POLITECNICO DI MILANO

School of Industrial and Information Engineering

M.Sc. in Automation and Control Engineering

A study of an LQR drifting controller for RC cars

Supervisor: Prof. Luca BASCETTA
Co-supervisor: Dott. Ing. Marco BAUR

Candidate:
SUN Yao ID: 852893

Academic year 2017-2018

Abstract

Drifting is a well-known driving technique especially in competitive sports and demonstration events. Drifting control is one of the hottest topics in vehicle control area. The present study intends to analyze the drifting condition for a car and design a proper controller to stabilize a Rear Wheel Driven (RWD), Radio Controlled (RC) car to a desired drifting equilibrium. The Linear Quadratic Regulator (LQR) is chosen in the present study. The design of the controller is based on a single track model with three state variables (longitudinal velocity, side slip angle and yaw rate) and two input commands (steer angle and throttle value). Those three state variables describe the dynamics of car precisely and all of them can be obtained by using an inertial measurement unit (IMU) and a motion tracking system. The steer and throttle commands are the most straightforward control variables for a driver. A feedforward-feedback control scheme is also applied to deal with the non-linear problem of single track model in this study. Simulink is a developed environment for designing complex system. The whole system is designed in Simulink and is compiled to ROS nodes with the function of MatLab directly. When the ROS nodes are generated, a series of tests with simple input commands are performed for validation of the procedure from Simulink model to ROS nodes. The LQR controller with a feedforward-feedback scheme are finally simulated in ROS. The simulation results are evaluated. It indicates that the controller can properly stabilize the model to a desired equilibrium from a standing still state with optimized parameters of LQR.

Keywords: Drifting, Linear Quadratic Regulator, RC car, ROS

Sommario

Il drifting è una tecnica di guida nota specialmente nella competizioni sportive e nello spettacolo. Il Controllo del drifting è uno degli argomenti più popolari nell'area di controllo del veicolo. Lo studio presente intende analizzare la condizione di drfiting per una macchina radio comandata (RC) e progettare un controllore per la stabilizzazione di un'auto a trazione posteriore (RWD) attorno ad un equilibrio di drifting desiderato. Il regolatore lineare quadratico (LQR) è stato scelto nel presente studio. Il controllore si basa su un modello bicicletta con tre variabili di stato (la velocità longitudinale, l'angolo di slittamento laterale e la velocità di imbardata) e due comandi d'ingresso (l'angolo dello sterzo e il valore dell'acceleratore). Queste tre variabili di stato possono descrivere la dinamica di automobile con precisione ed esse si possono stimare a partire da una piattaforma inerziale (IMU) e da un sistema di motion tracking. I comandi dello sterzo e dell'acceleratore sono le variabili di controllo più diretti per un pilota. In questo studio, viene applicato anche uno schema feedforward-feedback di controllo per affrontare il problema non lineare del modello bicicletta. Simulink è un ambiente sviluppato per progettare i sistemi complessi. Il sistema intero si progetta in Simulink e si compila ai nodi di ROS direttamente tramite la funzione di MatLab. Quando vengono generati i nodi di ROS, una serie di prova con i comandi d'ingresso semplici sono state eseguite per la convalida della procedura dal modello di Simulink ai nodi di ROS. A conclusione di questo lavoro, il controllore LQR con una schema feedforward-feedback è simulato in ROS. I risultati della simulazione indicano che il controllore può stabilizzare il modello all'equilibrio desiderato, partendo dalla condizione di veicolo fermo, in presenza dei parametri ottimizzati dello LQR.

Parole chiave: Drifting, Regolatore Lineare Quadratico, Macchina RC, ROS

Contents

List of Figures

List of Tables

Introduction

Drifting is a driving technique where the vehicle is intentionally driven to cause loss of traction in the wheels. In the meanwhile, the driver keeps the vehicle under control through the whole cornering process. It is widely used in car race to maintain high velocity during turning. A vehicle under drifting suffers over-steering and has a large side slip angle which is not easy to control. The purpose of this study is to set up a drifting control system which can automatically drive the vehicle to a stabilized drifting condition safely.

In the present study, a single track model with rear wheel driven (RWD) is firstly designed as the study object. The equilibria of the model are analyzed and a general drifting equilibrium is chosen as the target condition. In the model, the steer angle and throttle value are inputs, two variables drivers can manipulate directly. The longitudinal velocity, side slip angle and yaw rate are state variables for the dynamic system, which can be obtained by an inertial measurement unit (IMU) and a motion tracking system.

In order to stabilize the model to the target drifting condition, a linear quadratic regulator (LQR) control strategy is used. The LQR control strategy is easy to manipulate and requires less analyses of the dynamics during the stabilization process compared with traditional pole placement method[1, 2]. It only needs to find proper parameters in a quadratic cost function to maintain the system working with the minimum cost. A feedforward-feedback control scheme is established for the LQR controller. It intends to work with the non-linear single track model which the LQR cannot deal with alone.

The whole system is completed in Simulink with the help of MatLab toolbox. Then all the parts of the system are compiled and generated directly to Robotics Operation System (ROS) nodes. ROS is a framework for robotic software development[3, 4]. It provides services which are widely used in a robotics system like hardware abstraction, low-level device control, and message-passing. Compiling the system in ROS gives a platform to test the controller in real car situation that, however, is not addressed in this study. The generated nodes are validated by a series of tests since the ROS nodes are composed of C++ code, which should be used in simulation after validation.

In the end, the controller is simulated in ROS framework. The controller gives positive simulation results with an initial state of standing still. The yaw rate is increasing for an instant and converges to the equilibrium value within 3s. The side slip angle is stabilized to its equilibrium around 4s after a small increasing. After that, the longitudinal velocity reaches its steady state.

Outline:

The structure of this document is here.

- Chapter one firstly introduces drifting and drifting control with some previously reported practical methods. Then, a list of all the hardware tools which are involved in this study is given.
- Chapter two mainly discusses the dynamics of vehicle model and the drifting equilibrium in details, including completing the single track model and analyzing the stability and bifurcation of drifting equilibrium.
- Chapter three mainly talks about how to design our LQR controller and gives simulation results in Simulink.
- Chapter four shows the compiling of the system from Simulink to ROS, validations of ROS nodes, and the evaluation of simulation results.

Chapter 1 Background and hardware

1.1 Drifting

Drifting is a driving technique and widely used in car race to maintain high velocity during turning. Normally it is manipulated by the drivers manually. Drifting competitions are held worldwide to test the ability of a driver in achieving drifting condition and keeping the car under that condition during a period of time.

The drifting condition mainly happens with the saturation of rear tire during turning[1, 5]. The saturated rear tire cannot afford more lateral force and cause an over-steering of the car. It leads the car to spin and causes a large side slip angle. If the driver cannot keep the car's body properly through counter-steering, the car will be out of control and cause danger. However, as long as the car is properly controlled, the drifting will keep the car at a high speed during turning. Thus, drifting is a widely accepted driving technique using in car race.

One of the important features of drifting condition is the high side slip angle of the car. The side slip angle β is defined as the ratio of the lateral velocity over the longitudinal velocity of the car. When this value is small, the car is under a normal cornering condition; when the side slip angle is large enough, the car is in the drifting condition. In the following discussion, both the high side slip angle and the saturation of rear tire are checked in order to distinguish the drifting from the normal cornering.

There are two main categories of drifting. One of them is transient drift and the other one is sustained drift. Just as the names imply, these two conditions have different time period of drifting. Transient drift requires a short time of drifting condition and is usually applied in an emergency condition such as avoiding collision. For example, a time-optimal control is designed[6, 7] for this condition in order to mitigate an unavoidable T-bones collision with an aggressive maneuver. Transient drifting is also considered to be a solution for problems of minimum time and maximum exit velocity turning[8].

Sustained drift requires the car keeping under the drifting condition for a long time. The car has not only to be driven to a drifting equilibrium but also to be maintained under this condition for a relatively long time. A skilled driver has an instinct of adjusting all the variables in a perfect range so as to make the drifting sustained and safe. However, it requires a plenty of training and even some talents. The aim of the study is to design an autonomous controller in order to stabilize a Radio Controlled (RC) car to a sustained drifting condition.

Figure 1. 1. Drifting in race

1.2 Drifting Control

In practice there are several considerations for drifting control in terms of saturation rear tire. As the total force applied on the tire should not exceed the available friction force, the car will easily reach the saturation condition when available friction force on the tire is reduced. Since the friction force is affected by the friction coefficient and the normal force, they are the main objects to consider.

1.2.1 Friction coefficient

Friction coefficient is affected by road condition and the longitudinal slip[9]. Road condition will not be discussed in this study, so the main point is the longitudinal slip λ.

$$\lambda = \frac{v-\omega r}{max\{v,\omega r\}} \tag{1-1}$$

In the equation (1-1), v is the velocity of vehicle body, ω is the angular wheel speed and r is the rolling radius. The relationships between the longitudinal slip and longitudinal friction coefficient, the longitudinal slip and lateral friction coefficient are shown in figure 1.2 and figure 1.3.

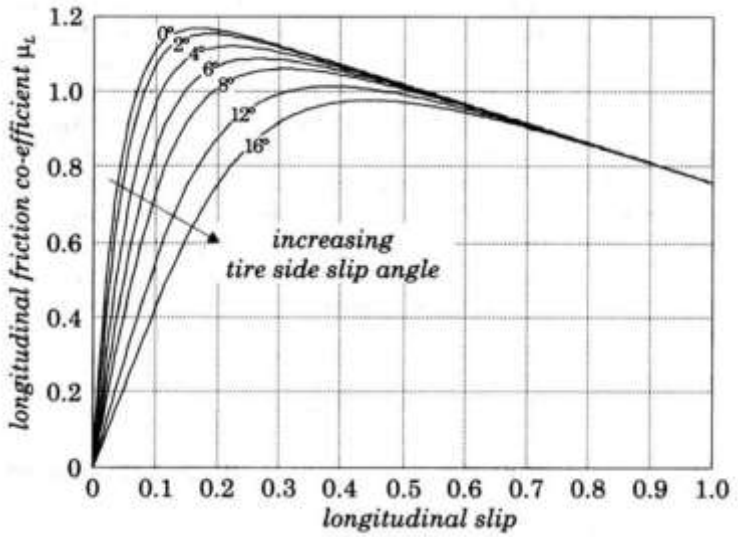

Figure 1. 2. Relation between the longitudinal slip and the longitudinal friction coefficient

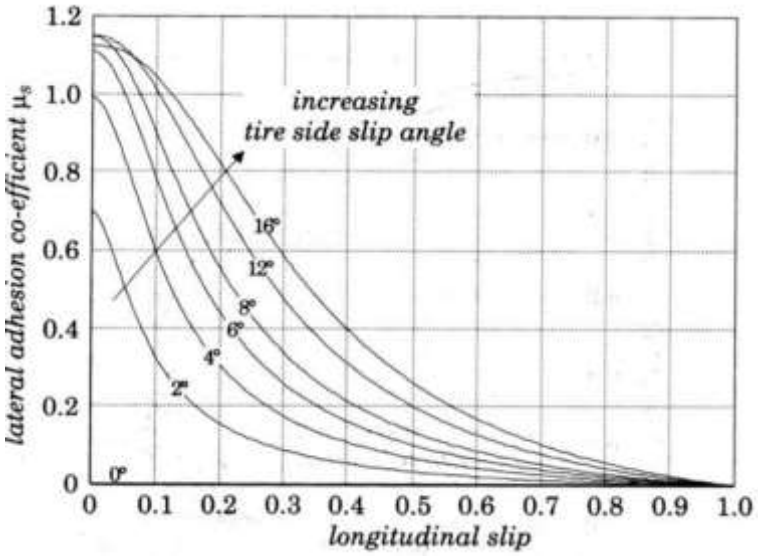

Figure 1. 3. Relation between the longitudinal slip and the lateral friction coefficient

When the car is going straightforward, there is no slip angle on the rear tire and the lateral friction force is null. Based on figure 1.2, there are two ways to decrease the longitudinal friction force: brake the rear wheels, or increase the throttle.

The first way is achieved by giving a hand brake. When the rear wheels are locked, the steer angle leads to a large spinning range of the car's body by over-steering. However, this drifting loses speed during turning. When the maneuver finished, drivers need to accelerate again. Since it violates the main target of keeping a large velocity during turning, this way is not used in most of the time.

Instead of braking, the second way is to increase throttle before turning. The large power

from engine forces the tire to lose the traction which leads to drifting condition by applying a steer angle. As soon as finishing turning, the throttle is loosed so that the grip on rear tire can recover.

1.2.2 Normal force

Apart from the friction coefficient, changing the normal force on the rear part is also a way to achieve the saturation of rear tire. This method is also widely used in professional race. It uses the weight transfer due to the brake.

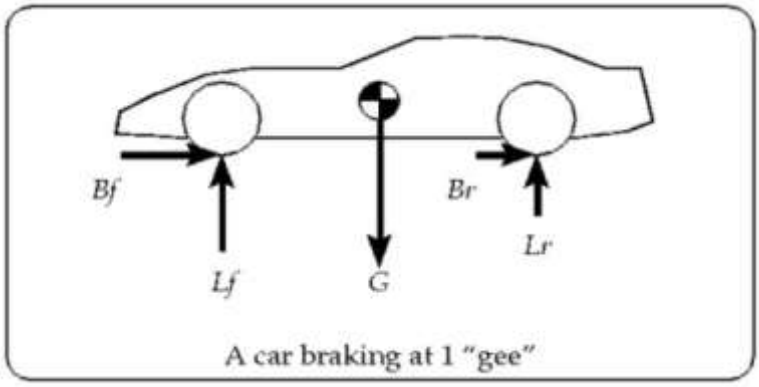

Figure 1. 4. Example of weight transfer[10]

When the car is standing still or going at a constant speed, the front load force Lf and rear load force Lr should balance each other to prevent the car rotating around its center of gravity. Under this condition, the braking forces Bf and Br are zeros. If the car starts braking, those two braking forces should create a torque which make the car rotate around its COG counterclockwise. In order to prevent this trend, the front load force should increase its value to balance the counterclockwise torque. Since the addition of front load force and rear load force is always equal to the weight of the car, the rear load force should be less than its previous value during the braking condition. The decreasing of rear load force will reduce the friction force of the tire. So the tire will goes into saturation part as expected.

1.2.3 Strategies for achieving an autonomous drifting control

Several methods designed for achieving an autonomous control of drifting have been reported previously. The one reported by Hindiyeh et al is based on a three-state model and regards the steer angle and longitudinal force as two inputs[1]. Instead of directly stabilizing the car at a desired equilibrium, it focuses on the analyses of dynamics among all the state variables during the stabilization. As for the controller, it uses two loops control scheme to stabilize the side slip angle in an inner loop with respect to a desired yaw rate which comes from an outer one. This strategy requires a physical intuition to design the controller. The method reported by

Voser et al is simpler. This method decouples the whole controller into two parts: the longitudinal control and the steer control, respectively[2]. However, this division loses fidelity in the behavior of rear tire forces and it requires further analyses to guarantee the stability. These two methods are all designed based on the traditional pole placement method.

Apart from pole placement based control scheme, applying LQR is another strategy [11-14]. The method reported in 2011 tries to use the slip ratio to stabilize an Rear Wheel Driven (RWD) car[11]. It applies a sliding model to regulate the rear wheel speed to the steady-state condition by the wheel torque. As for stabilizing a FWD car in a drifting equilibrium, Velenis et al derives from a handbrake-cornering maneuver with locked rear wheels[12]. These methods are considered in a four wheels model. To simplify the model, Velenis et al stabilizes a single track model with independent front and rear wheels for drifting equilibrium[13, 14]. However the wheel torque is not a direct variable for drivers. The most direct inputs of an RWD car during driving are the steer angle of front wheel and the throttle value acted on the rear wheel.

1.3 Hardware

The whole system is abstracted from an RC car, a 1:10 scale vehicle. All the design in the following chapters are based on the characteristics of that car. It guarantees the fidelity of this study and gives convenience to apply the controller to a real car in the later research.

1.3.1 Odroid-XU4

Odroid-XU4 is the main core of the system. It is a powerful computing device based on the ARM architecture, which is widely used for embedded 32-bit computing. In addition, it gives an open source platform for developing a large variety type of operating system, like Linux, Android, Debian and Fedora. Several open source packages are available in Odroid-XU4, and can be chosen for any kinds of design. In this study, Ubuntu 16.04 LTS is selected as the operating system (OS) which is compatible with the Robotic Operating System (ROS) perfectly.

Figure 1. 5. Odroid-XU4

Here are some technical specifications being listed here[15]:

Processor: Samsung Exynos5422 ARM Cortex™-A15 2GHz and ARM Cortex™-A7 1.4GHz with Mali Mali-T628 MP6 GPU

Storage: eMMC 5.0 module socket (8GB/64GB) and MicroSD card slot

Memory: 2Gbyte LPDDR3 RAM PoP (750Mhz, 12GB/s memory bandwidth, 2x32bit bus)

USB ports: 2 USB 3.0 ports and 1 USB 2.0 ports

Ethernet: RJ-45 Ethernet port 10/100/1000 Mbps speed

Power supply: 5V/4A DC input

WiFi: IEEE 802.11 ac/b/g/n WLAN module with dual band (2.4GHz and 5GHz) USB support

Size: 83 x 58 x 20 mm

1.3.2 Arduino UNO

Arduino acts as the micro-controller board which based on the 16MHz ATmega328 in the study. It is used to receive, manipulate and transfer signals among the Odroid-XU4, radio controller, sensors and actuators. Arduino is also an open source platform, and it is very convenient to be programmed.

