increased by the look ahead distance x_{la} and potential field gain k_{pf}

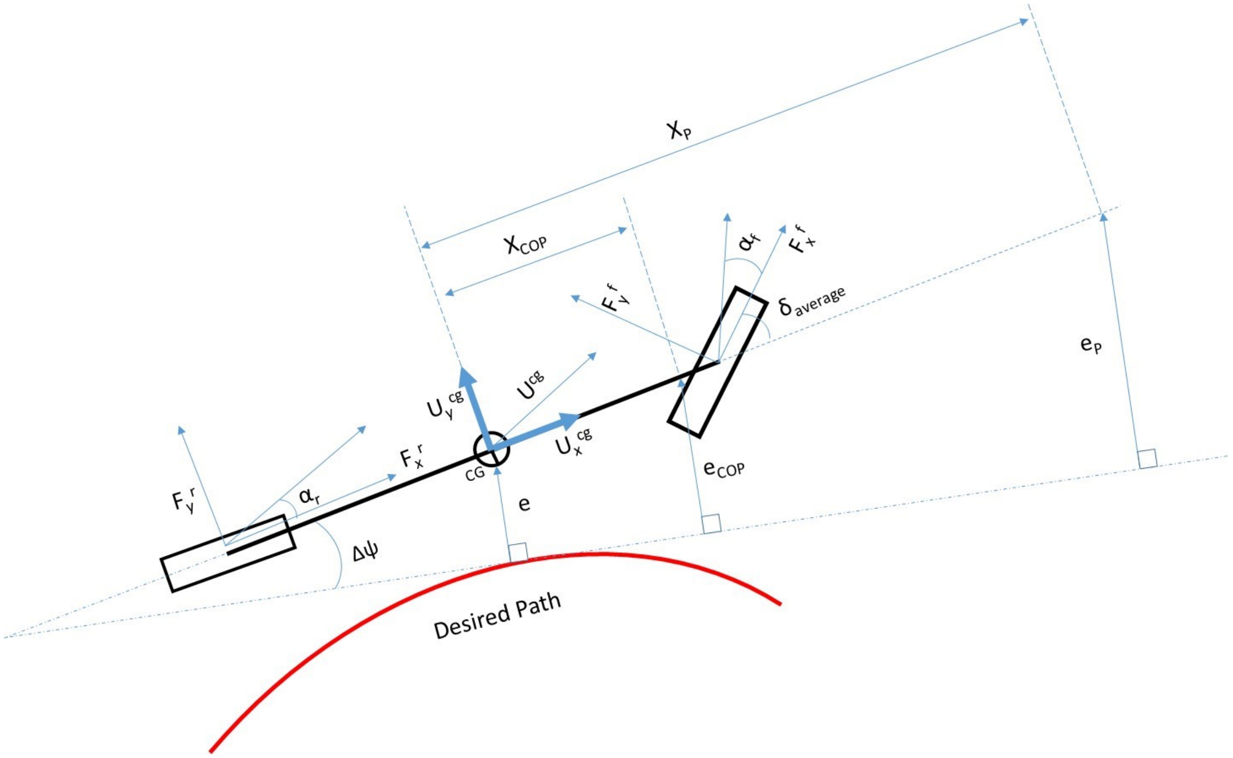

Figure 4.3: potential field controller diagram model

$$V_c = k_{pf}(e_{lat} + x_{la} \sin \psi)^2 \tag{4.22}$$

Deriving the virtual control force from the equation above by taking a derivative with respect to lateral error e_{lat} yields:

$$F_{pf} = -2k_{pf}(e_{lat} + x_{la} \sin \triangle\psi) \tag{4.23}$$

The force F_{pf} is the proportional force to maintain the vehicle in the middle of the lane. Using the force F_{pf} and assuming the vehicle is in the linear region of it's tire curve the desired steering angle can be found from dividing the force by the front lateral stiffness coefficient C_{cf}.

$$\delta = F_{pf}/C_{cf} \tag{4.24}$$

Since pure proportional gain controllers can not guarantee comfortable ride and may cause oscillations the force F_{pf} must be damped.

$$F_{pf} = -k_{LK}(e_{lat} + x_{la} \sin \triangle\psi) - k_d \triangle\dot{\psi} \tag{4.25}$$

where the k_d is the damping coefficient that must be tuned for comfortable ride.

4.2 Steering Control Architecture

The dynamic model based steering control methods, although designed for more detailed and higher fidelity models, require a larger amount of parameters to be known, C_r - tire cornering stiffness, C_l -longitudinal tire stiffness, a, b - distances to center of gravity from front and rear axles. The tire parameters are also functions of road friction and for different road surfaces will need to be modelled separately. Typically the modelling of the tire parameters is an expensive and time consuming process. In the given timeline for the CES 2017 and according to the budget tire parameter modelling was not possible. Therefore the steering methods described in [19] and [10], [11] were not chosen for implementation. Additionally, optimization based methods require the tire paramaters as well. Continuous optimization requires larger computational resources compared to the other methods and hence it gives us another reason not to use the LQR or MPC based controllers. As a result, since the speeds the demo was going to be taking place was considered to be relatively low and tire dynamics not going to play a significant role - geometry based steering control methods were chosen as desired method. According to [19] Stanley controller has significant advantages to Pure Pursuit as long as the path is smooth and the vehicle is driving at moderate speeds. It was decided, therefore, to implement the Stanley controller for CES 2017 demo with some minor modifications.

$$\delta(t) = k_{head}\triangle\psi(t) + \tan^{-1}\left(\frac{k_{lat}e_{lat}(t)}{v_x(t) + k_c}\right) + k_d\triangle\dot{\psi} \tag{4.26}$$

Namely damping term $k_d\triangle\dot{\psi}$ was added to reduce the oscillation due to heading error, k_{head} gain was added to scale down the heading error effect relative to other terms.

4.3 Longitudinal Control

Automated longitudinal control, whether for speed profile tracking or cruise control typically utilizes a structure as below:

$$a_x(t)_{des} = k_p(v_x - v_{x_{ref}}) + k_I\int_0^t (v_x - v_{ref})dt \tag{4.27}$$

where $a_x(t)_{des}$ is the desired longitudinal acceleration, $(v_x - v_{x_{ref}})$ is the velocity error, k_p and k_I are the proportional and integral gains accordingly. If the acceleration to brake or engine torque map is available it can be converted to corresponding throttle and brake pedal position. The use of the integral term in the speed control is introduced to cancel the effects of the disturbances such as: F_{aero} - aerodynamic drag force, F_{roll} - rolling resistance force, F_{turn} - is the longitudinal component of the steering angle during cornering and F_{grade} - the disturbance due to road grade. Since the given vehicle Linkoln MKZ 2017 has a complex hybrid gas-electric engine, continuously variable transmission and the throttle-engine output torque was not provided - the longitudinal control structure provided in [14] will need to be used with some modifications for final throttle pedal control input.

Dataspeed offers a baseline longitudinal controller with a proportional gain on speed and proportional-integral gain controller for throttle input.

$$\begin{aligned} &a_{cmd} = k_{thr}a_{cmd} + k_I \int_0^t (v_{x_{car}} - v_{ref})dt,\, a_{cmd} > 0 \\ &a_{cmd} = a_{cmd}m_v r_{wheel},\, a_{cmd} < 0 \end{aligned} \tag{4.28}$$

Throttle command keeps increasing over time until $(v_{x_{car}} - v_{ref})$ is zero or negative, in which case the brake is activated. The brake torque command, when published as a ROS topic, is converted into required brake pedal position via torque - pedal position map provided and implemented by Dataspeed, whereas the throttle command is published directly as a desired pedal percentage.

4.4 Controller Results

4.4.1 Steering Controller Results

The tracking performance can be summarized by three different maneuvers:

- converging to a path from a distance
- following a long curve
- following a tight swerve

The results of the autonomous platform's controller performance are generated from the real time data that was recorded on the vehicle's onboard computer via ROS topics.

The acceptable lateral error threshold was identified by the distance from the vehicle to the edge of the lane marking, such that the vehicle is centered along the path. Since the distance to the edge of lane marking on the Demo track was set to be 1m, the lateral error must not exceed that distance in order for the vehicle to stay in the lane at all times. Heading error was also identified by the lane marking, such that the vehicle if not positioned along the path would still stay within the lane markings. The heading error threshold was calculated to be 0.3 radians.

In Figure 4.4 the steering controller converges to a path by tracking the path heading and reducing the lateral error. This maneuver can demonstrate the performance of the controller when the vehicle is stopped on the sidewalk and needs to start driving and get on the path simultaneously. Since the proportional gain on the heading error term has

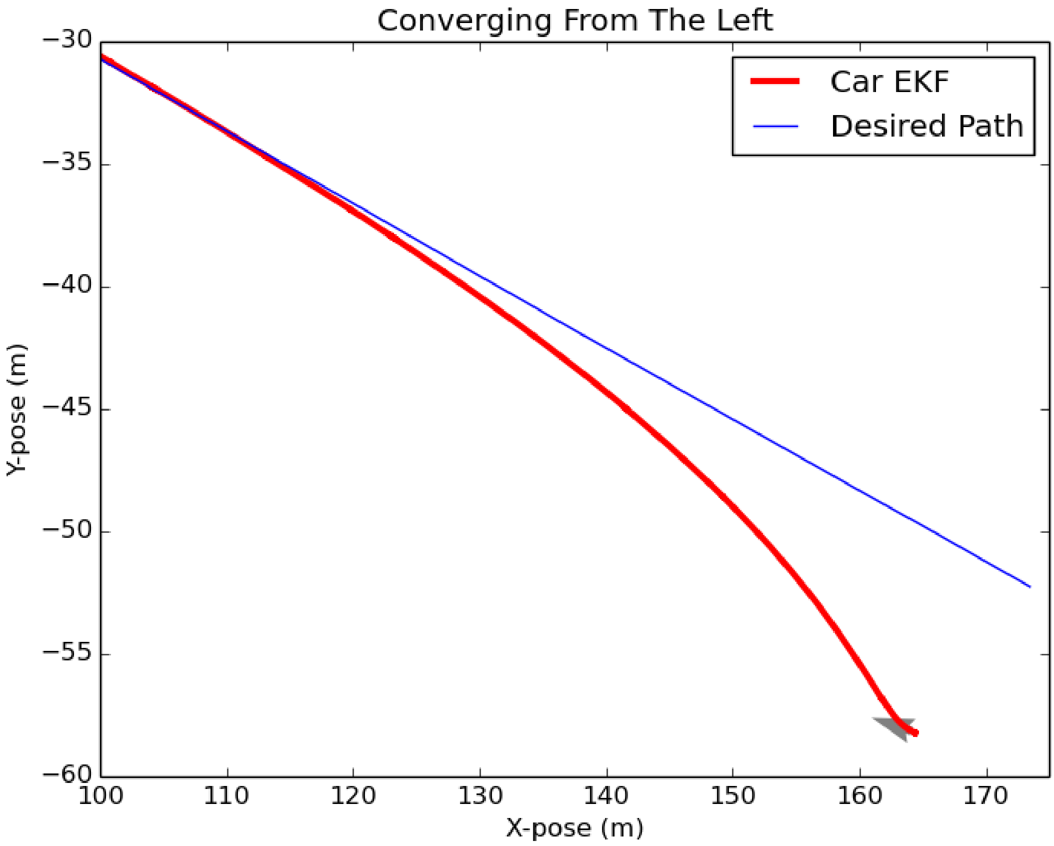

Figure 4.4: Converging From The Left

more significant influence compared to the lateral distance term, the vehicle converges from 15 meter offset initial position in approximately 60 meters. The vehicle converges to within few centimeters as may be seen in Figure 4.6 and performs the maneuver with about ± 0.35 radians of heading error. The heading error noise can be explained by the noisy measurements coming from the onboard IMU.