Figure 1. 6. Arduino UNO

Here are some technical specifications[16]:

Micro-controller: ATmega328 with 16MHZ clock

Operating voltage: 5V

DC input voltage: 7V-12V

Digital I/O pins: 14 (6 provide PWM output)

Analog input pins: 6

Memory: 32KB Flash, 2KB SRAM, 1KB EEPROM

1.3.3 IMU myAHRS+

An inertia measurement unit (IMU), is an electronic sensor device and is used to measure the accelerations, angular speed and magnetic field around a certain object. An IMU is usually composed of three accelerators and three gyroscopes to measure accelerations and angular rate along different axis. In some cases, it also contains a magnetometer to measure the magnetic field in the environment. In the present study, an attitude and heading reference system (AHRS) of the IMU is used to estimate the roll-pitch-yaw angles which intend to describe an accurate orientation of the RC car.

Figure 1. 7. myAHRS+

Here are some technical specifications[17]:

Sensor: 3-Axis Gyroscope, 3-Axis Accelerometer, 3-Axis Geomagnetic Sensor

Output Data: Euler angle, Quaternion

Sensor Data: Acceleration, Angular Velocity, Geomagnetic Orientation

Output Rate: Max 100 Hz

Communication Interface: USB with Virtual COM PORT, UART with maximum 460,800 bps, I2C

Dimension: 21 x 27mm

1.3.4 Optitrack Motive

Optitrack Motive is the motion tracking system used in the present study. Optitrack offers high-performance optical tracking which can be used in the scholarship and industrial domain, like the movement sciences, virtual reality, animation and robotics[18].

Motive is a unified software platform of Optitrack. It has two main parts, the tracker and the body. The tracker tracks an object with 6DoF in real time and offline workflows. After setting and managing all the cameras and frames, it provides precision results of the target during the whole motion. As to the body, it mainly deals with the action of a certain skeleton, which is usually used in the animation and film. It will not be discussed in detail here.

Figure 1. 8. Camera of Optitrack

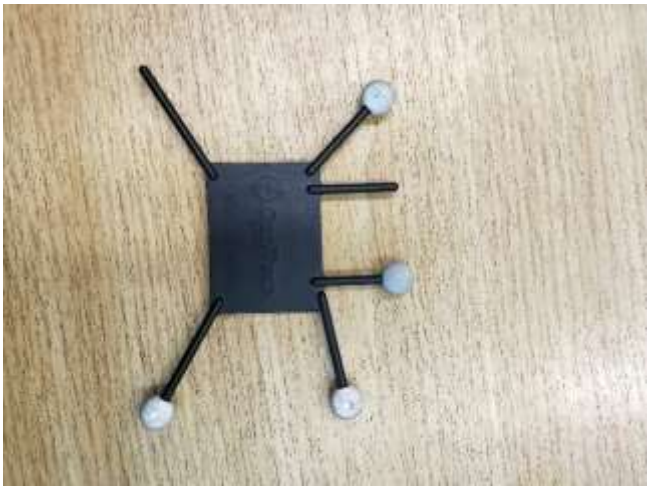

Figure 1. 9. Marker set

In Motive, the car is labelled with a rigid body marker set. The marker set is a black plastic rigid body with a hook Velcro on the car. It has 6 attachment points on which the markers are placed so as to track the body of the RC car.

1.3.5 RC cars

An RC car which has a 1:10 scale with rear-wheel drive is the object in the present study. The model type of car is Sakura D4. The large range of steer angles and rear-wheel drive make it much easier to drift.

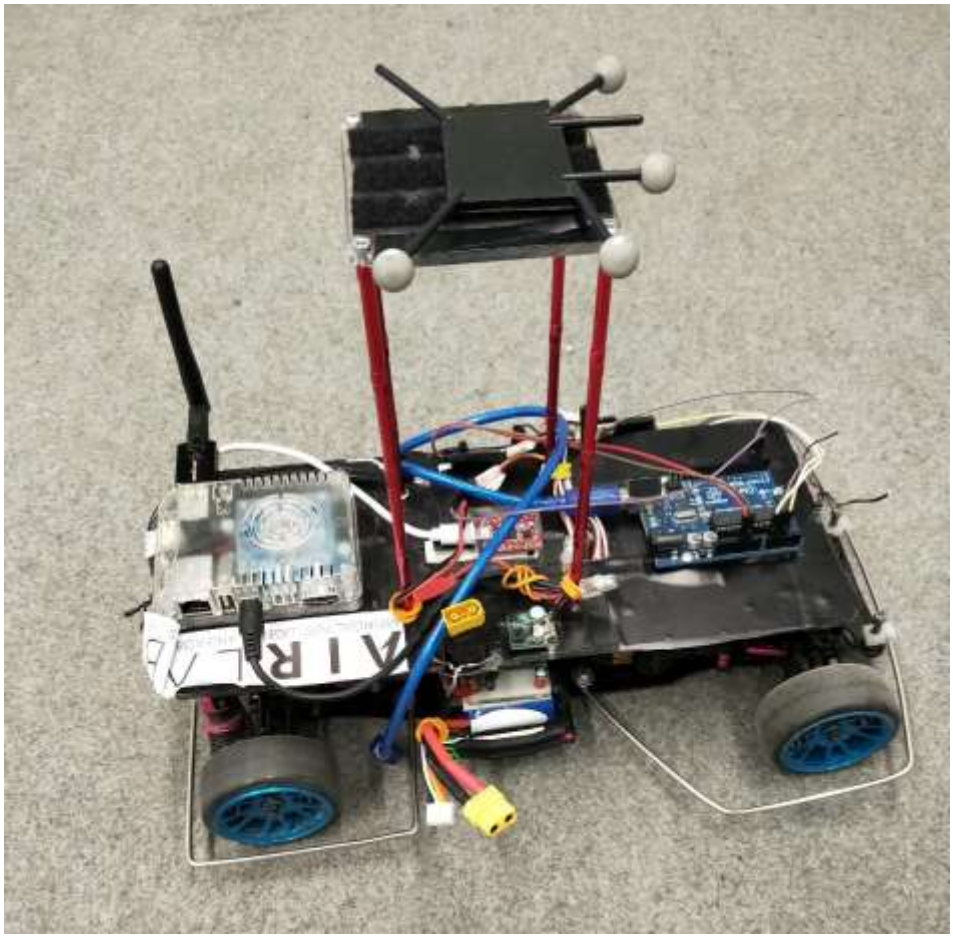

Figure 1. 10. An RC car

Here are some technical specifications of the main chassis[19]:

Length: 363mm

Track width: 180mm for front and 178mm for rear

Wheelbase: 255mm

Weight: 735g

Internal gear ratio: 2.06:1

1.3.6 Radio controller

Flysky 3CH 2.4GHz radio controller is used in the present study. As its name suggests, it has 3 signal channels. The first and the second channel are used for the steer and throttle command, respectively. The last channel is used for a command to switch the control mode from manual to automatic. The receiver of radio commands is on the body of car. It controls actuators of the car both by the receiving commands in manual mode and the computed commands from Ordoid-XU4 in automatic mode.

Figure 1. 11. Radio controller

1.3.7 Motor servo

Cheetah 1/10th 60A Sensored Combo set is the motor servo in the car. It is composed of two parts, the sensored combo motor and the sensored electronic speed control (ESC). The motor is brushless and has 5 kinds of K_v rating available for adjustment of a proper velocity and torque. The ESC is designed to control the speed of the motor by regulation of maximum current. Since it is charged by LiPo battery. It also functions as a transfer of the voltage from DC to three-phase for driving the motor.

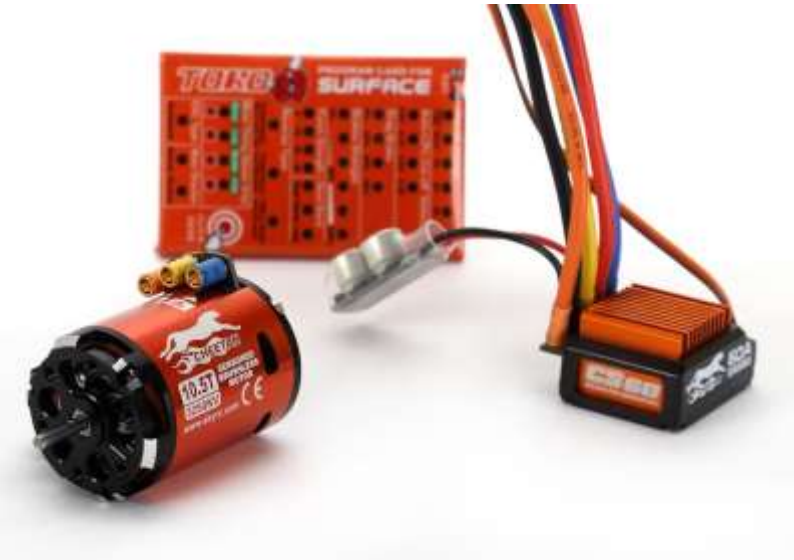

Figure 1. 12. Motor and ESC

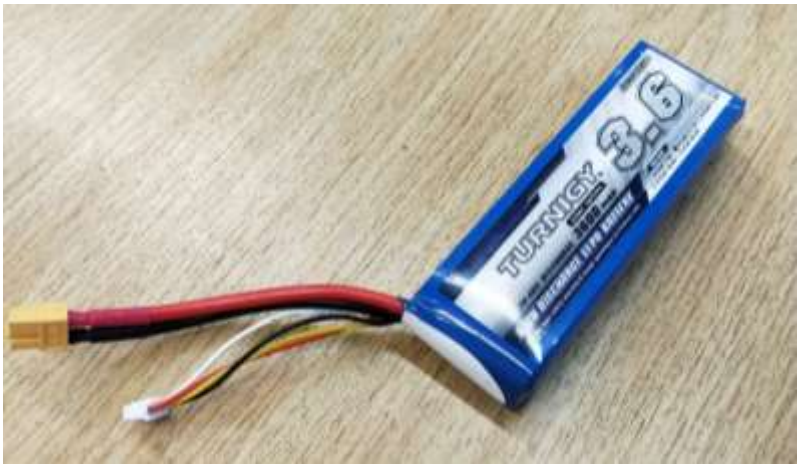

Figure 1. 13. LiPo Battery

Chapter 2 Vehicle model and drifting equilibrium

A single track model is established to describe the car during the whole motion process in the present study. The model focuses on two main parts of the motion, the kinematic part and the dynamic part. In kinematic part, the important variables considered are the longitudinal velocity, lateral velocity and yaw rate. As to the dynamic part, the dynamic features of variables above are influenced by forces on both wheels and torques on the front and rear body of a car. The forces and torques are found by tire characteristics. After completing the vehicle model, a drifting equilibrium is chosen as the target equilibrium for the design of control scheme.

2.1 Dynamic state equations

Two accelerations along longitudinal and lateral axes and the acceleration of yaw rate are obtained by the forces acted on both wheels as well as the torques on the front and rear wheel axes.

$$ma_x = \Sigma F_x \tag{2-1}$$

$$ma_y = \Sigma F_y \tag{2-2}$$

$$J\dot{r} = l_f F_f - l_r F_r \tag{2-3}$$

Where:

- m is the mass of car
- a_x and a_y are the accelerations on the longitudinal and lateral orientations, respectively
- F_x and F_y are the forces on the longitudinal and lateral orientations
- J is the inertia moment
- r is the yaw rate during turning
- l_f and l_r are the lengths from COG to front and rear axes
- F_f and F_r are the lateral forces on the front and rear axes

From equations (2-1) to (2-3), three points have to be considered for the dynamics of a car: the acceleration along the longitudinal orientation, the acceleration along the lateral orientation, and the yaw motion along the z-axis.

Apart from the parameters of the car itself, the accelerations and yaw rate are found by

kinematic part. The forces of longitudinal and lateral orientations are described by tire model.

2.2 Accelerations along longitudinal and lateral orientations

In a turning condition, the velocity of car changes both its magnitude and orientation. The orientation is described by side slip angle β. The amount of side slip angle represents how much the velocity is away from the longitudinal orientation of car.

This angle is associated with the yaw rate, steer angle and lateral acceleration of the car. Thus, it is an important factor that must be taken into consideration in the present study. In most of the cases, this variable is estimated by some indirect methods [20-22] but not by measuring due to requirements of high performance and expensive optical sensors. In this study, it is estimated with the help of a motion tracking system, Optitrack Motive.

The definition of the side slip angle is showed by the following formula:

$$\tan\beta = \frac{V_y}{V_x} \tag{2-4}$$

In formula (2-4), V_y and V_x do not depend on the world reference frame. V_x is the longitudinal velocity aligned with the orientation of the car's body. V_y is the lateral velocity which is vertical to the longitudinal part. In the following discussion, the same rule is always used to describe the meaning of x and y.

When the car is turning, it will rotate around its z-axis and lead to a yaw rate. So, when considering the accelerations of the car, the rotating part has to be taken into account.

$$V = V_x i + V_y j \tag{2-5}$$

$$\dot{V} = \left(\dot{V}_x - rV_y\right)i + (\dot{V}_y + rV_x)j \tag{2-6}$$

Where i and j are the unit vector of longitudinal and lateral orientations, r is the yaw rate of the car when it is turning.

In the equation (2-5) and (2-6), the accelerations on longitudinal and lateral orientations are not only the linear acceleration of velocity but also have a part with respect to the yaw rate and the other velocity. Since that, the dynamic equations are revised to the following form:

$$ma_x = \Sigma F_x + mrV_y \tag{2-7}$$

$$ma_y = \Sigma F_y - mrV_x \tag{2-8}$$

2.3 Single track model

Before finding out the resultant forces on two orientations for the equations above, it is necessary to apply a proper vehicle model to abstract a real car. The most used model in vehicle is the single track model or bicycle model, which is shown in figure 2.1. In figure 2.1, the complex body is simplified by a single axis and two lumped wheels. The orientation of car body is aligned with the axis.

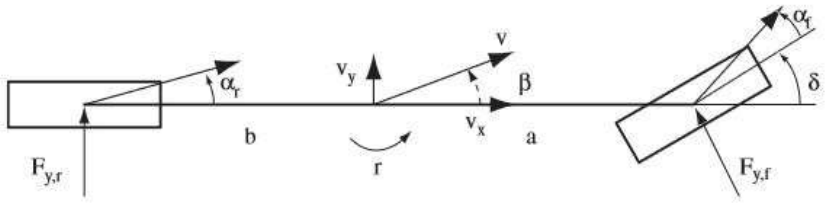

Figure 2. 1. Single track model

As mentioned above, the velocity has an angle β from the center axis called side slip angle. It is divided into two parts. The longitudinal velocity V_x is aligned with the rear tire because in an RWD car the front tire is the only steer one. The lateral velocity V_y is vertical to the center axis which shows how much the car goes to the orientation vertical to the center axis.

When the velocities of front and rear wheels are considered individually, there exists two extra angles between the individual velocity and the wheel. The front and rear wheel velocities must be set on both sides of the resultant one. These extra angles are shown as the angle from the wheel's orientation to their individual velocity, which are α_f and α_r.

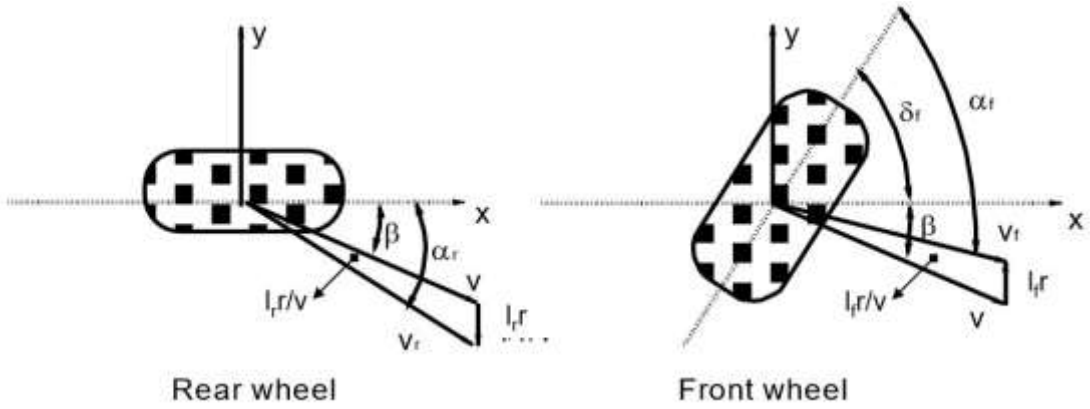

Figure 2. 2. Wheels in detail

Those two angles are called tire drift angler because they show how much the velocity goes aside from the wheel's orientation. They are very important in the further study since they are associated with the forces on the tire.

$$\alpha_r = \beta + \frac{l_r r}{V} \tag{2.9}$$

$$\alpha_f = \beta + \delta - \frac{l_f r}{V} \tag{2-10}$$

Please be careful that equation (2-9) and (2-10) can only be used under the particular condition in figure 2.2, where the side slip angle has opposite sign to the steer angle. If the side slip angle has the same sign with the steer angle, the equations must be changed. To avoid making any ambiguity in the following description, anti-clockwise is considered as the positive orientation for angles. Then, two equations should be changed to the following form.

$$\alpha_r = \beta - \frac{l_r r}{V} \tag{2-11}$$

$$\alpha_f = \beta - \delta + \frac{l_f r}{V} \tag{2-12}$$

If doing an approximation with the resultant velocity and longitudinal velocity, the equation (2-11) and (2-12) are rewritten as the following form:

$$\alpha_r = \beta - \frac{l_r r}{V_x} \tag{2-13}$$

$$\alpha_f = \beta + \frac{l_f r}{V_x} - \delta \tag{2-14}$$

Considering about the forces on the wheels, the front wheel of an RWD car is the only one involving in the steering condition, and the driving force from engine is only acted on the rear wheel. Furthermore, this decouples two main actions of the car. The driving action is only on rear wheel and steering action is only on the front wheel. This will simplify the further analyses much.

Since the front wheel has only action of steering, there is only lateral force on the tire. Please be aware that this 'lateral' is based on the wheel itself instead of car's body. So, the orientation of that lateral force is respect to the steer angle δ. As to the rear wheel, since it will never change its orientation away from the center axis, the longitudinal and lateral force will be aligned with the car's body.

Back to the dynamic equations, since the lateral velocity has a relationship with the longitudinal one and the side slip angle, the variable V_y is transferred to side slip angle. Therefore, three state variables in this model become the longitudinal velocity V_x, the yaw rate

r and the side slip angle β. These three variables describe the total dynamics of a car and they are implemented by a control scheme to their desired values.

Due to the single track model which has been analyzed above, three dynamic equations are revised in the following form.

$$\dot{V}_x = \frac{1}{m}\left(F_{xr} - F_{yf}\sin\delta\right) + rV_x\tan\beta \tag{2-15}$$

$$\dot{r} = \frac{1}{J_z}\left(aF_{yf}\cos\delta - bF_{yr}\right) \tag{2-16}$$

$$\dot{\beta} = \frac{1}{mV_x}\left(F_{yf}\cos\delta + F_{yr}\right) - r \tag{2-17}$$

From equation (2-15) to (2-17), m is the mass of car and J_z is the momentum of inertia around z-axis. Besides, a and b are the lengths from front and rear wheels to the center of gravity (COG), respectively. Those variables are all the physical parameters which can be measured or obtained easily. Apart from them, the three forces (F_{yf}, F_{yr} and F_{xr}) are described by a tire model.

2.4 Magic formula

Magic formula[23, 24], or Pacejka magic formula tire model is a universally model to study the characteristic of tire force and moment. It is shown by four dimensionless coefficients B, C, D and E for the stiffness factor, shape factor, peak value and curvature factor, respectively. In a more general form, the formula also involves two offset coefficients S_v and S_h. They represent the shift in vertical and horizontal orientation respectively.

$$F_{x0} = D_x\sin[C_x\arctan\{B_x\lambda_x - E_x(B_x\lambda_x - \arctan(B_x\lambda_x))\}] + S_{vx} \tag{2-18}$$

The formula (2-18) shows the longitudinal force of tire for pure slip. The subscript x refers to the condition along the x axis. The factor λ is the slip ratio here and is also the input for this formula. Similarly, the formula of lateral force is shown below:

$$F_{y0} = D_y\sin\left[C_y\arctan\left\{B_y\alpha_y - E_x\left(B_y\alpha_y - \arctan\left(B_y\alpha_y\right)\right)\right\}\right] + S_{vy} \tag{2-19}$$

The subscript y means the condition along the y axis. The difference from equation (2-18) is the factor α, which shows the slip angle of tire.

The reason why the Pacejka model is called as ‘Magic formula’ is because it fits most of the conditions without a certain physical basis. When all the coefficients are estimated beforehand, it is easy to manipulate and accurate in calculation. Thus, this is widely used in a professional condition such as high skilled race. However, it requires a large set of complicated

coefficients estimations. It is not used in the present study.

2.5 Fiala tire model

In this study Fiala model[2] is chosen as the tire model. In Fiala model, the forces are considered as a function of slip angles α_f and α_r, and are shown in the following formula.

$$F_{yf} = \begin{cases} -\mu F_{zf} \sin\alpha_f & |\alpha_f| \geq \tan^{-1}\frac{3\mu F_{zf}}{C_f} \\ -C_f \tan\alpha_f + \frac{C_f^2}{3\mu F_{zf}}|\tan\alpha_f|\tan\alpha_f - \frac{C_f^3}{27\mu^2 F_{zf}^2}\tan\alpha_f{}^3 & |\alpha_f| < \tan^{-1}\frac{3\mu F_{zf}}{C_f} \end{cases} \quad (2\text{-}20)$$

$$F_{yr} = \begin{cases} -\mu F_{zr} \sin\alpha_r f & |\alpha_r| \geq \tan^{-1}\frac{3\mu F_{zr} f}{C_f} \\ -C_r \tan\alpha_r + \frac{C_r^2}{3\mu F_{zr} f}|\tan\alpha_r|\tan\alpha_r - \frac{C_r^3}{27\mu^2 F_{zr}^2 f^2}\tan\alpha_r{}^3 & |\alpha_r| < \tan^{-1}\frac{3\mu F_{zr} f}{C_r} \end{cases} \quad (2\text{-}21)$$

Where f is a factor which can be described as:

$$f = \begin{cases} 0 & \sqrt{(\mu F_{zr})^2 - F_{xr}^2} \leq 0 \\ \sqrt{\frac{(\mu F_{zr})^2 - F_{xr}^2}{(\mu F_{zr})^2}} & \sqrt{(\mu F_{zr})^2 - F_{xr}^2} > 0 \end{cases} \quad (2\text{-}22)$$

As shown above, the lateral forces on both front and rear tires are computed directly. The longitudinal force on the front tire is zero because there is no action on it. Then, the longitudinal force on the rear tire is considered as an input variable of the car because it can be determined by the throttle. By the way, the other input which can be determined is the steer angle δ.

As to the parameters, F_z is the normal load; μ is the friction coefficient; α are the slip angles; C_f and C_r are the cornering stiffness of front and rear tires, respectively. When the Fiala model is introduced into the dynamic equations, the unknown variables are three state variables and two inputs, the steer angle δ and the longitudinal rear fore F_{xr}.

Please be aware, all forces acted on tires are from the friction force. They have a maximum limit with respect to the friction circle. It means the resultant force of lateral force and longitudinal force should never exceed the friction of tire. This is one of the key constrains in the further analysis.

2.6 Equilibrium points

Since the car is controlled at an equilibrium states, all the dynamic equations are set to be zero to obtain steady-state values. The longitudinal velocity V_x is set to 1.5m/s as the equilibrium value in this study.

$$0 = \frac{1}{m}(F_{xr}^{eq} - F_{yf}^{eq}\sin\delta^{eq}) + r^{eq}V_x\tan\beta^{eq} \quad (2\text{-}23)$$

$$0 = \frac{1}{J_z}(aF_{yf}^{eq}\cos\delta^{eq} - bF_{yr}^{eq}) \quad (2\text{-}24)$$

$$0 = \frac{1}{mV_x}\left(F_{yf}^{eq}\cos\delta^{eq} + F_{yr}^{eq}\right) - r^{eq} \quad (2\text{-}25)$$

Where $\boldsymbol{F_{xr}^{eq}}$, $\boldsymbol{F_{yr}^{eq}}$, $\boldsymbol{F_{yf}^{eq}}$, $\boldsymbol{\delta^{eq}}$, $\boldsymbol{r^{eq}}$ and $\boldsymbol{\beta^{eq}}$ are the equilibrium variables of longitudinal force on rear tire, lateral force on rear tire, lateral force on front tire, steer angle, yaw rate and side slip angle, respectively. V_x is a determined value. F_{yf} and F_{yr} are obtained by the Fiala model. Particularly, those two forces are functions of side slip angle, steer angle, yaw rate and velocity. If given a certain value of δ, there are three equations for three unknowns. So numerical results of equilibrium points are computed directly.

In drifting condition, there is a high side slip angle. Otherwise under a small side slip angle, there should be normal cornering. So all side slip angles are divided into three parts along the whole span, which are two large angle parts with opposite signs and one part around zero. Then, with a certain value of δ, other three variables are obtained with respect to the three parts of side slip angle β.

As for the steer angle, it is not to get only one equilibrium point but all possible equilibrium points in the present study. So, a span of δ from -20deg to 20deg is chosen. As the longitudinal velocity V_x is set to 1.5m/s, all the equilibrium points along the span of steer angle are shown below.

From figure 2.3 to figure 2.8, three different types of equilibrium are shown in different colors. Two drifting conditions with different signs of side slip angles are in dark blue and yellow, respectively. The normal cornering condition is in light blue.