To improve lateral convergence the proportional lateral error gain k_{lat} can be increased. To see the combined effect of the lateral distance and heading error on the steering controller performance the tracking of the continuous arc was performed, as can be observed from Figure 4.7. The vehicle closely follows the path since the heading error gain $k_{head.}$ dominates in the steering control law, the lateral distance error is small at the initial position and the curved path only introduces continuously changing heading. The lateral error for curved path is within acceptable 1 meter threshold as can be seen in Figure 4.9 and the heading error stays within 0.1 - 0.2 radians with an exception of slight change of heading rate at 40 meters along the curve.

To examine the performance of the controller due to continuously varying heading and

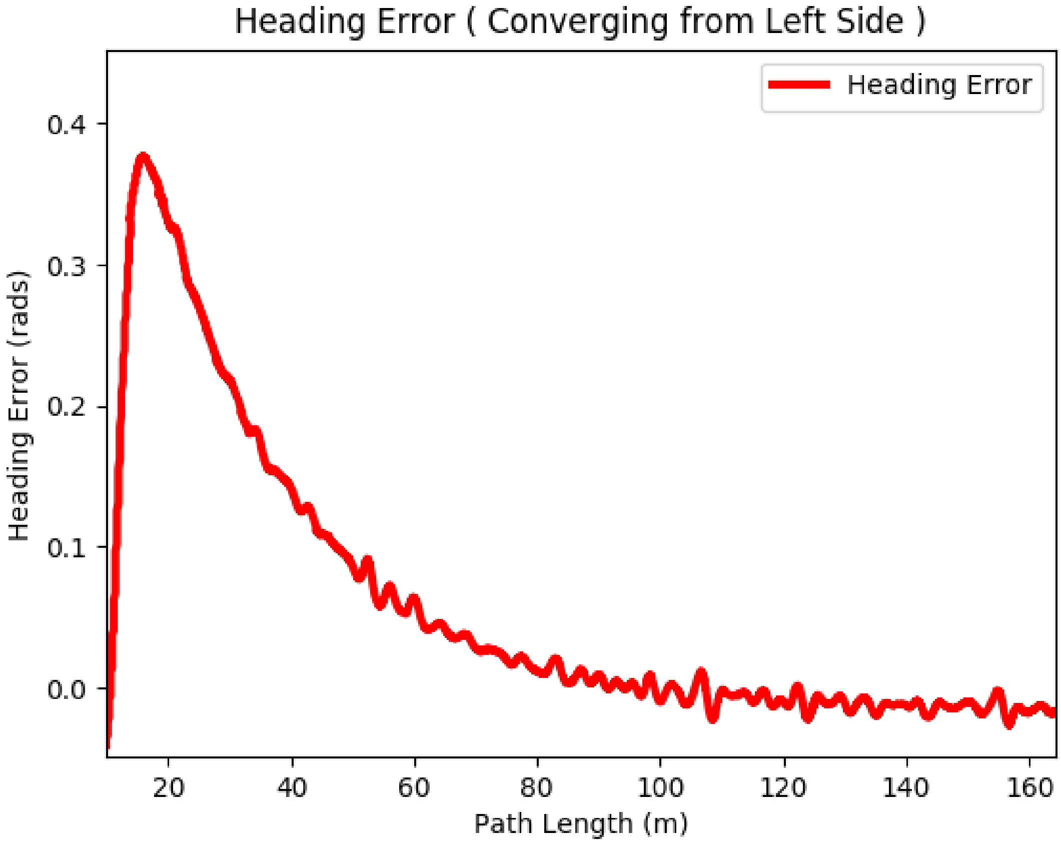

Figure 4.5: Heading Error (Converging From The Left)

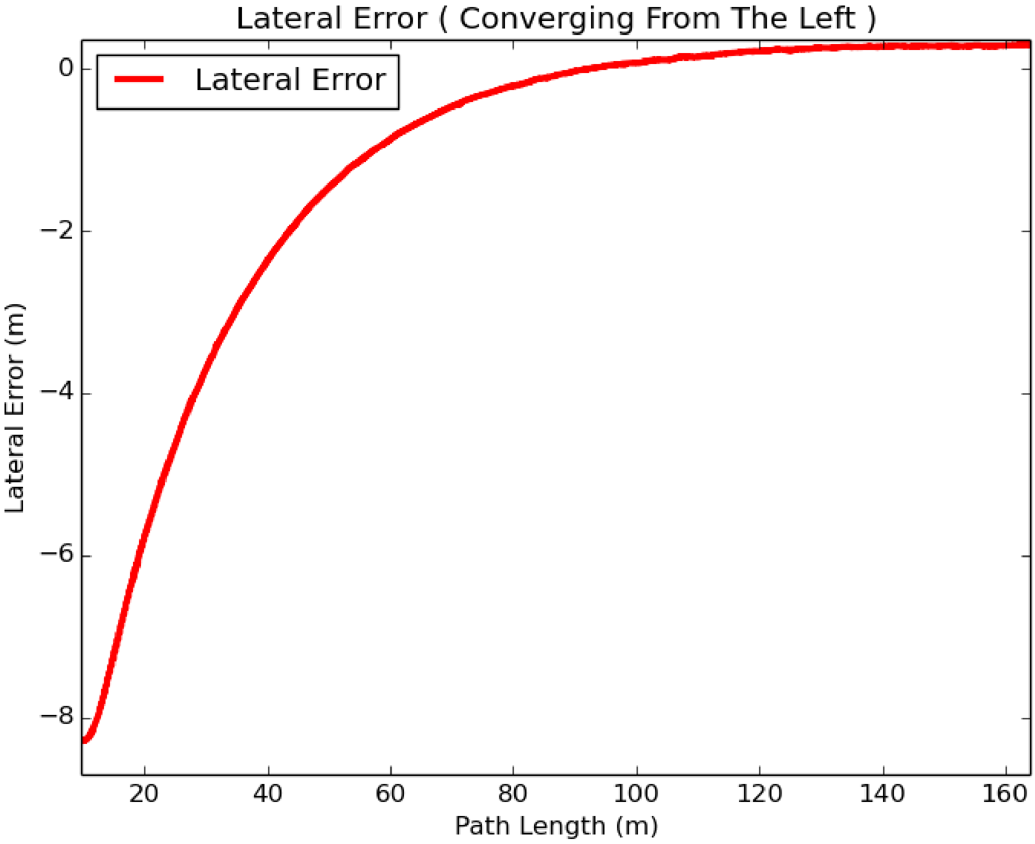

Figure 4.6: Lateral Error (Converging From The Left)