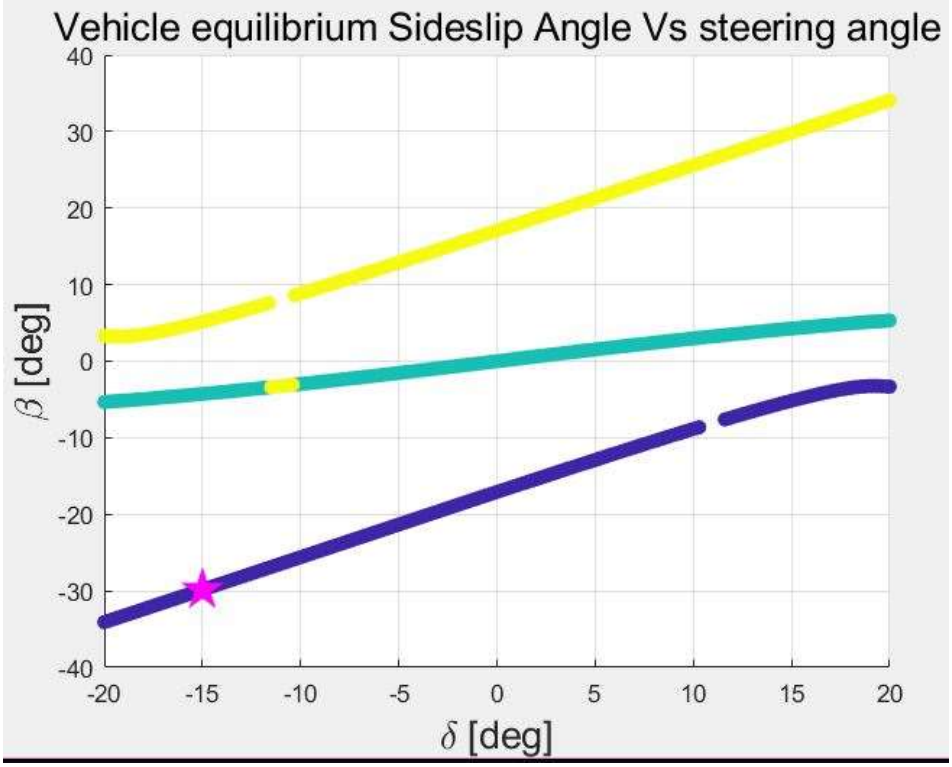

Figure 2. 3. Side slip angle to steer angle in equilibrium

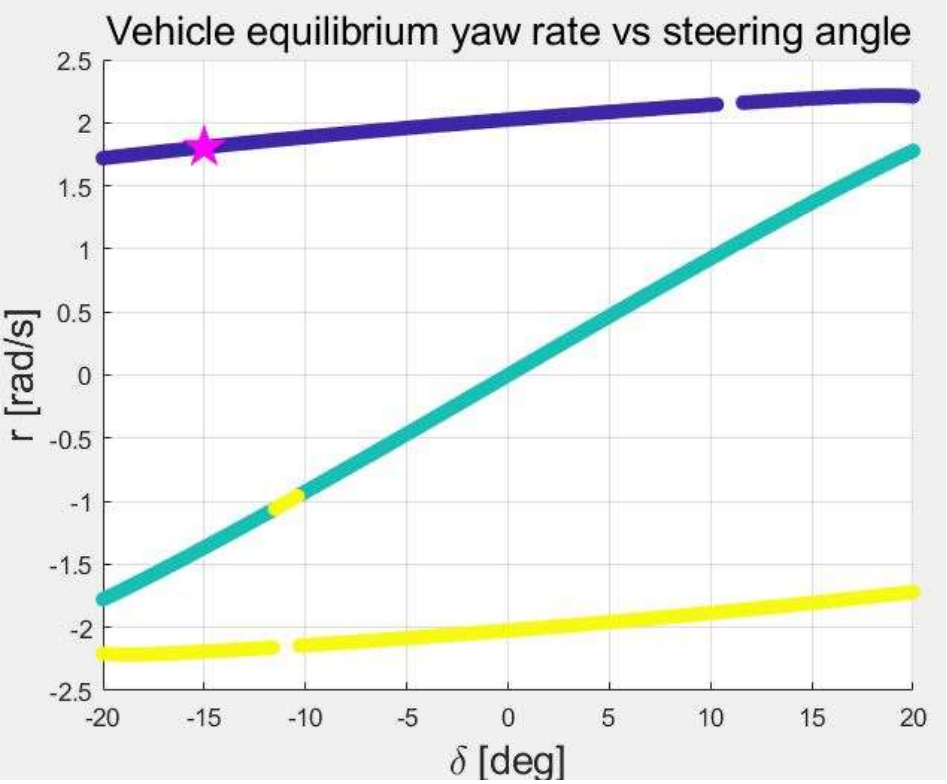

Figure 2. 4. Yaw rate to steer angle in equilibrium

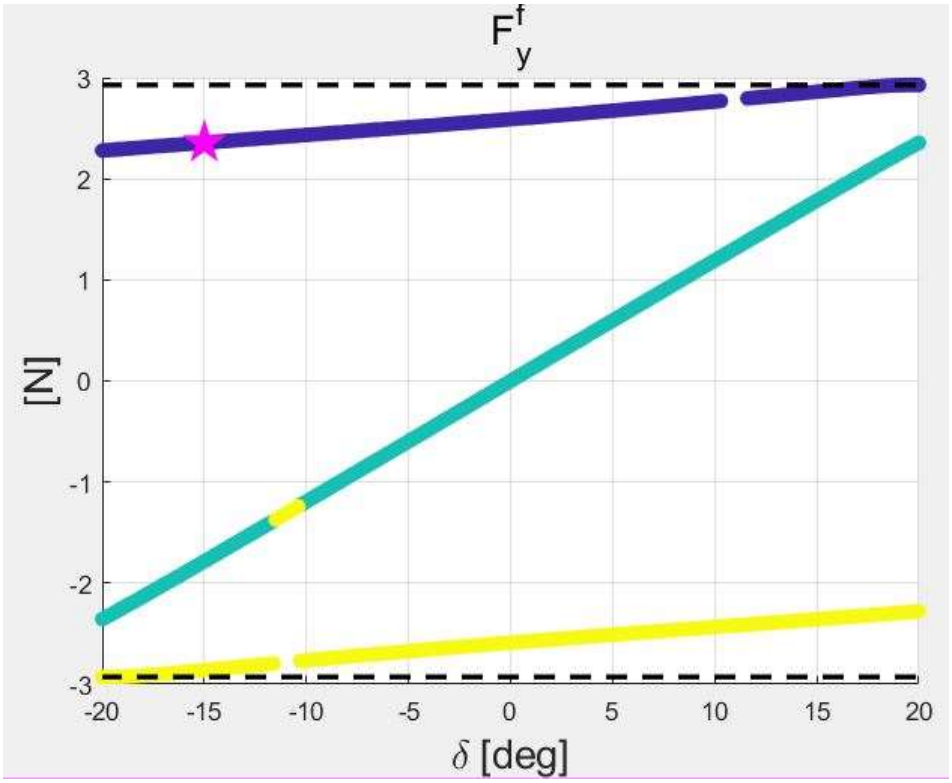

Figure 2. 5. Lateral force on front tire to steer angle in equilibrium

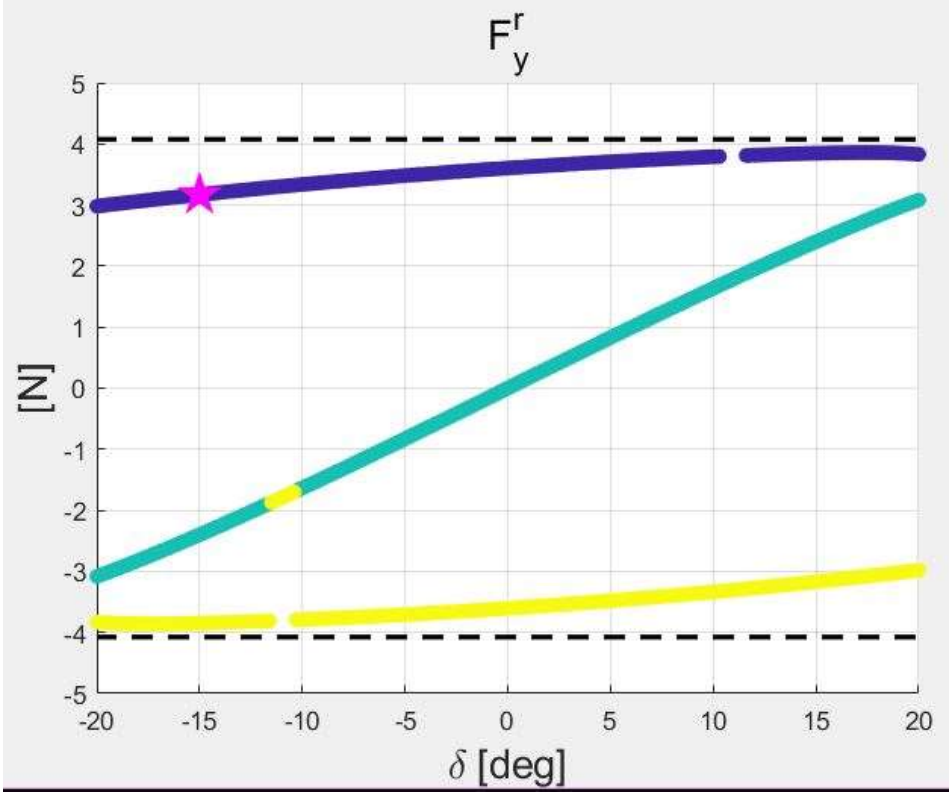

Figure 2. 6. Lateral force on rear tire to steer angle in equilibrium

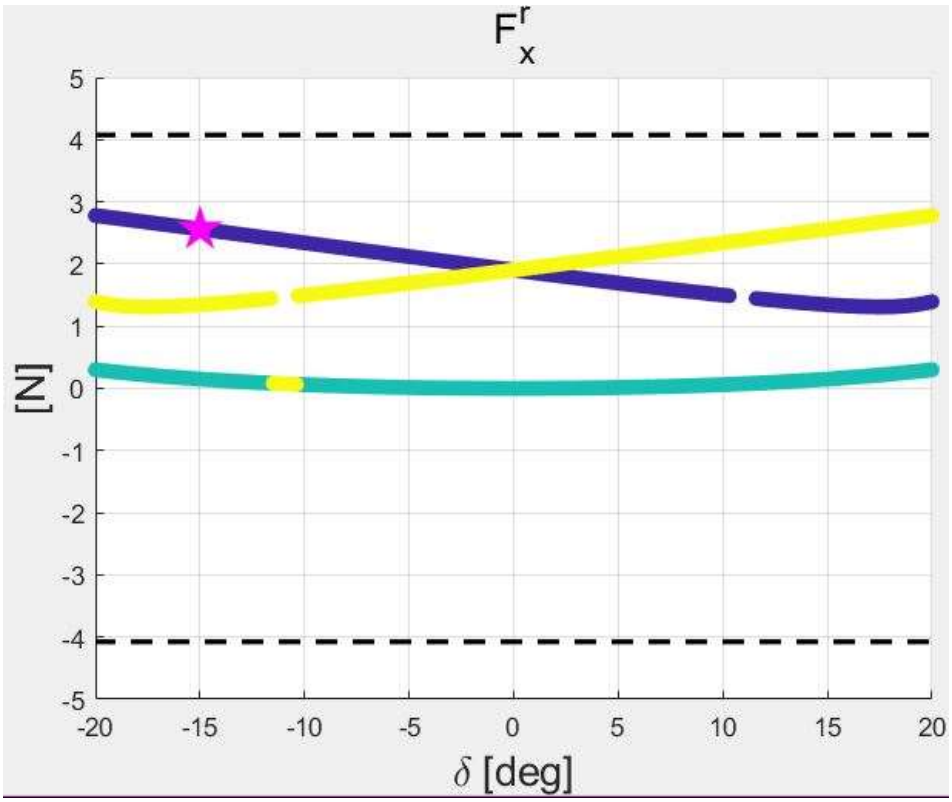

Figure 2. 7. Longitudinal force on rear tire to steer angle in equilibrium

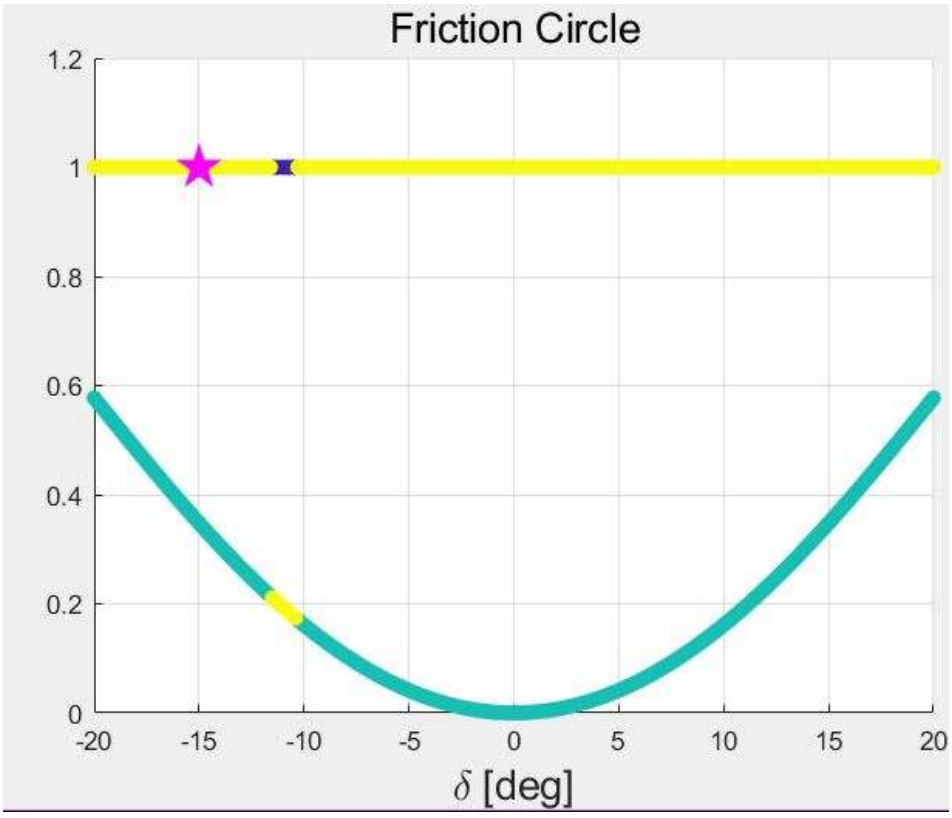

Figure 2. 8. Friction circle of rear tire in equilibrium

-Normal cornering

Figure 2.3 shows that no matter how much the steer angle is, the side slip angle always remains at a low value around zero, which indicates that the equilibrium of normal cornering is low side slip angle cornering. In this condition, to avoid a car going aside from the track, the side slip angle cannot be large.

Then, comes the yaw rate of normal cornering. The yaw rate is aligned with the steer angle. When a positive steer angle is given, which expects a left-hand turning, the yaw rate is also positive so that the car does have a left-hand turning. It is trivial because the steer angle controls the front wheel. When the front wheel turns to left-hand side, the car will just follow the wheel and go left.

Forces are also important variables in analyzing the equilibrium. The dashed black line in figure 2.5-2.7 indicates the limit. It is the limit of friction forces on the tire. As shown in figure 2.5, the front lateral force is far from the limit value in the function of δ. Thus, the front tire is not saturated without any doubts. As for the rear tire, apart from the edge part with high steer angle, the longitudinal force is always around zero (figure 2.7). The lateral force on rear tire is similar to the front tire and far from the limit line (figure 2.6). Therefore, the rear tire is not saturated as well, which is supported in the figure 2.8 of friction circle. All these features are consistent with the condition of normal cornering.

-Drifting

Since two drifting conditions are similar, except the sign, only one condition has to be taken into consideration, that is the condition with a negative side slip angle.

In figure 2.3, the dark blue line represents the drifting condition with negative side slip angle. It shows that the side slip angle has a dramatic range to change along the span of steer angle. When the steer angle is large, the side slip angle is over 20deg. It is an obvious drifting feature.

In figure 2.4, the steer angle and side slip angle are negative, whereas the yaw rate is positive. It is a condition that a driver tries to turn left but his steer wheels turn to right and the car is slipping out of the track to the right-hand side. This is called counter-steering or opposite lock in most of the cases.

The whole process can be shown in the figure 2.9. When the car turns right, it firstly steers slightly to the right. However, since the rear tire is saturated and the lateral rear force is limited, the rear part of body rotates due to the unbalanced moment of the car. Thereafter, the side slip angle occurs by the over-steering condition.

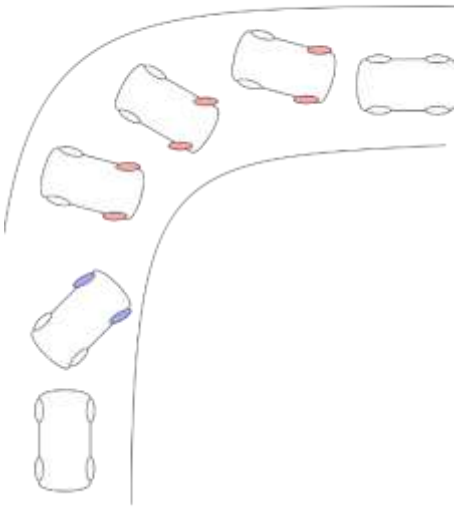

Figure 2. 9. Counter-steering

With the help of side slip angle, the car will slide aside from the track. In the meanwhile, the counter-steering appears and induces an opposite steer angle. This steer is used to keep the car not spinning too much and to maintain it in the desired track. This condition remains until the car finishes the cornering, and the saturation of rear tire is recovered.

As for the force of drifting condition, due to the counter-steering, all the forces have opposite signs to the steer angle. Moreover, the lateral force on the front tire is near the limit line. At the same time, the lateral force on the rear tire is also found near the limit line with a relatively large longitudinal force. It might not quite straightforward to judge whether the rear tire is saturated. In the friction circle in figure 2.8, it shows the ratio of the resultant forces on the tire over the maximum friction force. Since all the forces which act on the tire are from the friction, the resultant force must not exceed the unit circle.

$$\boldsymbol{F_{yr} + F_{xr} \leq \mu_r F_{zr}} \tag{2-26}$$

Where F_{yr} and F_{xr} are both vectors and F_{zr} is the normal force on rear tire and μ_r is the friction coefficient of rear tire.

In friction circle (figure 2.8), the light blue line which stands for the normal cornering is quite trivial, in which all the values are less than one. However the yellow line, which is overlap with dark blue line, is equal to one in the function of δ. These indicate that during the drifting condition, the rear tire of the car is saturated all the time.

Up to now, the main features or characteristics of drifting condition are found. At first, in a drifting condition, the front tire is nearly saturated and the rear tire is already saturated. Then,

during the drifting, the side slip angle β must be large enough. At last, the yaw rate of car has an opposite sign to the steer angle, which is counter-steering.

-Equilibrium point

Now it is time to choose one of the equilibrium points, and make it as the desired equilibrium. Since it should be a drifting condition, it is obvious that the side slip angle β should be large enough. The longitudinal velocity is set to be 1.5m/s in the present study. The unique equilibrium point along the span will be found directly as long as a desired steer angle is chosen.

In this study the steer angle δ is chosen to be -15deg, and the side slip angle is set to be negative. So, in this condition the car is turning left-hand side and the steer angle is opposite to the yaw rate because of counter-steering.

The parameters, which were estimated in previous work from our group, needed to describe the car, are shown in table 2.1:

$\boldsymbol{m}\ [kg]$	2.040	$\boldsymbol{\mu_f}$	0.35
$\boldsymbol{a}\ [m]$	0.1513	$\boldsymbol{\mu_r}$	0.35
$\boldsymbol{b}\ [m]$	0.1087	$\boldsymbol{C_f}\ [N/rad]$	47.86
$\boldsymbol{l}\ [m]$	0.2600	$\boldsymbol{C_r}\ [N/rad]$	127.77
$\boldsymbol{J_z}\ [kg/m^2]$	0.03		

Table 2. 1. Parameters of the RC car

The steer angle δ is set to -15deg and longitudinal velocity V_x is determined to 1.5m/s. Then the chosen equilibrium point is derived by solving the nonlinear equations. This point is shown as a purple star in the figures above (figure 2.3-2.8). It is in the dark blue line area and indicates a drifting condition with negative side slip angle.

Since all the equilibrium points along the span of steer angle δ is obtained, there is no need to compute the non-linear dynamic equations again to find the values of the desired equilibrium point. Instead, a simple searching with δ and the range of side slip angle will find a point which is the nearest one to the point we need.

The specific values of all the variables are shown in table 2.2:

$\boldsymbol{\delta^{eq}}$ $[\boldsymbol{rad}]$	-0.2618	$\boldsymbol{F_{yf}^{eq}}$ $[\boldsymbol{N}]$	2.3752
$\boldsymbol{r^{eq}}$ $[\boldsymbol{rad/s}]$	1.7934	$\boldsymbol{F_{yr}^{eq}}$ $[\boldsymbol{N}]$	3.1934
$\boldsymbol{\beta^{eq}}$ $[\boldsymbol{rad}]$	-0.5208	$\boldsymbol{F_{xr}^{eq}}$ $[\boldsymbol{N}]$	2.5329

Table 2. 2. Equilibrium value for variables

A double check of the saturated condition can be done here. The resultant force of rear tire is composed of F_{xr} and F_{yr} and the value is 4.0760 N. The friction force on rear tire is proportional to the normal force of rear tire with a parameter μ_r. The normal force F_{zr} can be obtained by the dimension of center axis. After the computing, the friction force turns to be 4.0760 N as well. Therefore the rear tire is saturated without any doubts.

2.7 Stability analysis

The features of normal cornering and drifting are discussed due to the figures of equilibrium points. The desired equilibrium point in drifting condition is also found. However, the stability of this point and whether there is bifurcation of equilibrium should be concerned about. Fortunately, there is a method called phase portraits which can help to know the property of the system in state space.

Phase portraits shows the dynamics of system with a certain V_x and steer angle δ. Since the longitudinal velocity V_x is fixed, the remaining state variables of the system are yaw rate and side slip angle. So a plane of yaw rate r and side slip angle β are created. Phase portraits plot all the possible trajectories of states from any initial conditions in this system. In figure 2.10 below, arrows are used to indicate the trajectories of state variables (β, r).

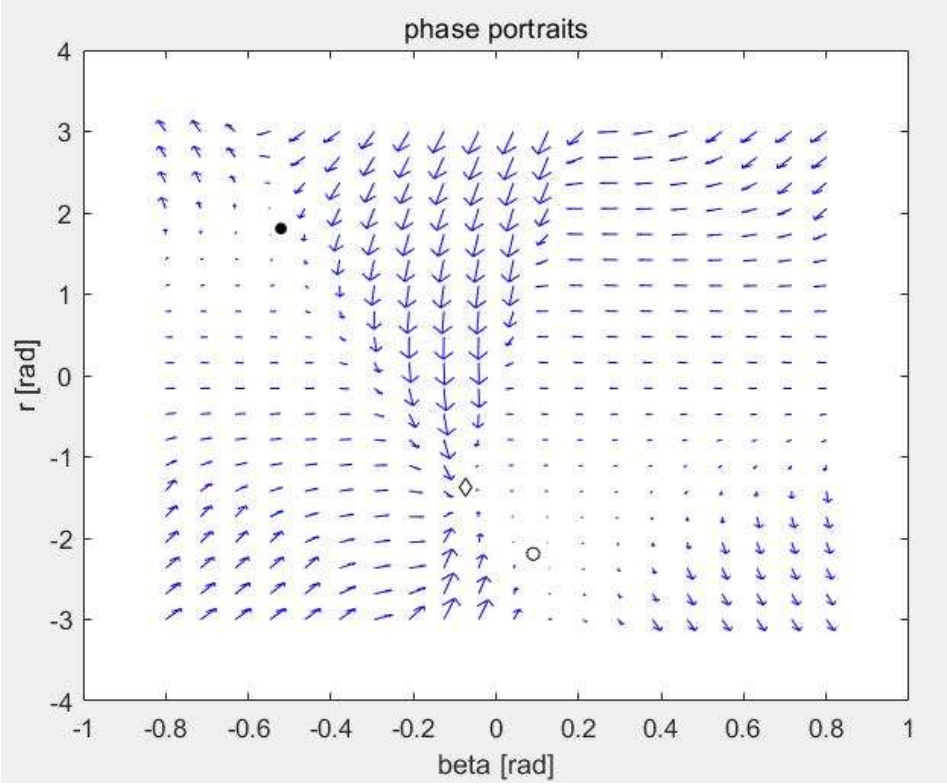

Figure 2. 10. Phase portraits and equilibrium point with δ=-15 [deg]

It is not hard to see that there are three equilibrium points in the phase portraits (figure 2.10). One point is shown with a black dot. The other two are on the third and fourth quadrants and shown with diamond and circle, respectively. These indicate that there is a normal cornering equilibrium and two drifting equilibria.

The point in diamond, is the equilibrium point with normal cornering. It is a stable node (figure 2.10). The arrows indicate the trend of side slip angle and yaw rate, and all the state points around have the tendency to be attracted to the equilibrium

The dot point on the left is the drifting condition with negative side slip angle. It is obvious that this equilibrium should be an unstable saddle. Since all the vectors in the neighborhood go away from this point, eigenvalues with one real positive and one real negative should be obtained if linearizing the equations around this equilibrium. This means that drifting equilibrium is an unstable condition and it is hard to maintain the vehicle at this point. Therefore, a controller that can stabilize this condition is worthwhile to explore, which is one of the targets in this study.

2.8 Bifurcation

When setting the steer angle to zero, a new figure of phase portraits can be drawn (figure 2.11). The diamond is still used to describe the point with normal cornering, in addition to the dot and the circle which correspond to the two drifting conditions. The diamond point is quite straightforward to consider viewed from figure 2.11. This stable node is just the condition that

the car simply goes ahead. Since the longitudinal velocity is fixed, the most trivial equilibrium is that both the yaw rate and side slip angle are zero. If there is no steer angle send to the car, the car will go forward.

The two drifting equilibrium points here are symmetric around the original. They are as unstable as they are before in figure 2.10, and the only difference between them is the orientation of the cornering.

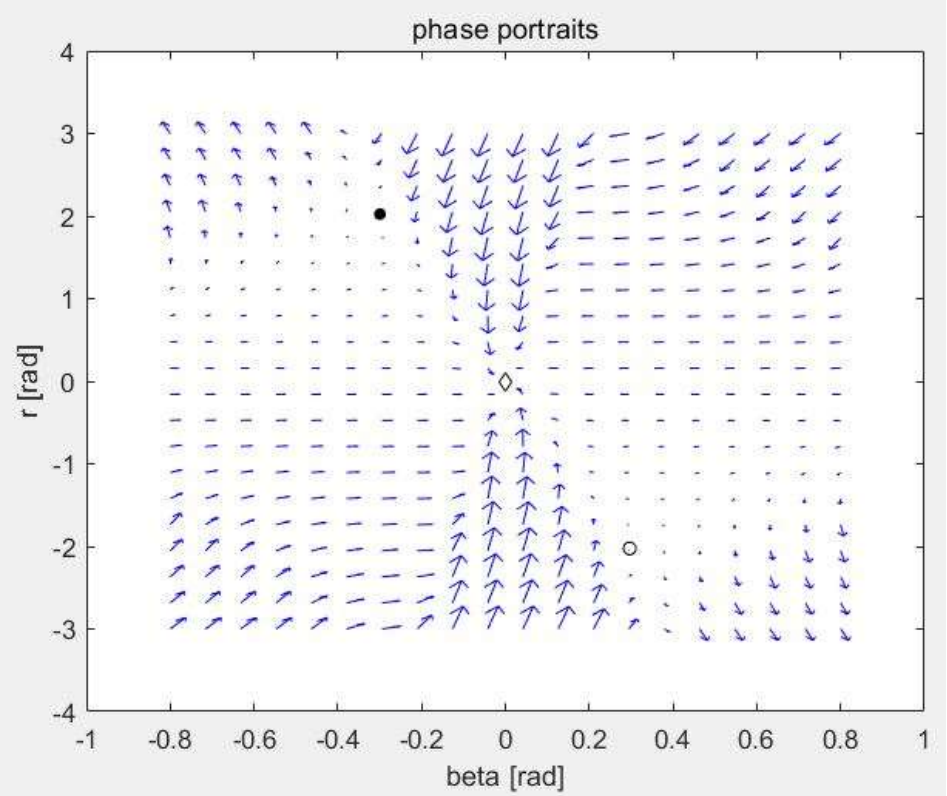

Figure 2. 11. Phase portraits with δ=0

Figure 2.12 is drawn when the steer angle is set to the half value between zero and -15. Comparing this figure with the formal one (figure 2.11), the stable node goes away from the original and to the third quadrant. At the same time, the other drifting equilibrium in circle corresponding to the right-hand drifting goes to the vertical axis a little bit.

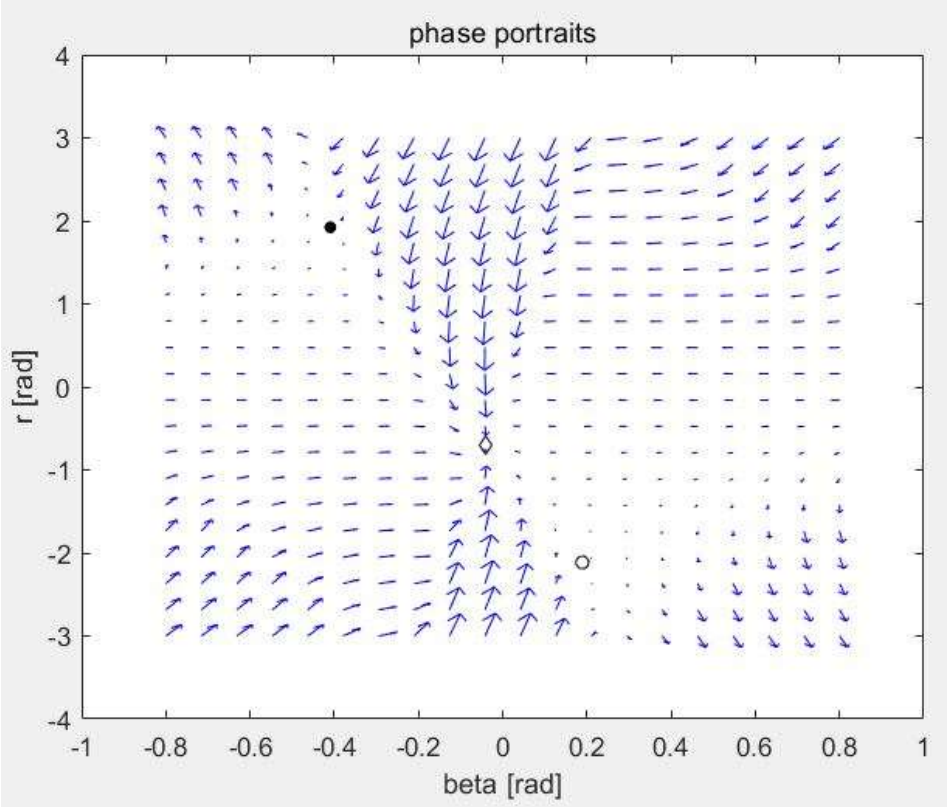

Figure 2. 12. Phase portraits with δ=-7.5 [deg]

The stable node here means that the car will converge to a normal and stable right-hand cornering. It is also an intuitive result. In most of the cases during cornering, the car will finally converge to a condition with a small side slip angle which has the same sign with respect to the steer angle, and a moderate yaw rate with respect to the steer angle. Moreover, the drifting condition corresponding to the left-hand side moves away from the vertical axis and obtain a larger side slip angle.

If comparing three figures together, a trend can be viewed. The normal cornering equilibrium is going nearer the drifting condition in the fourth quadrant. Thus, it can be predicted that if continuing increasing the value of steer angle, those two equilibrium points will finally collide and lead to a saddle-node bifurcation. In this bifurcation, those two points will annihilate each other and there will be no equilibrium point around that region. The only equilibrium point is the unstable saddle corresponding to left-hand drifting. Therefore a high steer angle may lead to a condition that no stable equilibrium points exist in the open-loop system.

Chapter 3 LQR control scheme

This chapter is focused on designing a proper controller to stabilize the car to a desired drifting equilibrium. The basic maneuver of drifting control is from a behavior of human being drivers. An LQR strategy is applied on a linearized system of the vehicle model. All the LQR parameters contributed to the control process are analyzed individually. At last, a feedforward-feedback control scheme with the LQR controller is designed for the original model in the study.

3.1 Analysis of controller

A general behavior of car during drifting is analyzed in this section. The car to drift is destabilized at first from a stable equilibrium in order to obtain enough large side slip angle. This process is achieved by inputting a steer angle with the same sign as the desired corner orientation and by increasing the throttle to make the rear tire saturated. When the saturated rear tire and the steer angle destabilize the car by a sudden yaw rate, it leads to an increasing of lateral velocity which results in a significant value of side slip angle.

Up to this step, the side slip angle starts increasing with the opposite sign to cornering angle. When the steer angle does not change much, the state of car will goes from the stable node to the left-hand unstable saddle as mentioned in chapter 2. At that time, a closed-loop controller should be activated. This controller stabilizes the drifting condition and keeps the car under this condition without spinning out until the cornering being finished. This process requires properly counter-steering and modified throttle.

Thus, the whole behavior is divided into two parts. The first part is to drive the car leaving away from steady state and going to a state around the chosen drifting equilibrium. The second part is to stabilize the car to the drifting equilibrium.

There is a suggestion about how to design a closed-loop controller for the second part[2]. It decouples the longitudinal part where the state is longitudinal velocity and the steer part where yaw rate and lateral velocity take the role of states. Thus, two controllers are designed.

It is a nice way to simplify the work but loses fidelity. The resultant force on the rear tire never exceeds the limit of friction force. When the car is going to drift, the rear tire is saturated. So, there is a constraint about the longitudinal force and lateral force.

$$F_{yr} = \sqrt{(\mu_r F_{zr})^2 - F_{xr}^2} \tag{3-1}$$

Due to the constraint (3-1), the longitudinal force has a relationship with the lateral force. When trying to control the longitudinal velocity with F_{xr}, the steer part must be influenced by the change of F_{yr}. The longitudinal part should never be decoupled from the steer part.

Another suggestion believes that the controller should be composed of two parallel goals[1].

One is to stabilize the car to the desired side slip angle, the other is to maintain the car around the desired longitudinal velocity. There is no need to decouple the longitudinal controller and the side slip angle controller. The relationship between the side slip angle and the yaw rate are considered firstly. Then followed by the establishment of a cascade control scheme, there are two loops in the controller. The outer loop uses a desired yaw rate with respect to yaw rate of equilibrium point and the error of side slip angle. It regulates the side slip angle to the desired value due to the relationship between yaw rate and side slip angle. The inner loop is to regulate yaw rate to the desired value. However, the design of this cascade controller is based on a physical intuition about the drifting dynamic. It still needs further stability analyses.

-Linear quadratic regulator

In this study, the Linear Quadratic Regulator (LQR) is used. LQR is one of the optimal control strategies. In optimal control, the aims of it is to find a control law that dealing with a problem of system at a minimum cost. It involves a system which can be described by a set of differential equations and a cost function which is respect to the state and control variables. When the differential equations are linear and the cost functions is described by a quadratic function, the solution of this system is provided by LQR.

In pole placement, the most important thing is to find proper closed-loop poles for a desired performance. Instead in LQR the error of state variables can be eliminate by the scheme for sure, but the system's performance during this period is not guaranteed. The main target in LQR is to find proper parameters of cost function to make the system work with a desired performance. However, there is no instruction way to guide how to choose the parameters of LQR, so it will be an iterative work during designing.

A continuous-time linear system is shown:

$$\dot{x} = Ax + Bu \tag{3-2}$$

If an infinite horizontal quadratic cost function is considered, this function should have the following form.

$$J = \int_{t=0}^{\infty} (x^T Q x + u^T R u) dt \tag{3-3}$$

Where x is a vector of state variables, and u is a vector of control variables. Q and R are the weighting factors of state variables and control variables, respectively. On the one hand, if choosing a large Q, the state variables will converge quickly but it requires a large control effort. On the other hand, if applying a large R, a large input energy is not required but the convergence of state error will be slower.

The derivation of LQR method is omitted in the present study. The result of control law is directly shown below.

$$u = -R^{-1}B^{T}Px \tag{3-4}$$

Where P is the result by solving the Riccati Equation:

$$PA + A^{T}P + Q - PBR^{-1}B^{T}P = 0 \tag{3-5}$$

Velenis[11] applied LQR to stabilize the linearized system by providing the steer command on the front tire and the slip ratio on the rear tire. He also applied a sliding mode controller to restrict the system along the linear region with wheel torque by the relationship between the angular rate of rear tire and the longitudinal slip. In this study, the throttle value is chosen as the input instead of the wheel torque, which is a command easy to manipulate for a driver. Another scheme is used to drive the system into a linear region rather than applying a slide mode.

3.2 Linearized system

The vehicle model is linearized before applying LQR. It is linearized around the drifting equilibrium which is chosen in chapter two. After linearization with the form of (3-2), the state matrix A is shown below.

$$A = \begin{bmatrix} \frac{\partial f_1}{\partial V_x} & \frac{\partial f_1}{\partial \beta} & \frac{\partial f_1}{\partial r} \\ \frac{\partial f_2}{\partial V_x} & \frac{\partial f_2}{\partial \beta} & \frac{\partial f_2}{\partial r} \\ \frac{\partial f_3}{\partial V_x} & \frac{\partial f_3}{\partial \beta} & \frac{\partial f_3}{\partial r} \end{bmatrix} \tag{3-6}$$

Where:

$$f_1 = \frac{1}{m}\left(F_{xr} - F_{yf}\sin\delta\right) + rV_x\tan\beta = \dot{V}_x \tag{3-7}$$

$$f_2 = \frac{1}{mV_x}\left(F_{yf}\cos\delta + F_{yr}\right) - r = \dot{\beta} \tag{3-8}$$

$$f_3 = \frac{1}{J_z}\left(aF_{yf}\cos\delta - bF_{yr}\right) = \dot{r} \tag{3-9}$$

As for the input matrix B, it is revised a bit. It is known that the input variables of a real car during the cornering should be throttle value and steer angle. However, the variables which directly affect the car's behavior should be the forces acted on the tire. So in the matrix B the

longitudinal force on the rear tire is chosen as an input which is corresponding to throttle, and the lateral force on the front tire is chosen as the other input which is corresponding to steer angle.

The lateral force on the rear tire F_{yr} is respect to the rear slip angle α_r, which corresponds to the state variables. The longitudinal rear force F_{xr} and the lateral front forces F_{yf} are the determined variables. As to the steer angle δ, it is described by rearranging the geometric equation of α_f (equation 3-10). Then the Fiala model can be used to map it to the determined lateral front force. By the way, here is an approximation on the tire slip angle. Since the magnitude of longitudinal velocity does not differ from the magnitude of resultant velocity much, the resultant V is replaced to longitudinal V_x for simplicity.

$$\delta = \beta + \frac{ar}{V_x} - \alpha_f \tag{3-10}$$

So, both inputs can be mapped to control variables which directly acted on the car. Then the input matrix B of linearized model should be in this form (3-11).

$$B = \begin{bmatrix} \frac{\partial f_1}{\partial F_{yf}} & \frac{\partial f_1}{\partial F_{xr}} \\ \frac{\partial f_2}{\partial F_{yf}} & \frac{\partial f_2}{\partial F_{xr}} \\ \frac{\partial f_3}{\partial F_{yf}} & \frac{\partial f_3}{\partial F_{xr}} \end{bmatrix} \tag{3-11}$$

Since the linearized system is around the desired equilibrium point, it can be rewritten as the following form (3.12).

$$\begin{vmatrix} \delta\dot{V}_x \\ \delta\dot{\beta} \\ \delta\dot{r} \end{vmatrix} = A \begin{vmatrix} \delta V_x \\ \delta\beta \\ \delta r \end{vmatrix} + B \begin{vmatrix} \delta F_{yf} \\ \delta F_{xr} \end{vmatrix} \tag{3-12}$$

The variables with a prefix δ mean that they are errors from the desired equilibrium point. It can be viewed that three poles are composed of a real positive value and one pair of conjugate values with negative real part. It is obvious an unstable equilibrium in the open-loop.

poles in linearized sys:

Pole	Damping	Frequency (rad/seconds)	Time Constant (seconds)
4.56e-01	-1.00e+00	4.56e-01	-2.19e+00
-5.23e-01 + 2.37e+00i	2.15e-01	2.43e+00	1.91e+00
-5.23e-01 - 2.37e+00i	2.15e-01	2.43e+00	1.91e+00

Figure 3. 1. Poles in linearized system

3.3 LQR control scheme

3.3.1 LQR for Linearized system

As showed in section 3.1, the LQR requires choosing a proper pair of weighting factors Q and R. In general, they are chosen as diagonal matrices. The diagonal values of Q for each variable affect their weighting in cost function and convergence speed. In the meanwhile, the diagonal values of R shows the control effort of two inputs.

Since there is no instructive suggestion to choose a proper value for each element in parameter Q and R, a variety of parameters are tested to adjust an optimal value for the desired behavior. It is a large work for study. Fortunately, there is a typical choice of Q and R[25] as the guideline.

$$Q = \begin{bmatrix} q_1 & & & \\ & q_2 & & \\ & & \ddots & \\ & & & q_n \end{bmatrix}, \quad R = \rho \begin{bmatrix} r_1 & & & \\ & r_2 & & \\ & & \ddots & \\ & & & r_m \end{bmatrix}$$

$$q_i = \frac{1}{t_{si}(x_{imax})^2}, \quad r_i = \frac{1}{(u_{imax})^2}, \quad \rho > 0$$

Figure 3. 2. Typical choice of Q and R[25]

In figure 3.2, X_{imax} and U_{imax} are the constraints on the individual state and input, respectively; t_{si} is the desired settling time of the individual state; ρ is chosen to balance the regulation of state and control effort. Since two inputs here are corresponding to two forces

with constraints, R matrix is chosen as the form in figure 3.2 but with different factors of two inputs (form 3-14). As for the Q matrix, there are no explicit constraint of three state variables. So it is chosen as a unity matrix with three different factors for those three diagonal values (form 3-13).

$$Q = \begin{bmatrix} q_{11} & 0 & 0 \\ 0 & q_{22} & 0 \\ 0 & 0 & q_{33} \end{bmatrix} \tag{3-13}$$

$$R = \begin{bmatrix} \rho_1 r_1 & 0 \\ 0 & \rho_2 r_2 \end{bmatrix} \tag{3-14}$$

Then the control law of linearized system is obtained.

$$K = R^{-1} B^T P \tag{3-15}$$

$$\delta u = -K \delta x \tag{3-16}$$

In MatLab, there is no need to solve the Riccati Equation to obtain P in the description of feedback control law. When the system is linearized to matrix A and B, and the parameters of Q and R are chosen, the optimal gain matrix from the function is computed by MatLab directly (figure 3.3).

```
[P, S, K]=care(A, B, Q, R);
%[K, S, e]=lqr(A, B, Q, R);

Kvxfy=K(1, 1);
Kbetafy=K(1, 2);
Krfy=K(1, 3);
Kvxfx=K(2, 1);
Kbetafx=K(2, 2);
Krfx=K(2, 3);
```

Figure 3. 3. LQR in MatLab

In figure 3.3, every gain from state variables to control variables is specified. The closed-loop system is shown in the following formula (3-17):

$$\dot{\delta x} = (A - BK)\delta x \tag{3-17}$$

It can be viewed that the closed-loop poles are composed of two conjugates with negative real part and a negative real value. So the equilibrium should switch to a stable equilibrium.

poles in closed loop:

Pole	Damping	Frequency (rad/seconds)	Time Constant (seconds)
-2.10e+00 + 1.53e+00i	8.09e-01	2.60e+00	4.76e-01
-2.10e+00 - 1.53e+00i	8.09e-01	2.60e+00	4.76e-01
-1.31e+01	1.00e+00	1.31e+01	7.64e-02

Figure 3. 4. Poles in closed-loop

If pulling the state away from the equilibrium a bit, it should return to the equilibrium after a while. However, since both the longitudinal rear force and the lateral front force have a limit subject to the friction force, two input variables must not exceed the limit.

A set of initial values is chosen:

$$\delta x_0 = [\delta V_{x0} \quad \delta \beta_0 \quad \delta r_0]^T \tag{3-18}$$

The car is starting with a normal cornering with this initial state. Figure 3.5 shows that the closed-loop controller perfectly stabilizes the car to the desired drifting equilibrium.