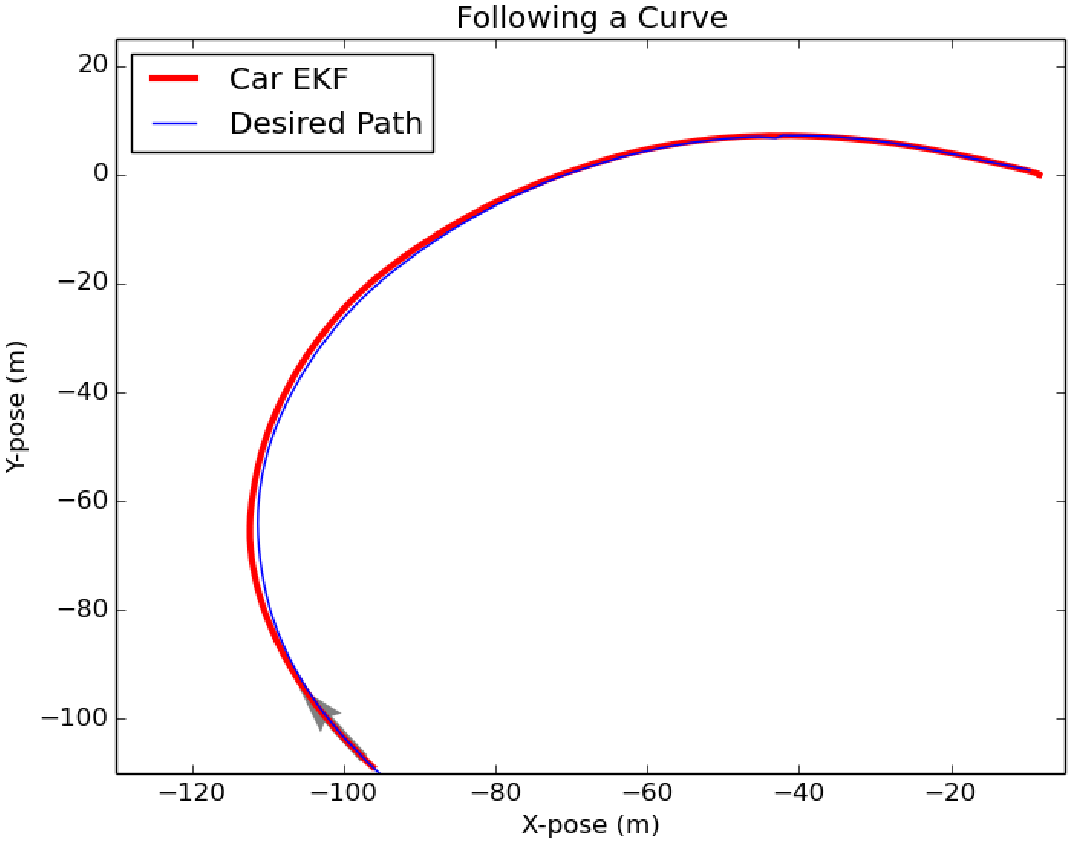

Figure 4.7: Following a Curve

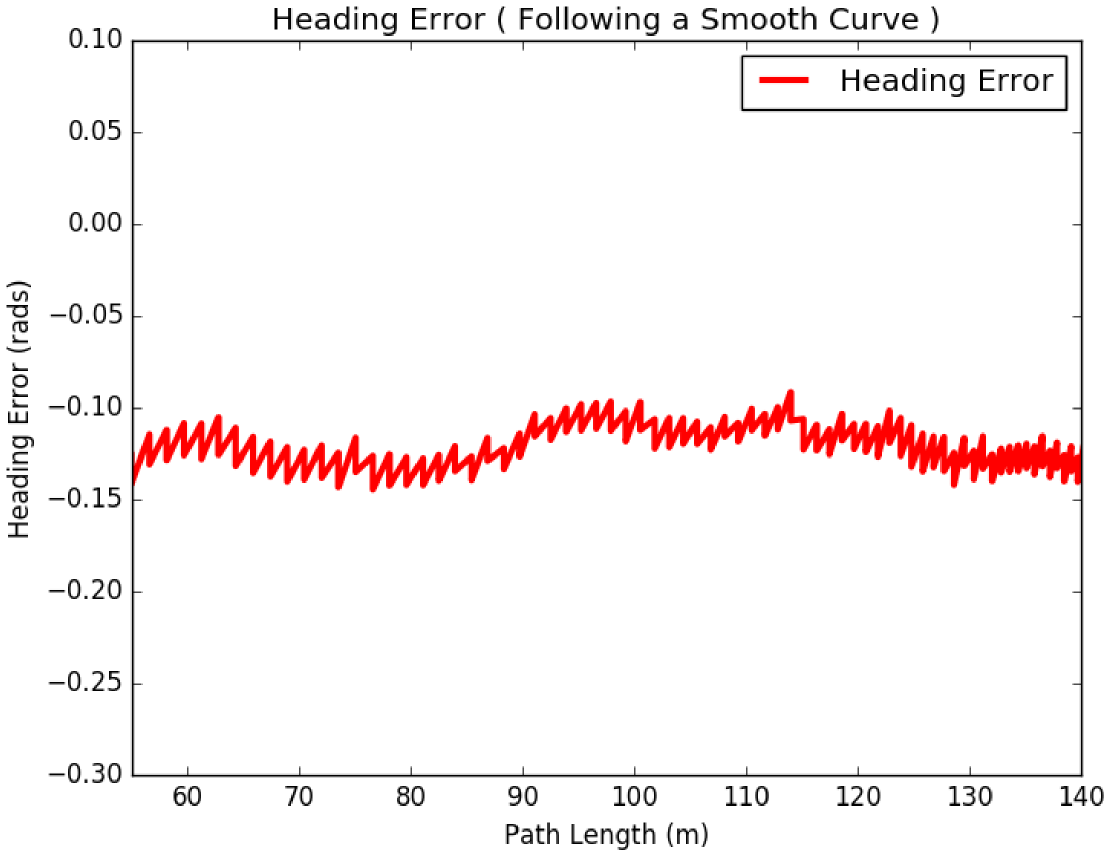

Figure 4.8: Heading Error (Following a Curve)

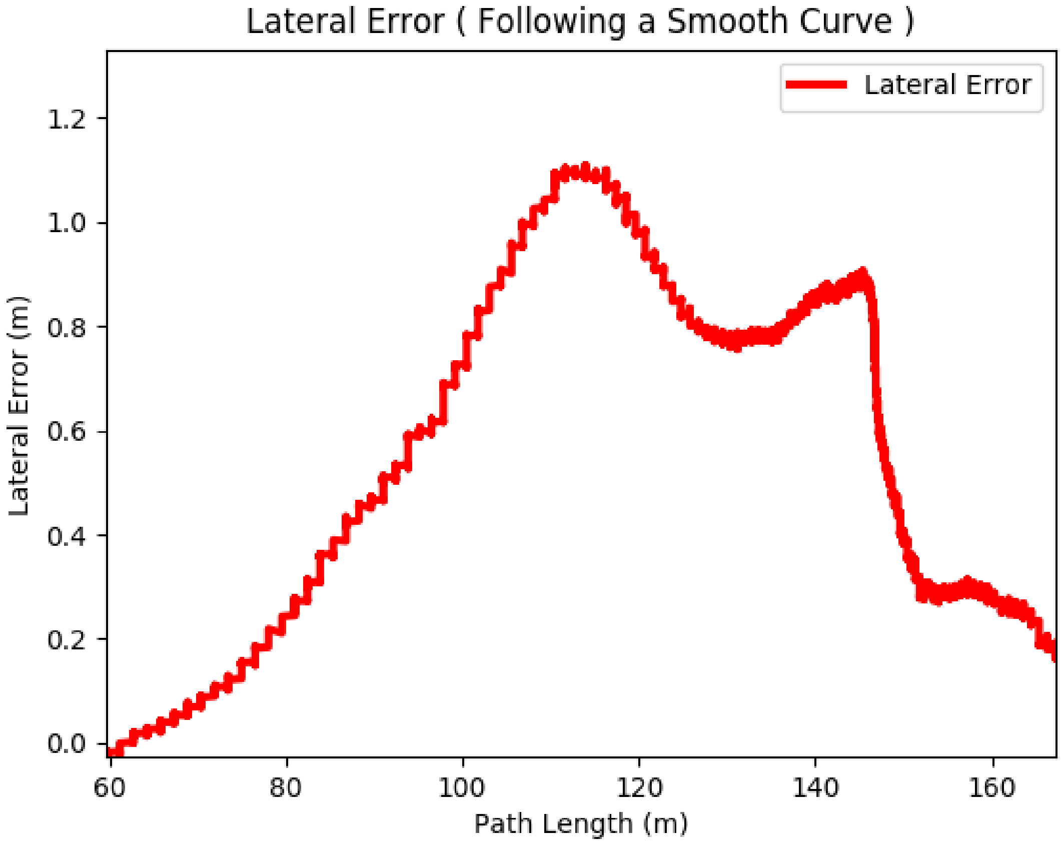

Figure 4.9: Lateral Error (Following a Curve)

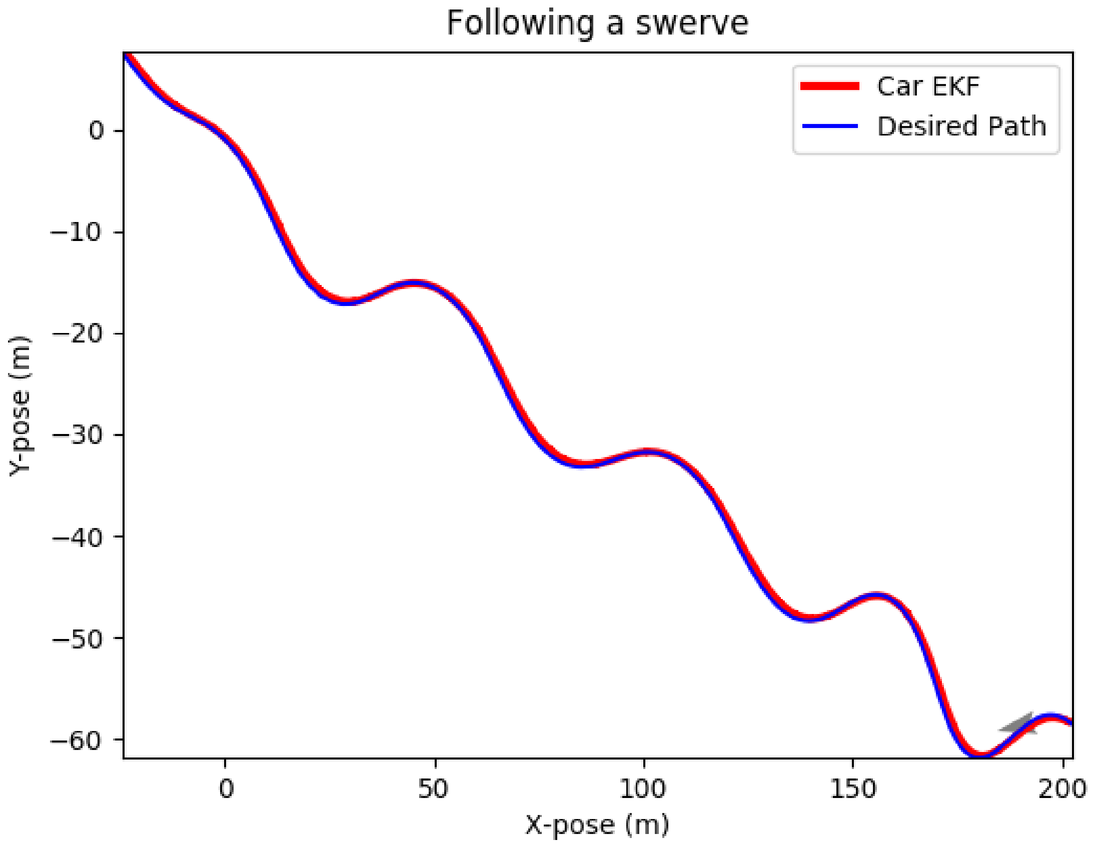

Figure 4.10: Following a Swerve

lateral distance - tracking results are presented for following a swerve in Figure 4.10. This maneuver can describe the performance of the controller in sudden maneuvers such as lane change or avoiding an obstacle. The controller performs well while tracking swerve with an average lateral distance error less than a meter and, therefore, keeps the vehicle within desired error threshold. The heading error stays within ± 0.1 radians and the lateral error is no more than 0.65 meters despite sudden changes in heading along the path as can be seen in Figures 4.11 and 4.12.

The steering controller tracking performance was within the required error limits that guaranteed the successful demo sequence completion.