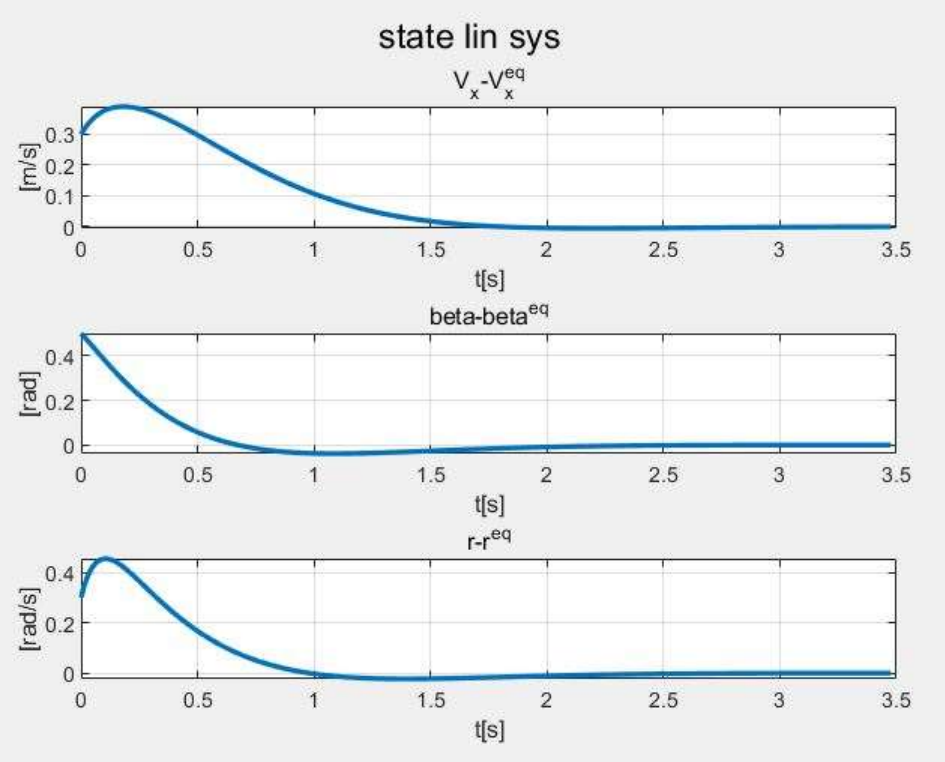

Figure 3. 5. State of linearized closed-loop system

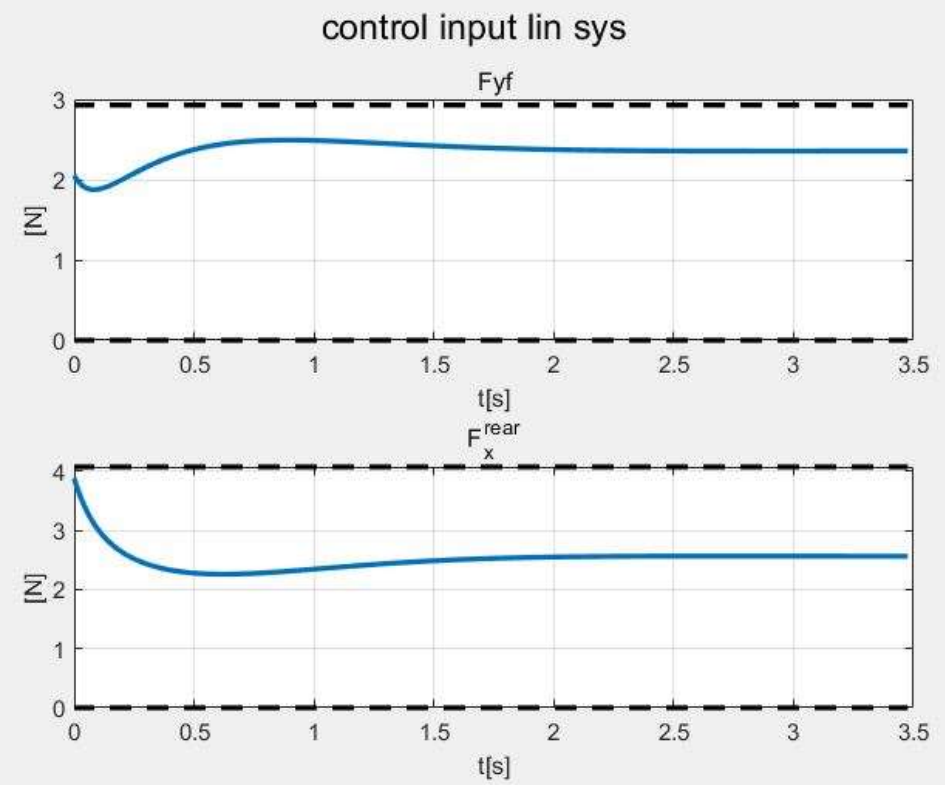

Figure 3. 6. Input of linearized closed-loop system

Three state errors are return to zero with some time and in the meanwhile two input variables do not exceed the dashed lines which are the constraints. Please be aware, the input variables in figure 3.6 are not δu but the total value.

Since the chosen LQR parameters are not the unique solution, there are several results with respect to the adjustment of all the parameters.

The first element in Q matrix, which is q_{11}, is modified. Three scales of q_{11} value are chosen, q_{11} is the value used in the controller, the other two are 5 times of and one fifth of q_{11}, respectively.

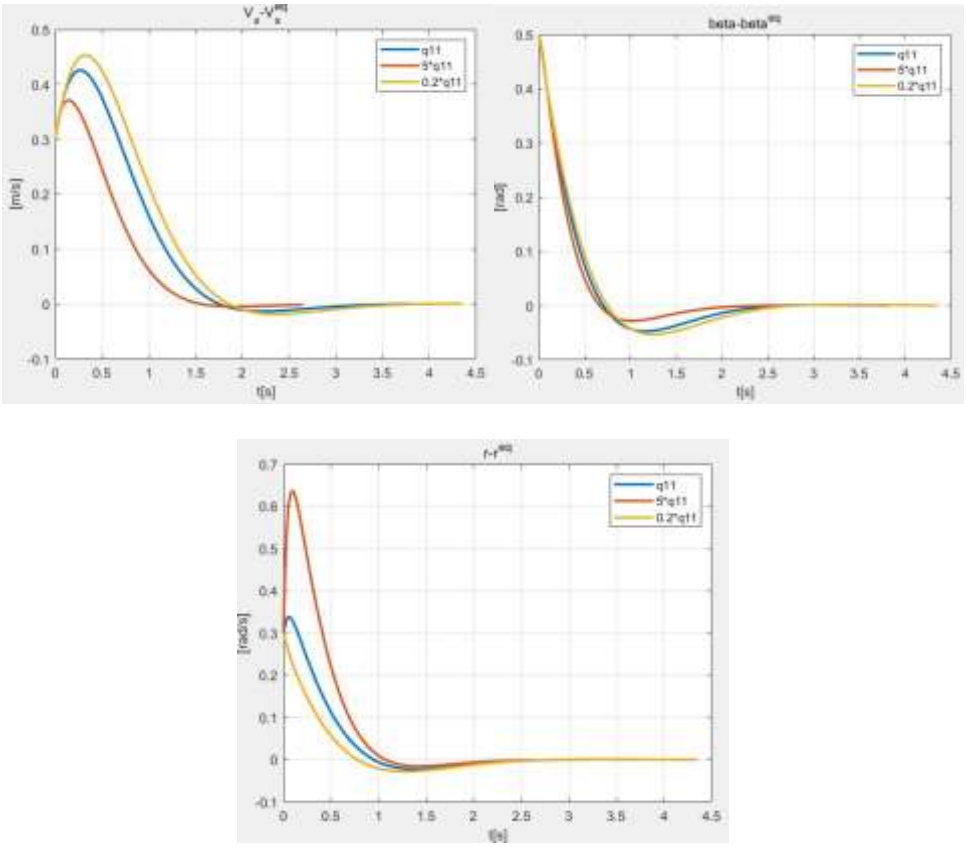

Figure 3. 7. Three states with respect to q_{11}

In figure 3.7 when increasing q_{11}, the settling time of three states will be reduced. The larger the q_{11} is, the faster the longitudinal velocity and the side slip angle are stabilized. But as for the yaw rate, a large q_{11} will lead to an overshoot peak during the transient of r.

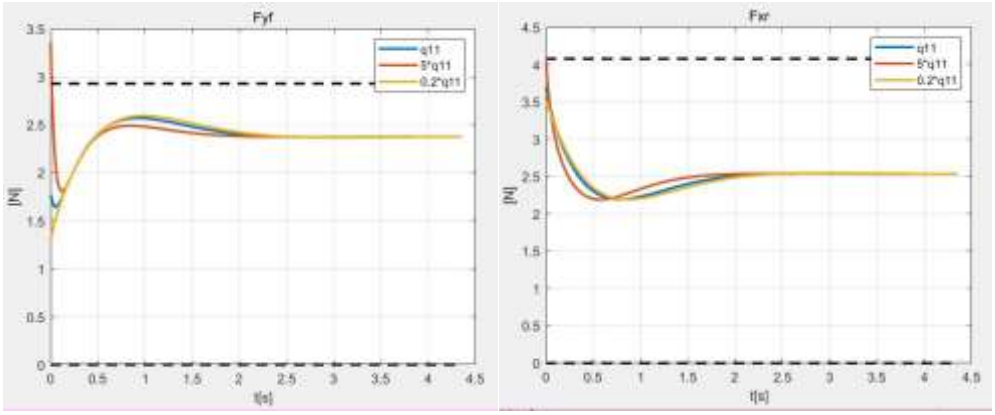

Figure 3. 8. Input force with respect to q_{11}

In figure 3.7 and figure 3.8, if it requires a higher performance of settling time, harder control effort is needed during the transient. However, as two input forces are limited by the total friction, q_{11} should be restricted in a relative moderate range. Otherwise an input beyond limit may lead to an unpredicted result.

Then the second diagonal value of Q, q_{22} is modified.

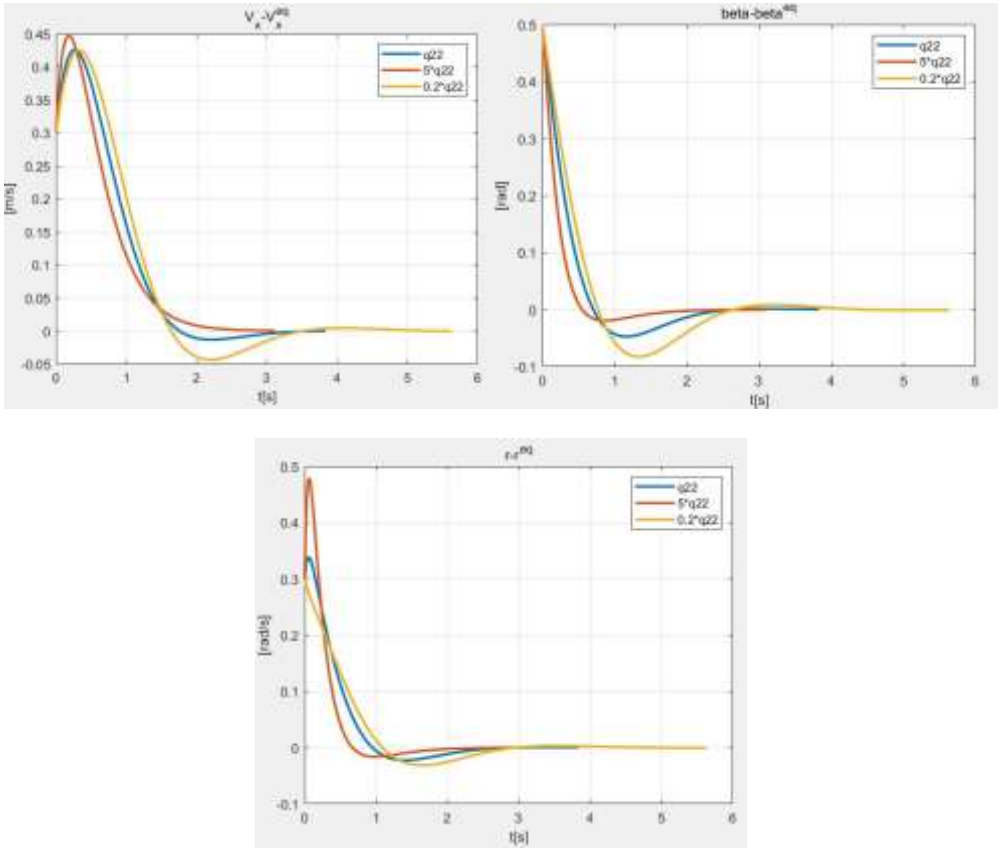

Figure 3. 9. Three states with respect to q_{22}

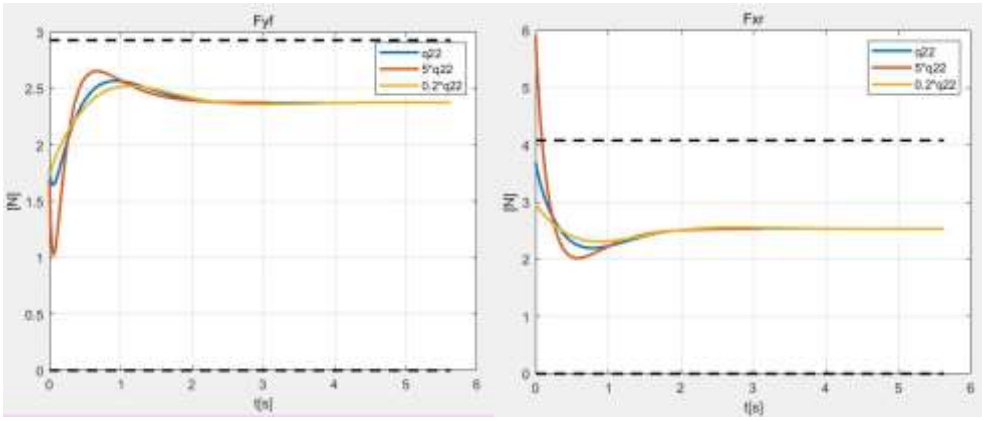

Figure 3. 10. Input force with respect to q_{22}

The parameter q_{22} has a similar behavior to q_{11}. A large q_{22} will make a quick settling on V_x and r, but in the meanwhile it applies large overshoot. On the contrary, the large q_{22} gives a good performance of side slip angle. When the performance of side slip angle β is considered to be the dominated one, this q_{22} is a good factor to modify. In figure 3.10, a high q_{22} value requires a high value of longitudinal force. It makes sense because a high performance of side slip angle requires a fast transient of rear tire's saturation.

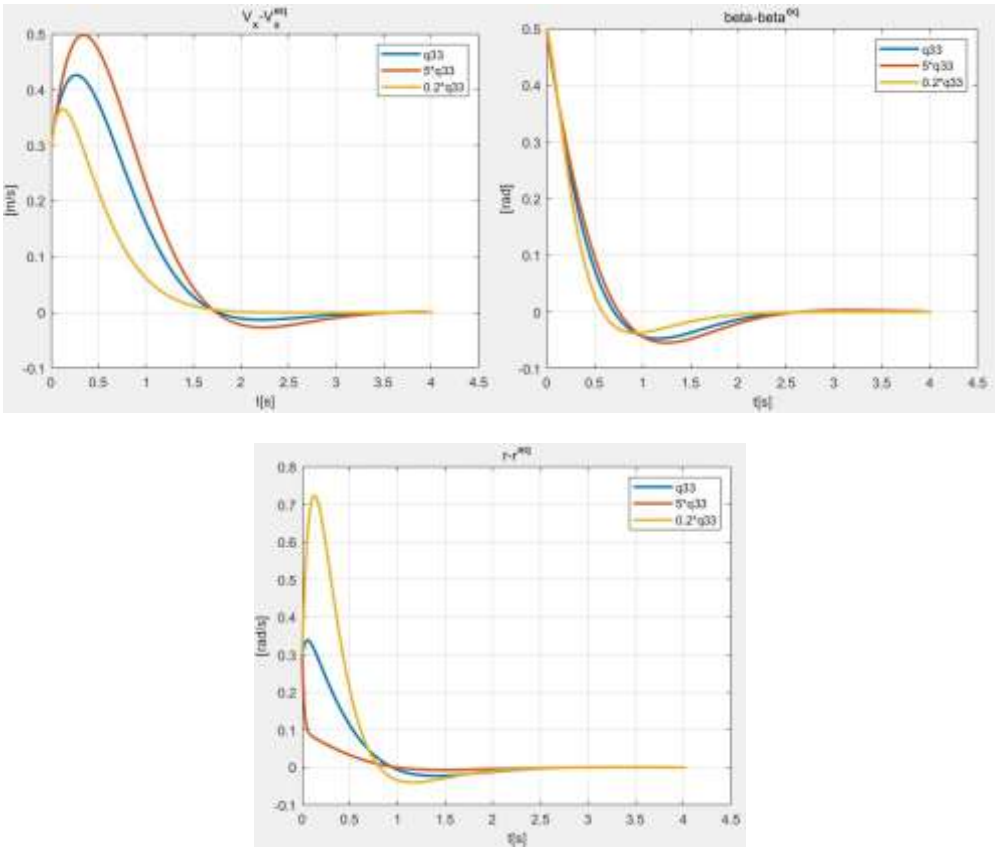

Figure 3. 11. Three states with respect to q_{33}

As to the last diagonal element q_{33}, this one has a significant effect on the yaw rate. If

increasing its value, the behavior of yaw rate can be improved a lot. But in the meanwhile, the transient of longitudinal velocity and side slip angle are worsen. Thus, this parameter can be modified in a condition that a high performance of yaw rate is mainly required.