4.4.2 Longitudinal Controller Results

The longitudinal controller performed within acceptable margins on a flat track. As can be observed from Figure 4.13 vehicle velocity tracks the desired profile very closely within 0.2 m/s error. The plot is generated using the swerve path and hence due to drag force

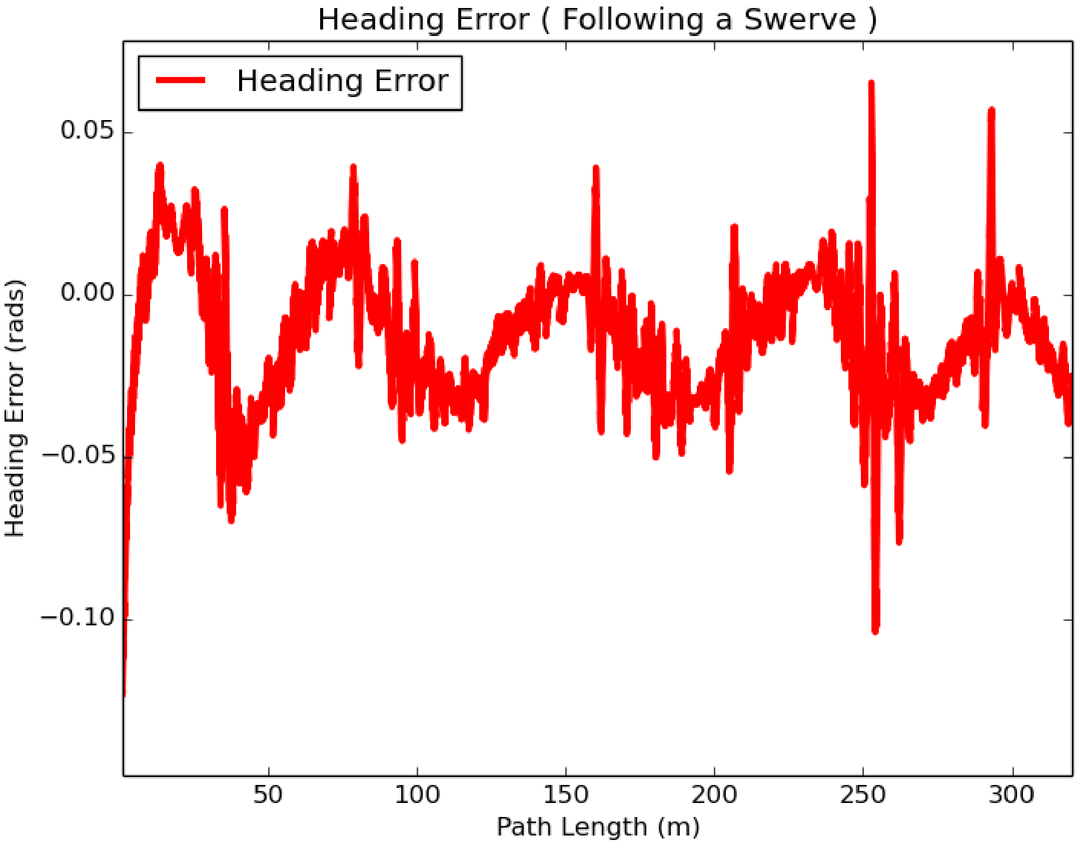

Figure 4.11: Heading Error (Following a Swerve)

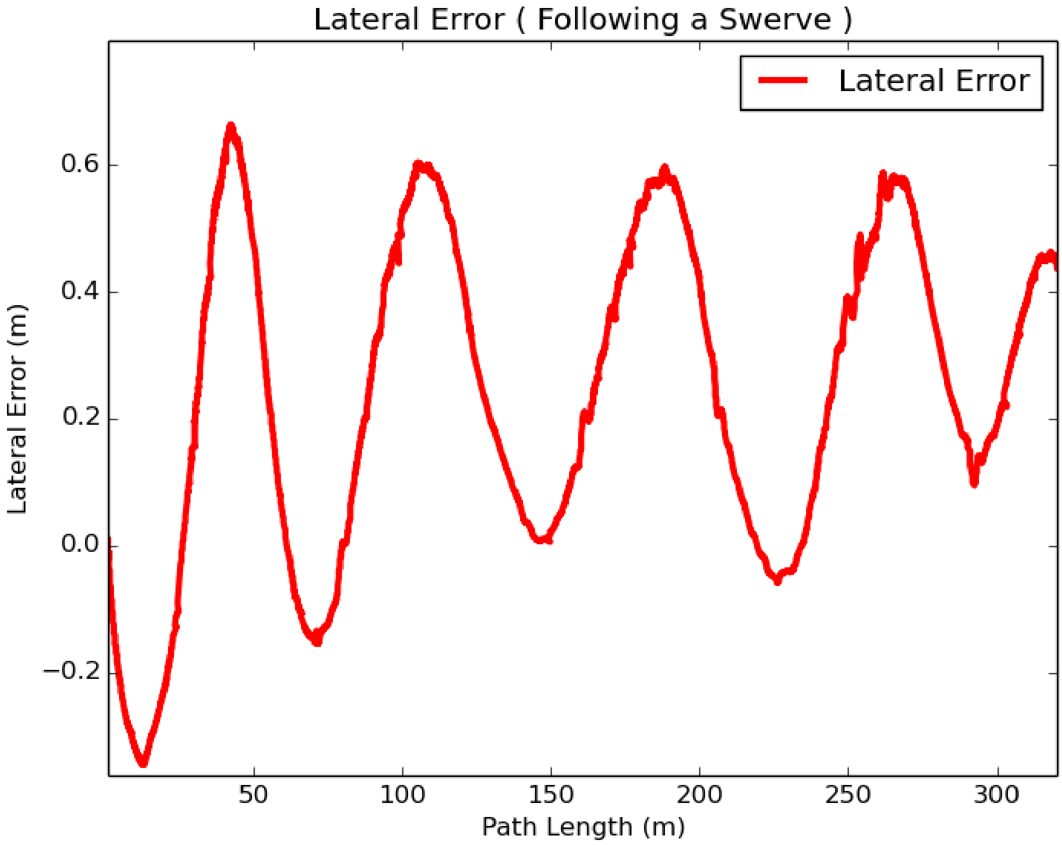

Figure 4.12: Lateral Error (Following a Swerve)

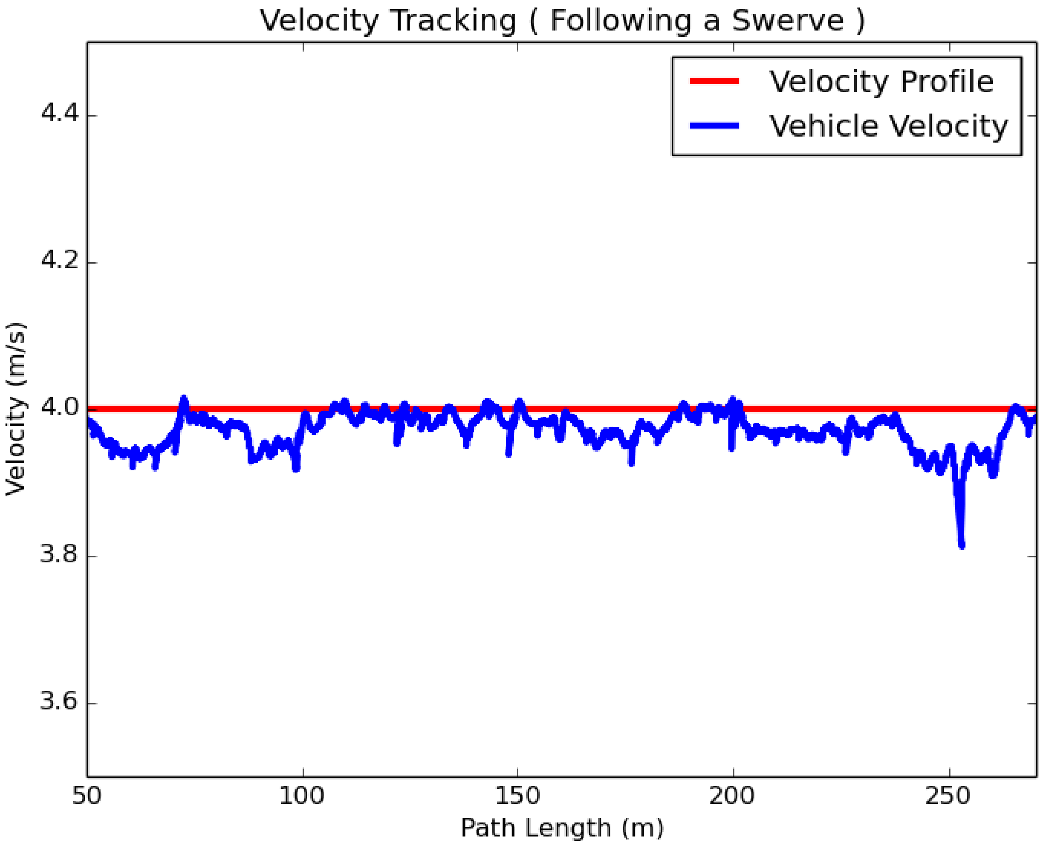

Figure 4.13: Velocity Profile Tracking

F_{turn} generated from the longitudinal component of the steering force the velocity profile fluctuates periodically.

The longitudinal controller performed well within the desired threshold as defined for the demo sequence. The limitations of the longitudinal controller can be eliminated after a complete powertrain system identification and a controller design that takes into account the drag forces acting on the vehicle and the forces available from the tire curve in low friction conditions.

Chapter 5

Aggressive Driving Controller

The controller architecture proposed in Chapter 4 and implemented for CES 2017 is feasible for low speed applications. For urban and highway driving the controller in Chapter 4 may not be safe for implementation since it does not account for tire dynamics and as a result, does not make sure the vehicle maintains the lateral force induced by steering actuator to be within acceptable tire limits. Similarly the longitudinal controller performed for CES 2017 in 4 does not maintain the tires within the allowable tire limits, as it only implements velocity tracking. To implement the autonomous driving in the urban or highway environment the controller needs to be designed using dynamic vehicle model for safety at higher speeds. To do so the vehicle control commands can be generated as longitudinal and lateral forces. The force commands can be converted into steering or throttle and brake commands. In [11] steering control method was proposed for tracking a path at the limits of friction. The controller generates the control lateral force that lies within the acceptable limits of tire force curve.

5.1 Steering Control Law

The total control steering force consists of two components: F_y^{fb} feedback force and F_y^{ff}.

$$F_y^{cmd} = F_y^{fb} + F_y^{ff} \tag{5.1}$$

Where F_y^{fb} is the feedback component and the F_y^{ff} is the feedforward lateral command force. The feedback steering force is calculated using the potential field method described in [16] and summarized in Chapter 4.

5.1.1 Feedforward Steering Force

The feedforward component of the steering control law is generated using the the concept of centre of percussion as described in [11]. The centre of percussion is the point along the

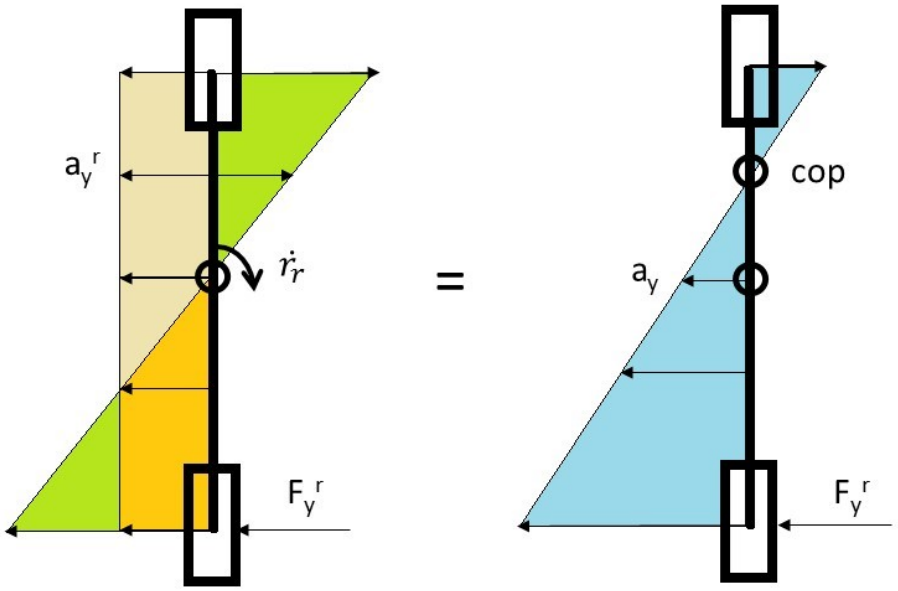

Figure 5.1: centre of percussion

longitudinal axis of the bicycle model at distance x_{cop} from cg, where the effect of lateral forces from the rear tires is canceled by the rotational acceleration such that:

$$\frac{F_y^r}{m_v} + x_{cop}\frac{-bF_y^r}{I_{zz}} = 0 \tag{5.2}$$

The feedforward force calculated at the centre of percussion reflects the effect of road geometry on the steering angle, similar to race driver steering in advance when reaching a corner. The derivation of the feedforward force starts with the expressing error dynamics of the potential field controller described in Subsection 4 as lateral error at the cg e,lateral error at centre of percussion e_{cop}, lateral distance at a lookahead distance from the cg, e_p, in vehicle's local coordinate frame.

$$\begin{aligned} e_p &= e_{lat} + x_p \sin(\triangle\phi) \\ \dot{e} &= v_y \cos(\triangle\phi) + v_x \sin(\triangle\phi) \\ \dot{s} &= v_x \cos(\triangle\phi) - v_y \sin(\triangle\phi) \end{aligned} \tag{5.3}$$

Distance traveled along the path s and its derivative can also be expressed in vehicle coordinates. Assuming small angles and that $\dot{v}_x\triangle\phi \approx 0$ the error dynamics can further be described as follows (5.4)

$$\begin{aligned}
&\triangle\dot{\phi} = \dot{\phi} - k\dot{s} \\
&\dot{e_p} = \dot{e} + x_p\triangle\dot{\phi} \\
&\triangle\ddot{\phi} = \ddot{\phi} - k\ddot{s} - \dot{k}\dot{s} \\
&\ddot{e_p} = \dot{v_y} + v_x\dot{\phi} - v_x k\dot{s} + x_p\ddot{\phi} - x_p(k\ddot{s} + \dot{k}\dot{s})
\end{aligned} \tag{5.4}$$

Using a substitution of $\dot{\phi}$ with an expression from Equation (3.9) and replacing the lateral acceleration in (3.9) with $a_y = \dot{v_y} + v_x\dot{\phi}$ the dynamics defining the evolution of the error,e_p, become:

$$\ddot{e_p} = \frac{F_y^f + F_y^r}{m_v} - v_x k\dot{s} + x_p\frac{aF_y^f - bF_y^r}{I_{zz}} - x_p(k\ddot{s} + \dot{k}\dot{s}) \tag{5.5}$$

Using the centre of percussion expression in (5.2), assuming the I_{zz} can be approximated by $I_{zz} = m_v ab$, assuming that the centre of percussion is at distance a from cg, and by equating (5.5) to zero the F_y^{ff} can be derived as:

$$F_y^{ff} = \frac{mb}{L_{wb}}(v_x k\dot{s} + x_{cop}(k\ddot{s} + \dot{k}\dot{s})) \tag{5.6}$$

5.1.2 Dugoff Tire Model

The total steering force can be converted to a desired steering angle by finding the desired slip angle from the tire curve if a nonlinear tire model or by dividing the total steering force by the cornering stiffness. In [21] a Dugoff tire nonlinear model was used for mapping the lateral control force to steering angle for better performance at the peak forces. The Dugoff tire model incorporates the effect of combined slip for lateral and longitudinal forces (5.7).

$$\begin{aligned}
&F_x = C_l\frac{\zeta}{1+\zeta}f(\lambda) \\
&F_y = C_c\frac{\tan(\alpha)}{1+\zeta}f(\lambda) \\
&\lambda = \frac{\mu F_z(1+\zeta)}{2\sqrt{(C_l\zeta)^2 + (C_c\tan(\alpha))^2}}
\end{aligned} \tag{5.7}$$

$$f(\lambda) = \begin{matrix} (2-\lambda)\lambda, & if \quad \lambda < 1 \\ 1, & if \quad \lambda \geq 1 \end{matrix} \tag{5.8}$$

Dugoff tire model is analytically derived and incorporates the reduction of both lateral and longitudinal tire forces as the tire approaches its force limits. Using the Dugoff tire model, therefore, offers more advantages over other methods for vehicle with front wheel drive powertrain, since the combined slip during acceleration can be impossible to account for with other tire models that assume no longitudinal forces.

5.2 Trajectory Planner

Multiple methods for nonholonomic trajectory planning exist in the autonomous driving literature. Some of the most successful ones are based on the following three major methods:

- Dubins Curves
- Clothoids
- Polynomial Spirals

5.2.1 Dubins Curves

In [15] an optimal path planning method is proposed that is based on utilizing the motion primitives such as a combination of straights and constant curvature arc segments. The minimum arc radius is defined by the maximum steering angle. A search is performed based on the desired final position and heading and starting configuration. Final result is a configuration of arcs and straight line segments that can take car-like robot to a desired position without violating the steering angle constraints. Multiple variations exist that utilize the planning method proposed in [15] and augment the final curves based on the required performance. In [5] Dubins curves are first used to find the admissible path through the cluttered environment using the optimal path planning technique proposed in [15] and later a sixth degree polynomial is fitted in each arc or straight segment using quadratic programming and minimizing the lateral and longitudinal jerk.

5.2.2 Clothoid

Another approach is using the elementary paths that consist of clothoid segments [17]. Clothoids are curves with linear function of their curvatures. Therefore instead of segments of constant curvatures like described in [15] the elementary paths utilize curve segments with linearly increasing or decreasing curvatures that allow for natural exit and entry to the cornering maneuver instead of sudden, discontinuous jumps in curvature that appear in Dubins curves. Elementary path segments can also be used in a search based method to find the best configuration in a nonholonomic motion planning problem. In [18] Dubins curves are first used to identify the optimal path to the desired position and heading, clothoid segments are then fit into the Dubins curve segments to replace the constant curvature arcs with linear function curvature clothoids. Bi-elementary clothoid segments were used for lane change maneuver at the limits of friction in [5] using the controller proposed in [10]. The final path is generated by the integrating of the curvature profile $k(s)$ (5.9):

$$\begin{aligned} x(s) &= x_0 + \int_0^{s_f} cos(\theta(s))ds \\ y(s) &= y_0 + \int_0^{s_f} sin(\theta(s))ds \\ \theta(s) &= \theta_0 + \int_0^{s_f} k(s)ds \end{aligned} \tag{5.9}$$

5.2.3 Polynomial Spiral

The clothoids can not have second order continuity at the point of junction of two segments since the curvature is a first order linear function. The curvature has a direct effect on the steering wheel angle and the smoothness of the curvature profile, therefore, ride comfort and can be defined to create a more natural steering angle dynamics. In [9] a polynomial spiral method is proposed for nonholonomic motion planning. The curvature of a polynomial spiral is an n-degree polynomial. The coefficients of a polynomial are used as parameters for optimization. In [6] a state space sampling planner was introduced that generates multiple feasible solutions that follow the center line or a middle of a lane with different offset distances, therefore allowing for obstacle avoidance along the centreline. In [13] a state lattice method was used that comprises of multiple smaller segments or predefined polynomial spirals and a search for shortest possible path that is built by small segments can be done. Additionally in [23] cubic and quartic polynomials were used where the first derivatives match at the point of junction of the spiral segments for smoother controller input. Polynomial spirals are better suited to trajectory planning in complex environments, as they provide more flexibility in defining the curvature, and enable minimization of higher order derivative objectives such as lateral jerk.

The polynomial spiral was chosen for implementation of the aggressive driving controller due to all of the above-mentioned advantages. The path can be generated similar to (5.9) by using the quadrature rule for integration. The trajectory planning method described in [9] generates the optimal path from an initial vehicle state $x_{in} = (x(0), y(0), \theta(0)) \in \mathbb{R}^3$ to a desired state x_{des}. The vehicle desired path $\boldsymbol{x} = (x, y, \theta, k)$ consists of a position x and y, heading angle θ and path curvature k. Each component of the path, including the curvature profile, is parameterized along the arc length s. The curvature is chosen to be represented by a cubic polynomial function $k(s) : \mathbb{R} \to \mathbb{R}$. The four polynomial coefficients (a, b, c, d) and final arc length s_f of the curvature profile $k(s)$ are used as optimization parameters in generating the desired path.

$$k(s) = a + bs + cs^2 + ds^3 \tag{5.10}$$

The curvature profile can be re-parameterized using knots, $\boldsymbol{p}$, that lie equidistantly along the length of the path [13].

$$\boldsymbol{p} = \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{pmatrix} = \begin{pmatrix} k(\frac{s_f}{3}) \\ k(\frac{2s_f}{3}) \\ k(s_f) \\ s_f \end{pmatrix} \tag{5.11}$$

The path components $(x(s), y(s), \theta(s))$ are generated by integrating the curvature to get the heading and integrating the heading to get the Euclidean position as shown in Equation (5.12).

$$\begin{aligned} x(s) &= x_0 + \int_0^{s_f} \cos(\theta(s))ds \\ y(s) &= y_0 + \int_0^{s_f} \sin(\theta(s))ds \\ \theta(s) &= \theta_0 + \int_0^{s_f} k(s)ds \end{aligned} \tag{5.12}$$

The position integral in this case is known as the Fresnel Integral and only has a numerical solution. In [9], a quadrature rule was proposed for calculating the integral and generating the path.