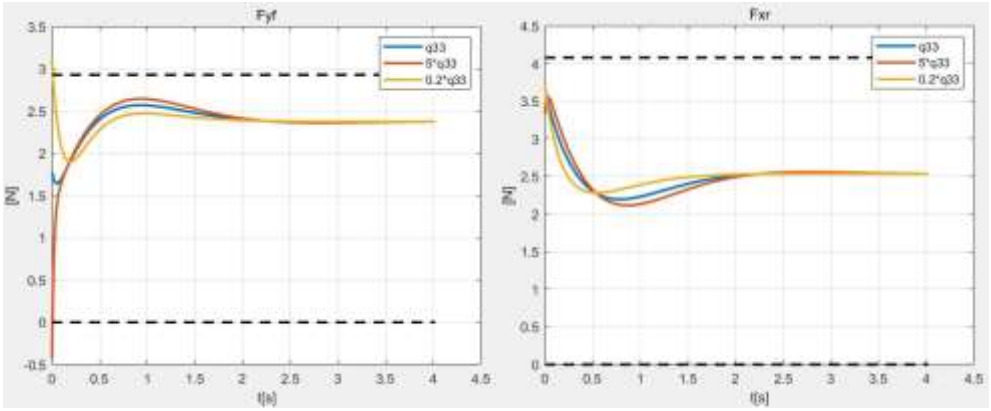

Figure 3. 12. Input force with respect to q_{33}

The effect of q_{33} on lateral front force F_{yf} is quite interesting. A large parameter requires a less initial input, while a small parameter may lead to a high input which exceed the friction limit. Since either of them is expected to happen, a moderate value of q_{33} should be chosen.

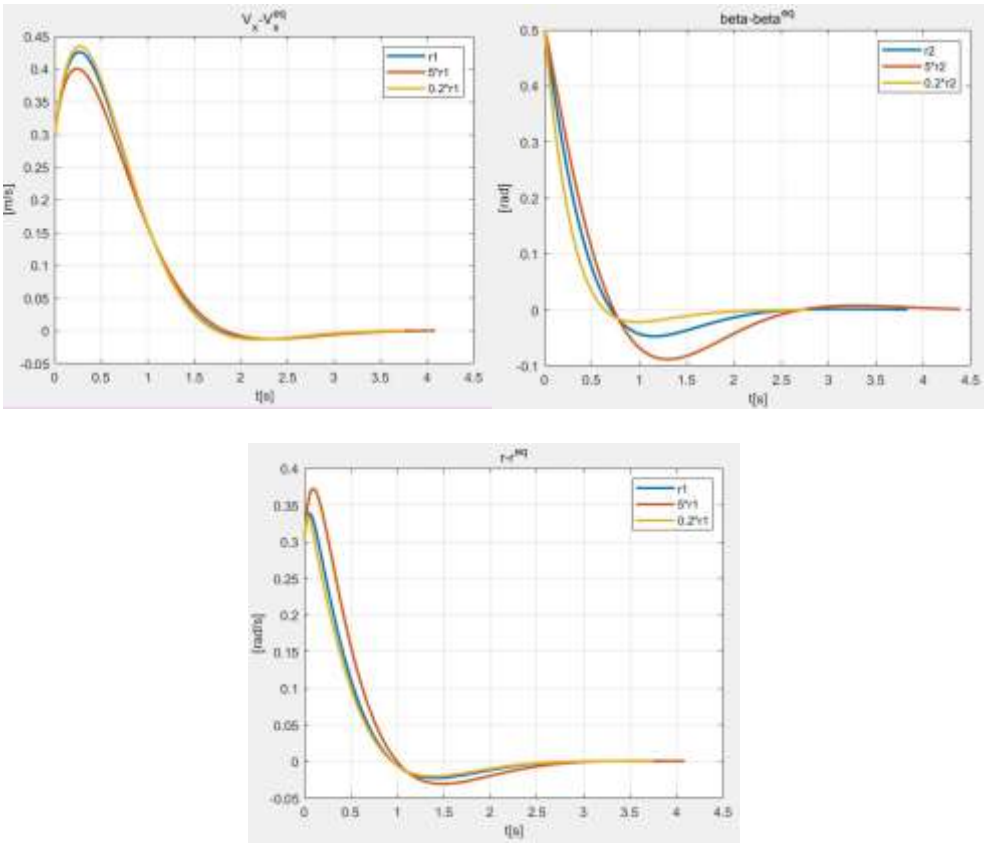

Figure 3. 13. Three states with respect to r_1

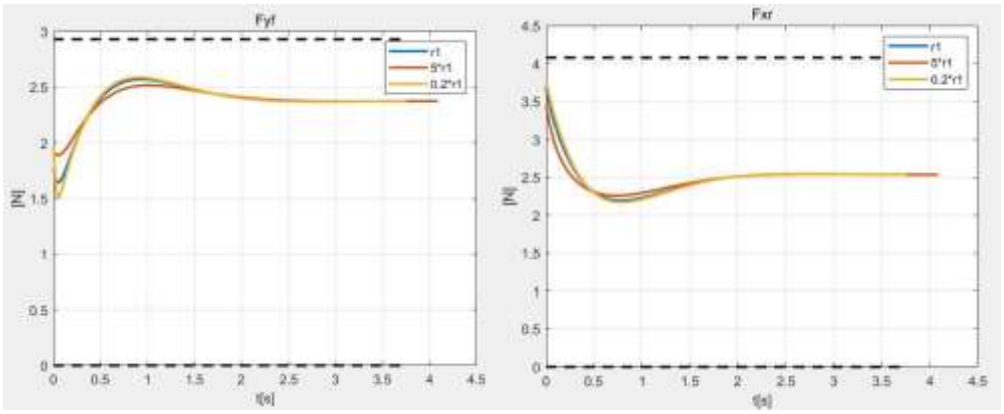

Figure 3. 14. Input force with respect to r_1

The parameter of r_1 and r_2 are factors to modify the control effort. In figure 3.14, a lower r_1 does not influence both the state's transient behavior and the control effort too much except the side slip angle. Furthermore, a less r_1 can improve the performance of side slip angle. When the system is under a condition that requiring a better performance of side slip angle without changing others' performance much, reducing r_1 could be a good choice.

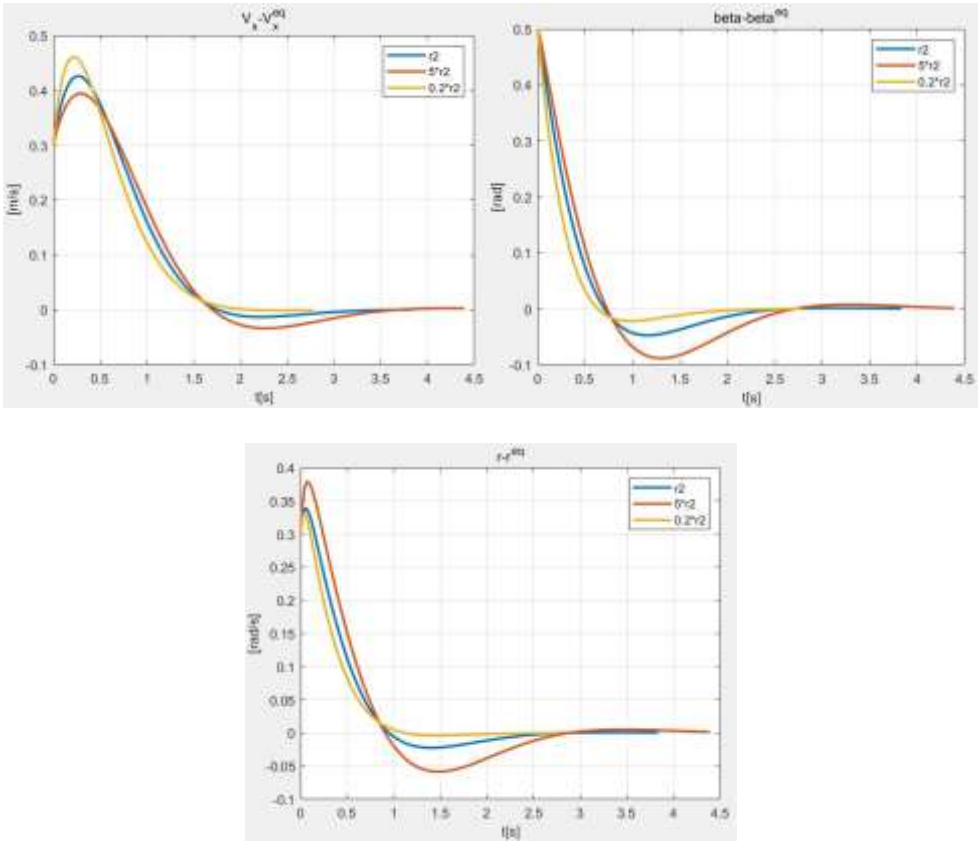

Figure 3. 15. Three states with respect to r_2

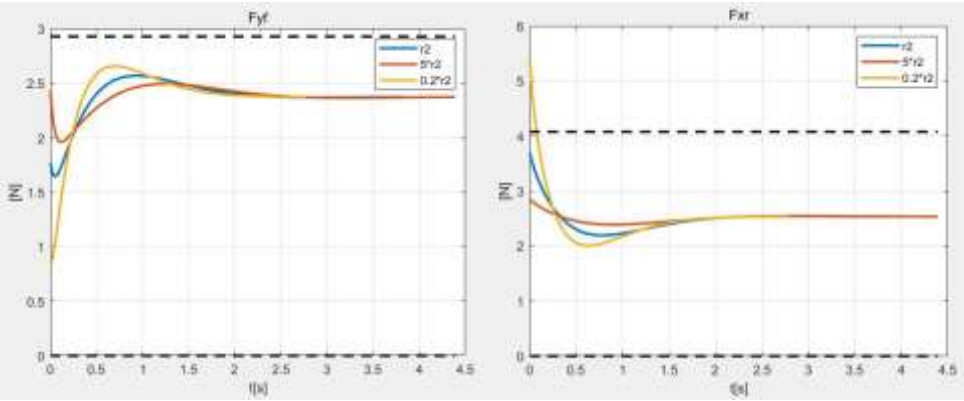

Figure 3. 16. Input force with respect to r_2

In figure 3.15 and figure 3.16, a less r_2 improves the settling time of three states and worsens the performance of longitudinal velocity V_x a bit. As to the input forces, this requires a high longitudinal rear force as the tradeoff.

Since all those five parameters give some contributions to performance of the whole system, they are tuned due to their behavior for many times so as to find good results of the system.

3.3.2 LQR for Nonlinear system

Although applying LQR in the linearized system is successful, the original model is nonlinear. It can be imaged that if there is no error from the desired equilibrium, the car should remain at that point forever with all the equilibrium values obtained in the 2.6. Thus, the total control law is not difficult to be found out with a feedforward-feedback form (3-19, 3-20).

$$u = \delta u + u^{eq} \tag{3-19}$$

$$u^{eq} = \begin{vmatrix} F_{yf}^{eq} \\ F_{xr}^{eq} \end{vmatrix} \tag{3-20}$$

The real forces applied to the tire are the addition of error parts and equilibrium parts. That is why the limit of force should be the total value in the previous section. In order not to lead to any unpredicted condition, the input values should not exceed the constraint.

$$u_{max} = \begin{vmatrix} \mu_f F_{zf} \\ \mu_r F_{zr} \end{vmatrix} \tag{3-21}$$

$$u_{min} = \begin{vmatrix} -\mu_f F_{zf} \\ -\mu_r F_{zr} \end{vmatrix} \tag{3-22}$$

Another problem arises. In the linearized model, a normal cornering condition is tested as the initial state and the controller succeeds to drive the car back to the equilibrium. However it is hard to say if the controller can still drive the car from another initial point to the equilibrium without violating the limit.

A possible solution[26] is shown here. This solution is similar to the control law discussed[2] in section 3.1. It separates the control scheme into two parts. The first part is an offline open-loop control. It gives a sequence of input commands to drive the car from its initial condition to a state which is near the desired equilibrium point. When this open-loop has finished its work, the closed-loop LQR is activating and stabilize the car to the desired equilibrium.

It is an intuitive solution. When a driver would like to make a car drifting, it should be easier to start with a cornering condition at first. After the side slip angle increasing and the car being away from the steady state, the driver focuses on stabilizing it to the drifting condition.

3.3.3 Open-loop scheme

As mentioned above, different initial states of the car will change the behavior a lot. An initial state which is near the desired drifting equilibrium will reduce the work load of controller a lot, and the input forces will also be moderated during the stabilization process. On the contrary, an initial state which is far from the desired point, for example a state that the car is standing still, will require large control effort to drive the car to the equilibrium. So choosing a proper initial state to activate the controller is important.

There is a way[26] to find a precise result about the open-loop region. The requirement can be described as the following form:

$$\max \ \gamma$$

$$s.t. \ \delta x^T P \delta x \leq \gamma$$

$$H\delta x \leq W$$

Where:

$$H = \begin{bmatrix} K \\ -K \end{bmatrix}, \qquad W = \begin{bmatrix} u_{max} - u^{eq} \\ u_{min} - u^{eq} \end{bmatrix}$$

Then due to Karush–Kuhn–Tucker (KKT) condition, it deduces that:

$$\gamma = \min\left(\frac{w_i^2}{\boldsymbol{h_i} P^{-1} \boldsymbol{h_i^T}}\right) \tag{3-23}$$

Where w_i is the i-th element of matrix W and h_i is the i-th row of matrix H.

This gives a result that all the states satisfying $\delta x^T P \delta x \leq \gamma$ can converge to the desired equilibrium point without violating the constraints. With the help of this reference region, an open-loop control is designed to drive the car at the beginning. When the state variables satisfying the γ is reached, the closed-loop gain will activate. The scheme in Simulink model can be viewed in figure 3.17 below.

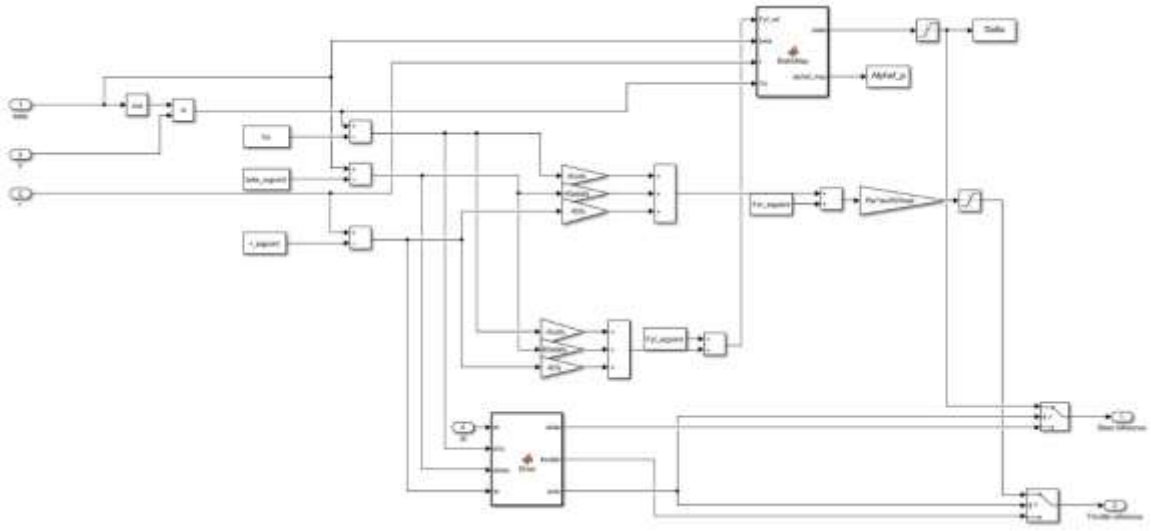

Figure 3. 17. Open-loop scheme

In this scheme, three state variables are obtained from the model and their errors are computed from the equilibrium values. Then the results of errors will go to two parts. In the closed-loop part, it computes the control variables with LQR feedback gains and adds the equilibrium input values to obtain the total inputs. Then the longitudinal rear force will be transferred to throttle by the transmission parameters and the lateral front force will be mapped to steer angle δ with the help of inverse Fiala model.

In the open-loop part, which is described by a function block in figure 3.17. A proper acceleration and steer angle are given at the beginning in order to drive the car to a normal cornering. Then the state variables of normal cornering are collected and checked. If the state variables are restricted in the region of γmin, this is a fine initial state to apply closed-loop controller. Otherwise, the acceleration and steer angle are modified to drive the car to another normal cornering condition.

However, after simulating this scheme, the region γ_{min} is found to be too strict and it is not an explicit threshold for to judge. Moreover, the further work in ROS cannot easily support a switch block from Simulink model, so another solution instead of it should be found.

3.3.4 Control scheme without open-loop part

Since a two-stage scheme with an open-loop in prior is not applicable in the present study, the feedforward-feedback control scheme is directly applied to the car from its original condition. In practice, the controller can stabilize the car to the desired equilibrium even though the inputs suffer a period of saturated values as long as optimized parameters of LQR are introduced. It is a tradeoff between the tuning and designing. It is known that the manipulation of the stabilization region mentioned above will lower the tuning work load of LQR parameters. However, since it is hard to manipulate the stabilization region of LQR, optimizing the parameters of LQR is used to solve this problem in the present study. So, the new control scheme without open-loop is shown in figure 3.18. In this scheme, the controller works from the initial state of the model. There is no doubt that the two commands has a period of saturation. The performance of the system is guaranteed by finding the optimized LQR parameters.