The path planner can be defined as a nonlinear optimization subject to multiple constraints. The objective function for the optimization can vary depending on the design of the planner and desired behavior. The optimization is constrained by the maximum and minimum curvature bounds that represent maximum vehicle steering angle and by desired final position, heading and curvature. If the optimization parameters can be denoted as $\boldsymbol{p} = (\ p_1 \quad p_2 \quad p_3 \quad s_f\ )$, then the constraint function, $\boldsymbol{g}(p)$, can be expressed using boundary condition, $\boldsymbol{h}(\boldsymbol{p})$, as shown in Equation (5.13):

$$\boldsymbol{g}(p) = \boldsymbol{h}(p) - \boldsymbol{x_f} = 0 \tag{5.13}$$

To find the smooth path from initial state $\boldsymbol{x}_{in}$ to final state $\boldsymbol{x}_f$ with upper and lower boundaries of the curvatures defined by $\pm k_{max}$ the following optimization problem needs to be solved (5.14):

$$\begin{aligned} &\textbf{min} \qquad J = \tfrac{1}{2} \textstyle\int_0^{s_f} k(p)^2 \\ &\text{s.t.} \quad \boldsymbol{g}(p) = \boldsymbol{h}(p) - x_f = 0 \\ &\qquad\qquad -k_{max} \leq \boldsymbol{p} \leq k_{max} \end{aligned} \tag{5.14}$$

This basic formulation of the path planner optimization can be extended to include static and dynamic obstacle constraints, and can be integrated into hierarchical motion planners [**?**].

5.3 Velocity Profile

Once the path planner generates a curvature profile, $k(s)$ and path to the desired final state $(x(s), y(s), \theta(s))$ the controller must execute it with minimum tracking error. It will be assumed the controller aims to drive the vehicle at the maximum possible velocity, given the desired path is chosen as was done in [**?**] and [13].

Velocity profile generation is based on the slip circle. The longitudinal $a_x(s)$ and lateral $a_y(s)$ acceleration along the path must be bounded by the friction limit of the road which is expressed as a product of gravitational force g and coefficient of friction μ(5.15).

$$(\mu \cdot g)^2 = a_x(s)^2 + a_y(s)^2 \tag{5.15}$$

At the point of maximum curvature, the longitudinal acceleration is set to zero and the vehicle corners at a constant velocity. Assuming steady state cornering the maximum velocity is found from acceleration which is proportional to curvature $k(s)$ and square of longitudinal velocity $v_x(s)$ (5.16).

$$\mu \cdot g = a_y(s) \approx k(s)v_x(s)^2 \tag{5.16}$$

In the path segments preceding the local maximum and following the next local maximum, however, the longitudinal acceleration does not need to be constrained to zero. Instead, the longitudinal acceleration can be expressed as a function of velocity, as follows.

$$a_x(s) = \frac{dv_x}{dt} = \frac{dv_x(s)}{ds} \cdot \frac{ds}{dt} = \frac{dv_x(s)}{ds} \cdot v_x(s) \tag{5.17}$$

Therefore, the velocity profile can be defined by a nonlinear differential equation.

$$\left(\frac{dv_x(s)}{ds} \cdot v_x(s)\right)^2 = (\mu \cdot g)^2 - k(s)^2 \cdot v_x(s)^4 \tag{5.18}$$

The left hand side of Equation (5.18) can be rearranged using forward differencing. The velocity at the current path segment is expressed as u_s and $u_{s+\triangle s}$ is the velocity at distance $\triangle s$ along the curve.

$$\frac{dv_x(s)}{ds} \cdot v_x(s) \approx \frac{\triangle v_x(s)}{\triangle s} \cdot v_x(s) = \frac{v_{s+\triangle s} - v_s}{\triangle s} \cdot v_s \tag{5.19}$$

The right hand side of Equation (5.18) represents the available longitudinal acceleration at a given point on the curve with the current curvature. When moving from the maximum curvature segment towards the lower curvature segment the vehicle should be able to accelerate but within the boundaries of the curvature of the next segment. Therefore an allowable acceleration and velocity must be found such that they do not violate the slip circle defined by Equation (5.18). By using the numeric approximation in (5.19) and assuming we start at the point of maximum curvature the following equation can be used for generating the velocity profile.

$$v_s \cdot \frac{v_{s+\triangle s} - v_s}{\triangle s} = \sqrt{(\mu \cdot g)^2 - k^2_{s+\triangle s} \cdot v^4_s} \tag{5.20}$$

where $k_{s+\triangle s}$ is the curvature at the new path segment and v_s is the velocity at the previous step. Solving for $v_{s+\triangle s}$ results in,

$$v_{s+\triangle s} = v_s \pm \triangle s \cdot \frac{\sqrt{(\mu \cdot g)^2 - k^2_{s+\triangle s} \cdot v^4_s}}{v_s} \tag{5.21}$$

The terms under the square root represent the available longitudinal acceleration if the vehicle was to travel at the previous speed on a new segment with a smaller curvature $k_{s+\triangle s}$. Depending on requirement, whether it is maximum acceleration or braking, the allowable longitudinal acceleration $\triangle s \cdot \frac{\sqrt{(\mu \cdot g)^2 - k^2_{s+\triangle s} \cdot v^4_s}}{v_s}$ is either added or subtracted from the initial velocity.

The resulting velocity profile can be used to calculate the maximum allowable acceleration profile for a given curvature profile using Equation (5.18). In order for the velocity profile planner to decide whether the obstacle avoidance maneuver should be performed with acceleration or deceleration, two profiles can be generated as two boundaries for possible velocity profiles. For braking, the given numeric approximation can be done until $v_{s+\triangle s}$ reaches zero - full stop. Since at every step the velocity is decreasing and no risk of violating the slip circle will occur even if the curvature is increasing, the profile generation for braking doesn't require a reverse integration step. If the calculation starts at the peak curvature then at every step the velocity profile will generate the maximum possible velocity with an acceleration that doesn't violate the slip circle. The polynomial spirals however can have up to two maximas and up to three different zeros and since the velocity only depends on the magnitude of the curvature finding the maximas and minimas of $|k(s)|$ would require a search through the polynomial in the domain of interest. To generate the maximum velocity profile forward and backward integration must be performed using (5.21). An algorithm for velocity profile generation is defined in Algorithm 1.

Algorithm 1 Velocity profile generation

```
forward integration
$N = \frac{s_f}{\triangle s}$ size of the path vector
for $i = 1 : N$ do
    if $(\mu \cdot g)^2 > k_{i+1}^2 \cdot v_i^4$ then
        $v_{i+1} = v_i + \triangle s \cdot \frac{\sqrt{(\mu \cdot g)^2 - k_{i+1}^2 \cdot v(i)^4}}{v(i)}$
    else $v_{i+1} = \frac{\mu \cdot g}{k_{i+1}}$
    end if
end for
reverse integration
for $i = N : 1$ do
    if $(\mu \cdot g)^2 > k_{i-1}^2 \cdot v_i^4$ then
        $v_{i-1} = v_i + \triangle s \cdot \frac{\sqrt{(\mu \cdot g)^2 - k_{i-1}^2 \cdot v(i)^4}}{v(i)}$
    else $v_{i-1} = \frac{\mu \cdot g}{k_{i-1}}$
    end if
end for
```

5.4 Longitudinal Control

The longitudinal controller finds the desired tracking velocity according to the velocity profile and the vehicle position as described in 4. Due to actuation delays and dynamic response of the vehicle tires the longitudinal controller in [10] uses lookahead distance for velocity profile tracking. The longitudinal controller used by [10] has only proportional gain terms, since the vehicle is supposed to drive at the limits of friction and introducing the integral gain term for velocity tracking may keep increasing the error even when the tire force is saturated, therefore the controller might get unstable overtime. The longitudinal force command is calculated by summing multiple force components.

$$F_x = F_x^{speed} + F_x^{slip} + F_{drag} \tag{5.22}$$

The F_{drag} consists of multiple terms that summarize the resistance forces acting on a vehicle 5.23.

$$F_{drag} = F_{rolling} + F_{aero} + F_{grade} + F_{turn} \tag{5.23}$$

The F_{aero} is the aerodynamic drag and can be calculated from $F_{aerodyn} = \frac{C_d A \rho v_x^2}{2}$. F_{roll} is the rolling resistance that can be assumed constant. $F_{grade} = -mg\sin(\theta_{grade})$ is the force due to road grade angle θ_{grade}, $F_{turn} = m\frac{b}{a+b}a_y|\tan(\delta)|$ is the longitudinal component of the steering.

5.4.1 Speed Tracking Feedback Controller

The virtual speed tracking force is calculated using a proportional gain K_{speed} and velocity error between the vehicle speed v_x and desired speed from the velocity profile v_{des}:

$$F_x^{speed} = K_{speed}(v_{des} - v_x) \tag{5.24}$$

The desired speed, v_{des}, is identified using the preview distance. The proportional gain K_{speed} is identified from experiments based on the vehicle and powertrain parameters. Since the goal of the aggressive controller is to keep at the vehicle at the edge of the slip angle there is no damping term in the F_x^{speed}.

5.5 Feedforward Longitudinal Controller

The feedforward force is calculated using the acceleration estimated by using acceleration profile from 5.17 after the velocity profile has been generated and the vehicle total mass m_v according to Newton's Law.

$$F_x^{ff} = m_v a_x(s) \tag{5.25}$$

5.5.1 Slip Circle Feedback Control

The slip circle is the representation of the available tire force during combined lateral and longitudinal slipping. The slip circle allows the controller to identify the available longitudinal force when the vehicle is cornering at the peak forces. The purpose of using the slip circle feedback is not to make a vehicle drift, but rather to keep it within the friction limit of the tire. The slip circle feedback force F_x^{slip} is calculated using the normalized slip circle diagram technique proposed in [10]. Two axes of the slip circle are the longitudinal slip ratio and lateral slip angle normalized by the peak values based on the tire force curve. The slip circle feedback force is only implemented on the longitudinal controller to make sure the steering controller always has maximum available lateral force to keep the vehicle on the path. The slip circle feedback force is summarized in the following equations 5.26:

$$F_x^{slip} = \begin{array}{ll} K_\zeta \triangle\bar{\zeta} + K_\alpha \triangle\bar{\alpha}, & \bar{\zeta} \leq 0 \\ -K_\zeta \triangle\bar{\zeta} - K_\alpha \triangle\bar{\alpha}, & \bar{\zeta} > 0 \end{array} \tag{5.26}$$

$$\begin{gathered} \triangle\bar{\zeta} = \begin{array}{ll} |\bar{\zeta}| - \sqrt{1 - \bar{\alpha}^2}, & \bar{\alpha} \leq 1 \\ |\bar{\zeta}|, & \alpha > 1 \end{array} \\ \\ \triangle\bar{\alpha} = \begin{array}{ll} 0, & \bar{\alpha} \leq 1 \\ |\bar{\alpha}| - 1, & \bar{\alpha} > 1 \end{array} \end{gathered} \tag{5.27}$$

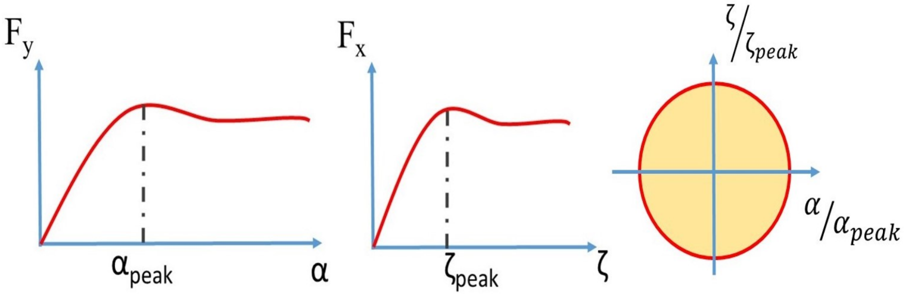

Figure 5.2: Tire curves and Slip Circle

The slip circle feedback force is calculated using the deviation away from the slip circle on the normalized longitudinal slip $\triangle\bar{\zeta}$ and normalized lateral slip angle $\triangle\bar{\alpha}$ axes and multiplying the corresponding gains K_ζ and K_α. The normalization is done based on the peak values of the lateral slip angle and longitudinal slip from the tire curve as can be seen from Figure 5.2. The force F_x^{slip} is positive if the slip is negative, which is similar to letting go off the brakes when the tires start becoming saturated. When the slip is positive, however, the F_x^{slip} is negative, which is similar to letting go off the throttle when wheels start slipping longitudinally. The deviation from the slip circle $\triangle\bar{\zeta}$ is equal to absolute value of slip ratio when the lateral slip angle passed the peak - the feedback force must restore the lateral force for steering. When the slip angle is less than the peak value the distance $\triangle\bar{\zeta}$ is equal to the deviation towards the edge of the circle. However, the deviation $\triangle\bar{\alpha}$ is zero when the slip angle is less than the peak value - allowing for the slip circle feedback force to be fully proportional to the slip ratio. If the slip angle is larger than the peak value $\triangle\bar{\alpha}$ is equal to the distance towards the edge of the slip circle. The final proportional force might get unstable if the vehicle is in combined slip and exceeds the peak values. In [21], the final force is, therefore, limited to a maximum traction force available F_{avail} 5.28.

$$F_x^{avail} = \sqrt{\frac{\mu m_v g b^2}{L_{wb}} - F_y} \tag{5.28}$$

During the combined slip the maximum traction force, F_x^{avail}, may not return the actual available force due to nonlinear behavior of the tire forces. Calculating the maximum available force using the Dugoff tire model instead offers a higher fidelity estimate of the available longitudinal force.

$$F_x^{avail} = C_\sigma \frac{\zeta}{1+\zeta} f(\lambda) \tag{5.29}$$

5.6 Simulation Results

The simulation was implemented on a high fidelity vehicle simluation software CarSim. To demonstrate the capabilities of the polynomial spirals, a smooth double lane change

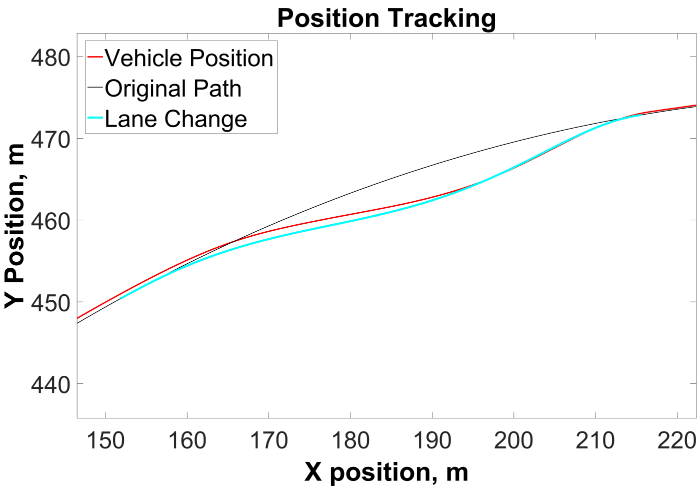

Figure 5.3: Position Tracking

maneuver is implemented from a single set of parameters while the vehicle is already driving at top available speed along a larger polynomial spiral segment at maximum speed. The position tracking results for lane change maneuver had the largest cross track error of 0.7m at the entry to the lane change curve and less than 0.01 m towards the exit as can be observed in Figure 5.3 and Figure 5.7. The heading error stays within 0.04 radians despite driving at the tire limits, which can be explained by the F_y^{ff} term in the lateral controller that allows to react to quickly changing curvature along the path by making a turn in advance. The potential field feedback force F_y^{fb} on the other hand, ensures that the vehicle stays within the lane and keeps the lateral error margin less than a 1 m at all times, despite the tire force saturation in the lateral direction. While having feasible position tracking the controller also keeps the vehicle within the road friction limits. In Figure 5.4 the vehicle acceleration is plotted with a circle of radius equal to a road friction capacity μg, and the vehicle stays within the vicinity of the circle during reference path following as well as during lane change. Since the longitudinal controller increases the control effort as vehicle starts deviating from the normalized tire force circle, the lateral acceleration readings become denser as they move away from the edge of the slip circle indicating the effect of the slip circle feedback force. The front tire average slip shown on the g-g diagram in Figure 5.5 stays in the vicinity of the unit circle on the lateral slip angle axis $\dfrac{\alpha}{\alpha_{peak}}$ since the longitudinal controller gives preference to having maximum lateral force available for stable steering. On the longitudinal slip axis $\dfrac{\zeta}{\zeta_{peak}}$, however, the slip ratio leaves the circle while accelerating as well as braking with bigger deviation from the slip circle compared to the lateral slip angle axis. One of the reasons for this is the look ahead distance on the feedback controller that keeps the vehicle speed at an offset from the velocity profile and, therefore, vehicle ends up being above the allowable vehicle speed limit as can be seen from Figure 5.6. The lookahead distance must be adjusted

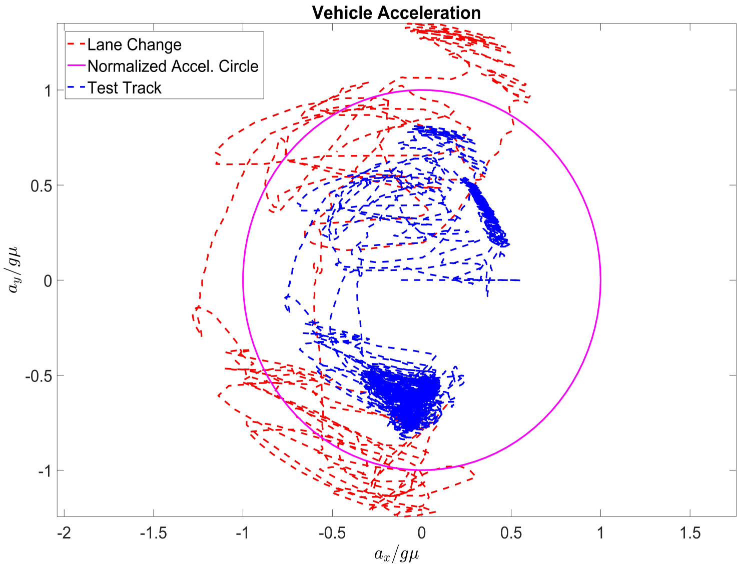

Figure 5.4: Acceleration against slip circle

according to the curvature slope to avoid the longitudinal controller overshoots.Another possible improvement could be an introduction of the velocity smoothening step, such that a smooth polynomial profile can be fit and optimized with constrained acceleration profile and, therefore sharp spikes and steep slopes on the speed profile can be reduced prior to generating longitudinal actuation command. Overall velocity tracking performance keeps the vehicle at the maximum possible speed while implementing reliable position tracking at the maximum available tire forces.

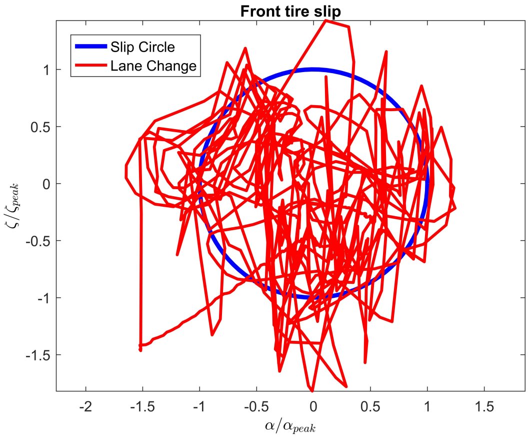

Figure 5.5: g-g diagram and tire slip

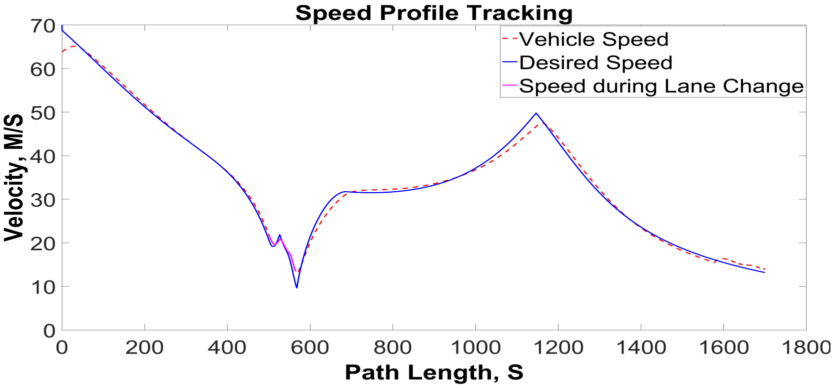

Figure 5.6: Velocity Tracking

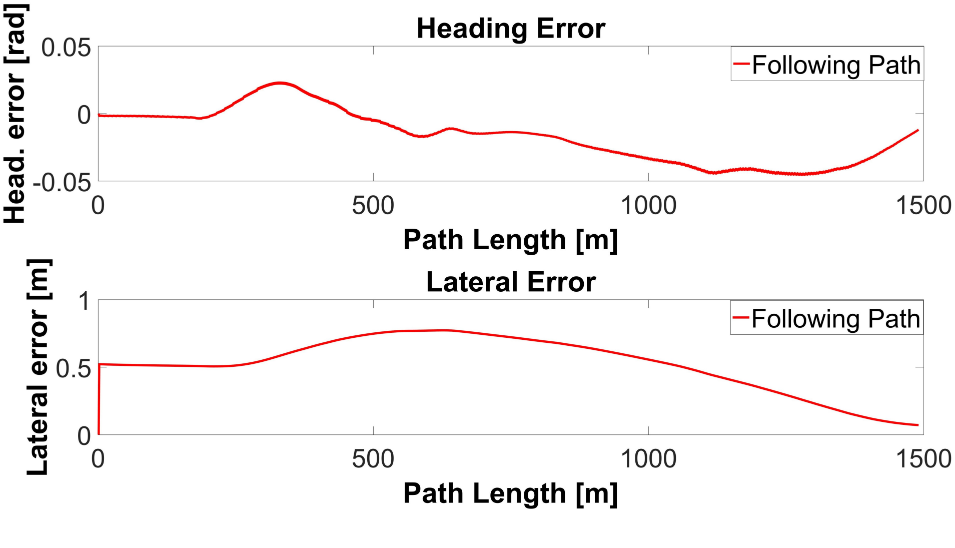

Figure 5.7: Tracking Error

Chapter 6

Conclusion

Autonomous driving is a rapidly growing field that has the potential to satisfy the long overdue transportation needs for people who can not drive themselves or prefer to be using their commuting time for business or entertainment. The possible significant influence on the financial sector and possibility of reduction of the carbon footprint due to introduction of autonomous delivery platforms, taxis, shuttle services or car share initiatives makes this research area so impactful. Not only can autonomous vehicles be programmed to drive more efficiency and utilized by multiple users, but they can also easily surpass the human drivers in safety performance. Unlike human drivers, autonomous vehicles do not stop paying attention to the road and the traffic conditions and do not fall asleep after long hours of driving, therefore it is always alert and is always making the best possible decision based on the information provided. The human drivers base their decision making while driving on their past experiences or intuition, whereas autonomous vehicles have the advantage of precisely estimating vehicle states by means of using analytical models and executing the required actuation command that tracks the desired vehicle pose, heading or velocity. Depending on the operation requirements, various analytical vehicle models can be used to implement the best candidates for the estimation or control tasks.

Multiple state estimation techniques exist that are either tailored for low speed driving, such as campus shuttle service for instance, or for urban and highway driving. Estimators that are designed for low speed driving utilize a vehicle model that is based on the geometry of the bicycle model and makes multiple assumptions, which reduces the needed number of states to track and as a result a number of sensors as well. Implementing estimators that take into account the vehicle kinematics as well as the dynamics, not only may require additional sensors, but also a complete vehicle modeling procedure to be performed in order to retrieve all the parameters that influence the dynamic behavior of the car.

This work offers an estimation method that was designed for low speed driving demo for CES 2017. The chosen kinematic and measurement models implemented in the estimator showed excellent results in reducing the oscillation and noise in the observed states. The measurements correlated well with the predicted states from the model and the estimator didn't diverge.

Similarly to the estimation design process, choosing the right controller depends on the task at hand and the requirements it introduces. While some of the control techniques based on the bicycle model kinematics require minimal computational resources and only need basic vehicle model parameters, their performance depends on the assumption of low speeds and no slip condition. The control techniques that utilize vehicle dynamics and/or optimization process, on the other hand, depend heavily on the vehicle model parameters that are not always easily accessible and may require a separate estimation and system identification procedure and additionally require more computational resources. Since the task was to implement a low speed demo sequence and very small amount of parameters were available, a steering controller based on the vehicle kinematics, a Stanley controller, was chosen and implemented. The steering controller performance was satisfactory and it satisfied all of the requirements of the demo sequence.

For further development of the autonomous driving platform for urban scenarios and highways in various adverse weather conditions, an aggressive driving controller framework was introduced in this work, that can be incorporated seamlessly into the autonomous driving software architecture. The aggressive driving controller was implemented with a path defined as a polynomial spiral curve. With the curvature profile provided by the local planner the aggressive driving controller generates maximum speed profile for a given friction coefficient and performs position and velocity tracking using bicycle model for vehicle dynamics and Dugoff model for the tire forces. The controller was tested using a high fidelity simulation software in a double lane change scenario to demonstrate a possible emergency maneuver execution during highway driving. The controller performed well in tracking position and desired path heading despite driving close to the edges of the slip circle. The longitudinal controller tracks the desired speed profile with a certain overshoot due to lookahead distance and use of proportional gain framework. Although this framework for the longitudinal controller delivers promising results while tracking the maximum speed profile, for autonomous driving purposes the additional speed profile smoothening step can be added that would remedy the performance of the controller in tracking maximum and minimum peaks on the speed profile. Additionally, an MPC framework can further be developed that can use the speed profile generation method introduced in this work, and can add cost to penalizing the slip circle feedback force in order to keep the vehicle away from the edges of the slip circle to ensure stability and safety. The use of lookahead distance in the longitudinal controller may cause overshoots in the velocity tracking, therefore, the lookahead distance can be chosen from an optimal lookup table that takes into account current speed, powertrain response model and required acceleration profile.

References

[1] David M Bevly, Jihan Ryu, and J Christian Gerdes. Integrating ins sensors with gps measurements for continuous estimation of vehicle sideslip, roll, and tire cornering stiffness. *IEEE Transactions on Intelligent Transportation Systems*, 7(4):483–493, 2006.

[2] F Borrelli, P Falcone, T Keviczky, J Asgari, and D Hrovat. Mpc-based approach to active steering for autonomous vehicle systems.

[3] Christopher R Carlson, J Christian Gerdes, and J David Powell. Practical position and yaw rate estimation with gps and differential wheelspeeds. In *Proceedings of AVEC 6th International Symposium*, 2002.

[4] Moustapha Doumiati, Alessandro Victorino, Ali Charara, and Daniel Lechner. Estimation of vehicle lateral tire-road forces: a comparison between extended and unscented kalman filtering. In *Control Conference (ECC), 2009 European*, pages 4804–4809. IEEE, 2009.

[5] Joseph Funke and J Christian Gerdes. Simple clothoid lane change trajectories for automated vehicles incorporating friction constraints. *Journal of Dynamic Systems, Measurement, and Control*, 138(2):021002, 2016.

[6] Thomas M Howard, Colin J Green, Alonzo Kelly, and Dave Ferguson. State space sampling of feasible motions for high-performance mobile robot navigation in complex environments. *Journal of Field Robotics*, 25(6-7):325–345, 2008.

[7] Yung-Hsiang Judy Hsu and J Christian Gerdes. The predictive nature of pneumatic trail: Tire slip angle and peak force estimation using steering torque. *Proc. AVEC08*, 8085, 2008.

[8] Yung-Hsiang Judy Hsu, Shad M Laws, and J Christian Gerdes. Estimation of tire slip angle and friction limits using steering torque.

[9] A. Kelly and B. Nagy. Reactive Nonholonomic Trajectory Generation via Parametric Optimal Control. In *The International Journal of Robotics Research*, volume 22, pages 583–601, 2003.

[10] Krisada Kritayakirana and J Christian Gerdes. Autonomous vehicle control at the limits of handling. *International Journal of Vehicle Autonomous Systems*, 10(4):271–296, 2012.

[11] Krisada Kritayakirana and J Christian Gerdes. Using the centre of percussion to design a steering controller for an autonomous race car. *Vehicle System Dynamics*, 50(sup1):33–51, 2012.

[12] S Laws, C Gadda, S Kohn, P Yih, JC Gerdes, and JC Milroy. Steer-by-wire suspension and steering design for controllability and observability. *IFAC Proceedings Volumes*, 38(1):204–209, 2005.

[13] Mikhail Pivtoraiko and Alonzo Kelly . Efficient constrained path planning via search in state lattices. In *The 8th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Munich, Germany, September 2005.

[14] R. Rajamani. *Vehicle Dynamics and Control*. Mechanical Engineering Series. Springer US, 2011.

[15] James Reeds and Lawrence Shepp. Optimal paths for a car that goes both forwards and backwards. *Pacific journal of mathematics*, 145(2):367–393, 1990.

[16] Eric J Rossetter, Joshua P Switkes, and J Christian Gerdes. Experimental validation of the potential field lanekeeping system. *International journal of automotive technology*, 5(2):95–108, 2004.

[17] A Scheuer and Th Fraichard. Planning continuous-curvature paths for car-like robots. 1996.

[18] Alexis Scheuer and Th Fraichard. Continuous-curvature path planning for car-like vehicles. In *Intelligent Robots and Systems, 1997. IROS'97., Proceedings of the 1997 IEEE/RSJ International Conference on*, volume 2, pages 997–1003. IEEE, 1997.

[19] Jarrod M Snider et al. Automatic steering methods for autonomous automobile path tracking. 2009.

[20] Sirui Song, Michael Chi Kam Chun, Jan Huissoon, and Steven L Waslander. Pneumatic trail based slip angle observer with dugoff tire model. In *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, pages 1127–1132. IEEE, 2014.

[21] Sirui Song, Andy Wong, Jan Huissoon, and Steven L Waslander. Autonomous vehicle control near the limit of friction. In *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*, pages 1008–1013. IEEE, 2015.

[22] Sebastian Thrun, Mike Montemerlo, Hendrik Dahlkamp, David Stavens, Andrei Aron, James Diebel, Philip Fong, John Gale, Morgan Halpenny, Gabriel Hoffmann, et al.

Stanley: The robot that won the darpa grand challenge. *Journal of field Robotics*, 23(9):661–692, 2006.

[23] Wenda Xu, Junqing Wei, John M. Dolan, Huijing Zhao, and Hongbin Zha. A real-time motion planner with trajectory optimization for autonomous vehicles. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 2061–2067, 2012.