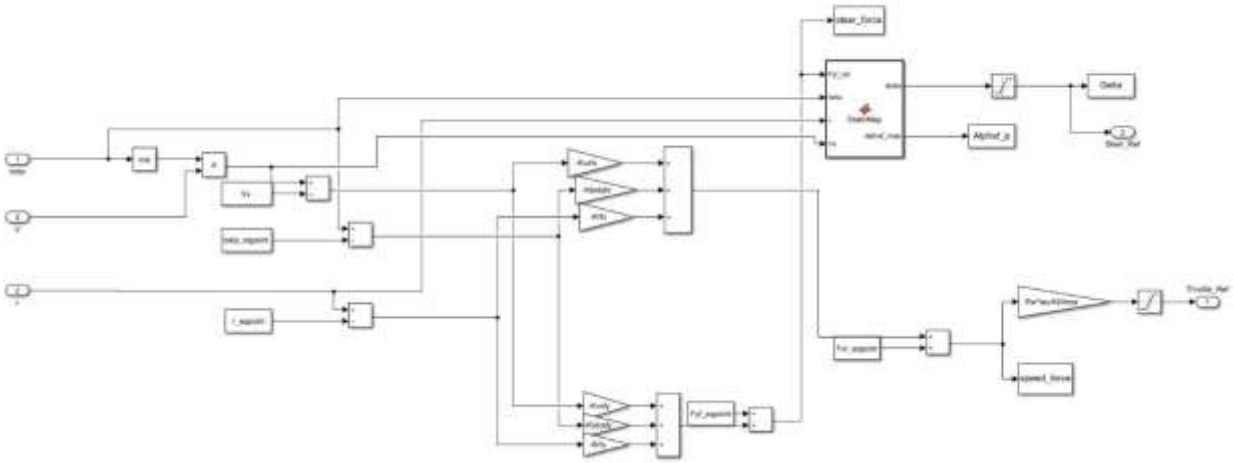

Figure 3. 18. Control scheme without open-loop part

3.4 Total scheme

In addition to the controller discussed above, the single track model in Simulink will be introduced in this section. This model mainly consists of two parts (figure 3.19-3.20). One part is about the state equations of single track model, which involves three state variables and their respective dynamics (figure 3.19). The other part is about forces derived from Fiala tire model, which allows to compute the lateral forces on the wheels and send them, thereafter, to the dynamic block for completing the differential equations of state variables (figure 3.20).

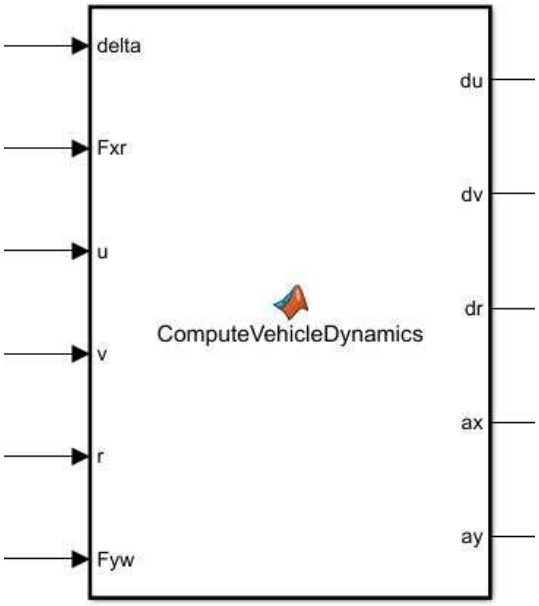

Figure 3. 19. Dynamics of single track model

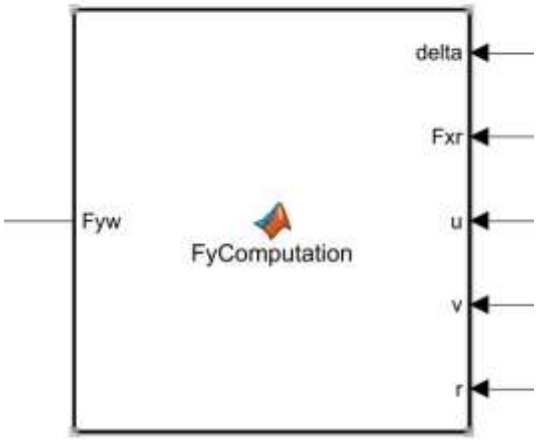

Figure 3. 20. Force computation

In order to close the control loop, a part that collects state variables and delivers them to the controller is required. In Simulink, the state variables are assumed to be sent to the controller directly if there are ideal sensors available to measure the variables. Thus, the total scheme is shown in figure 3.21.

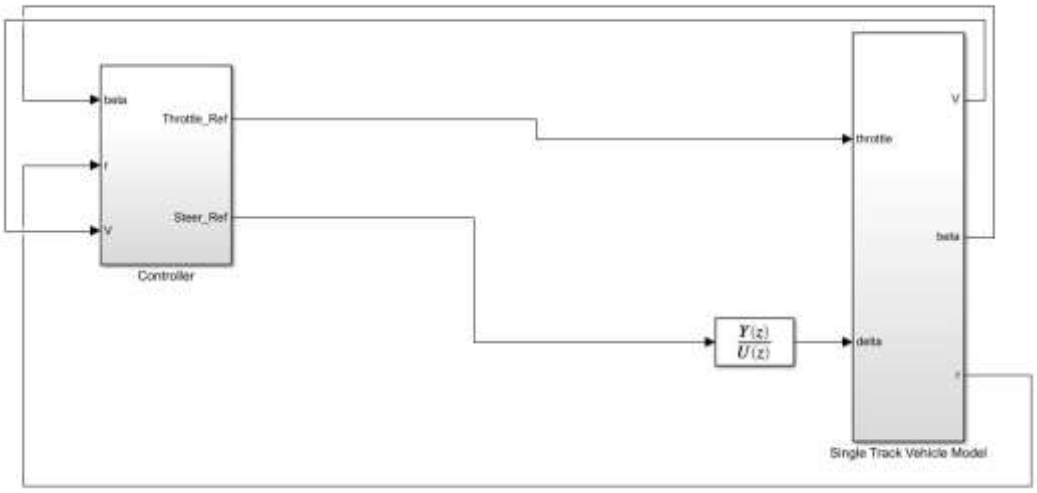

Figure 3. 21. Total scheme

3.5 Throttle transmission

As mentioned in the previous section, the longitudinal rear force F_{xr} is obtained from the LQR. If ignoring the dynamics of angular rate of wheels, there will be a balance between the motor torque and the friction torque acted on the tire.

In that case, the throttle value can be obtained by transferring F_{xr} with a gain block. The parameters in that block are shown in table 3.1 below:

R_{wheel} $[m]$	0.0245	K_t $[Nm/A]$	1/340.34
τ	0.09799	I_{max} $[A]$	13

Table 3. 1. Transmission parameters

Where R_{wheel} is the outer radius of the car, τ is the total transmission ratio from wheel to motor, K_t is the toque constant of the motor and I_{max} is the maximum current of motor.

As shown in figure 3.22, the addition block is composed of equilibrium F_{xr} and δF_{xr} from LQR. The saturation block at last limits the throttle value within the range of [-1, 1].

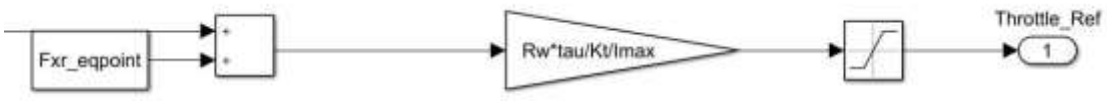

Figure 3. 22. The commands flow from F_{xr} to throttle

3.6 Delay of steer command

It is known that giving steer command to the front wheels is not an instant process. There is an actuator delay from commands sending to the wheels reacting. The delay estimated from previous work in our group is shown below:

$$\tau_{delay} = 0.09[sec]$$

$$\omega_f = 8\ [Hz]$$

Thus, there should be a filter between the desired steer angle and the real applied steer angle with a time delay of 0.09 second and a bandwidth of 8 Hz (equation 3-24). This information can be viewed in the total scheme (figure 3.21).

$$Delay = e^{0.09s}\frac{2\pi*8}{s+2\pi*8} = \frac{Y(z)}{U(z)} \tag{3-24}$$

3.7 Mapping from lateral front force to steer angle

In the present study, Fiala tire model is chosen to compute the forces on the tire (section 2.5). Among all forces, the lateral front forces are obtained with respect to the slip angle α_f which is associated with the steer angle, side slip angle, yaw rate and the longitudinal velocity. In control scheme, the desired lateral front forces are computed by LQR. At the same time, all the state variables are estimated. Thus, an inverse mapping is used in order to map the computed lateral forces to slip angles. The desired steer angle is obtained from the relationship between steer angles and slip angles (figure 3.23).

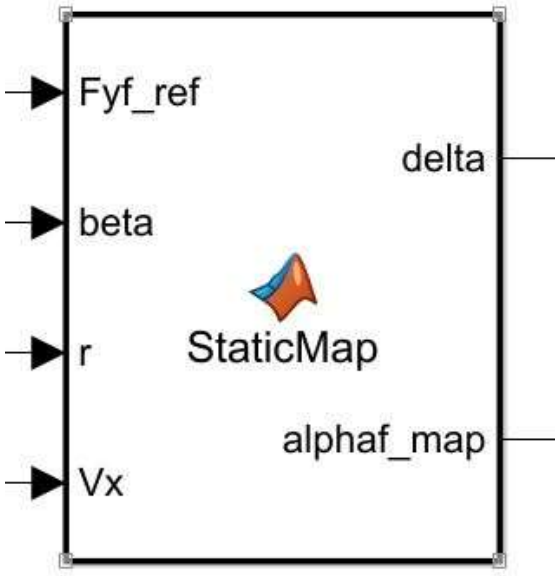

Figure 3. 23. Mapping from lateral forces to steer angles

3.8 Simulation results in Simulink

Below are the initial states for simulation in Simulink.

$$V_{x0} = 0.1[m/s]$$

$$\beta_0 = 0\ [rad]$$

$$r_0 = 0[rad/s]$$

The initial value of longitudinal velocity is set to be a non-zero small value in order to avoid errors in mathematics computation. The other two variables are set to zero. The controller is simulated starting from this state.

The simulation results are shown below (figure 3.24-3.30):

Three state variables are shown in figure 3.24-3.26. The red dashed lines indicate the equilibrium values. It can be viewed in the figures that the behavior of car is similar to the human drivers' behavior (section 3.1).

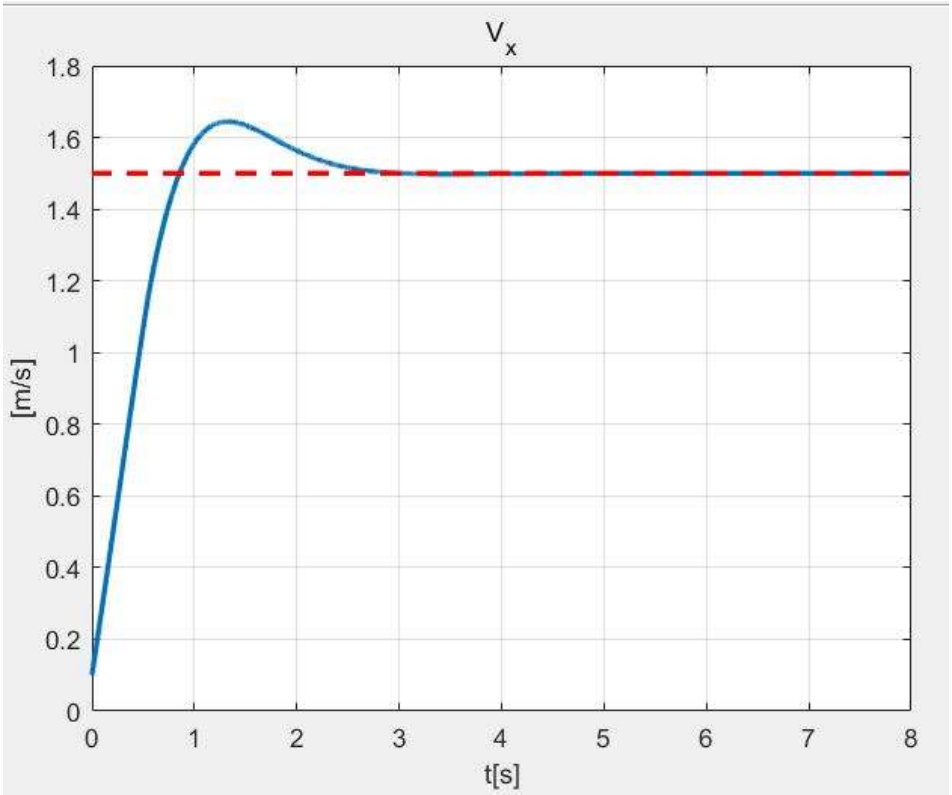

Figure 3. 24. Changes of longitudinal velocity in the function of time

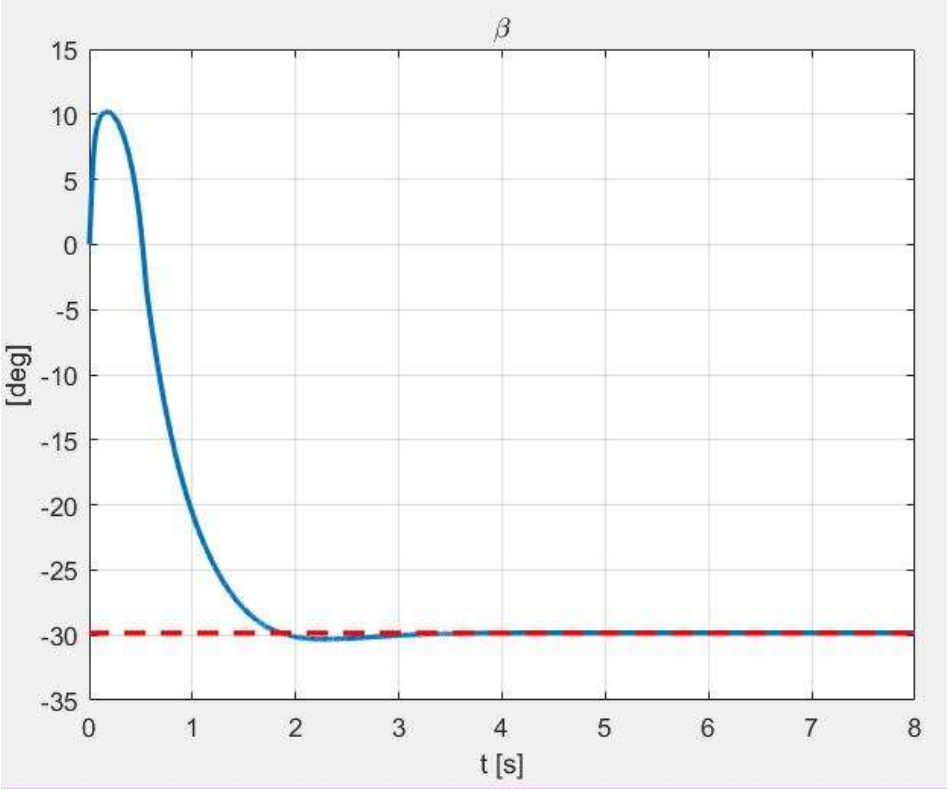

Figure 3. 25. Changes of side slip angle in the function of time

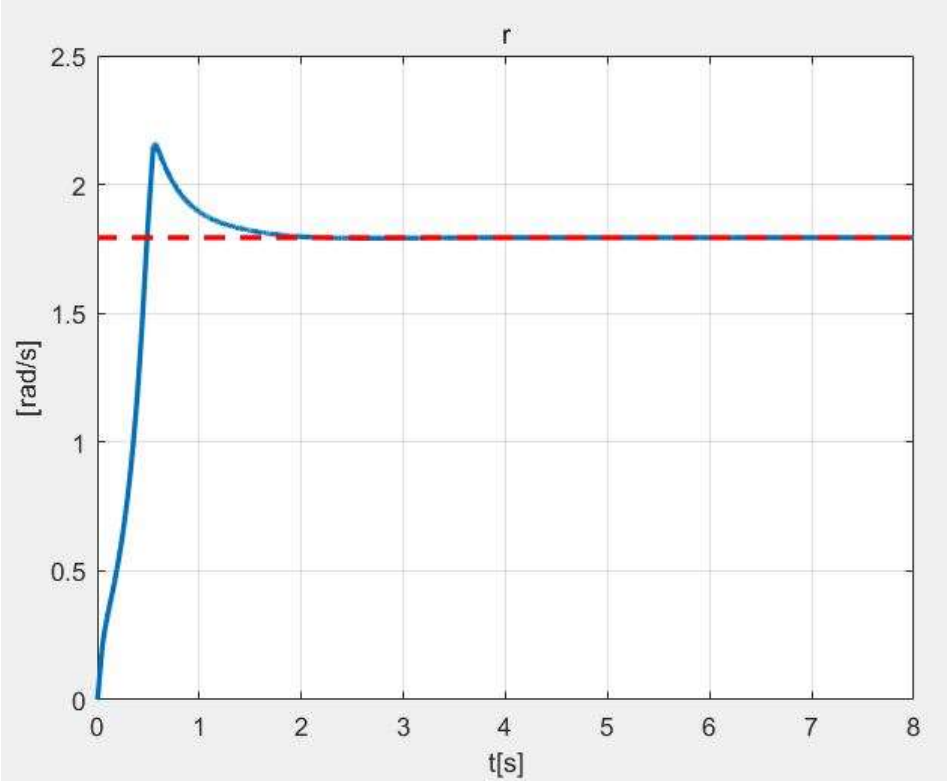

Figure 3. 26. Changes of yaw rate in the function of time

In the time period of 0-1, the car accelerates and it will achieve a value which is larger than the equilibrium. At the same time, the side slip angle has an increase phase at the very beginning but then decrease to a value with larger absolute value. As to the yaw rate, it increases dramatically within a very short time. The decreasing of side slip angle is caused by the large yaw rate of car's body.

From 1s to 3s, the longitudinal velocity decreases and finally converges to the equilibrium. This should be the time period that the stabilization procedure of the controller starts working. In the meanwhile, the yaw rate decreases by counter-steering, and the side slip angle starts to converge.

Two input commands are shown in figure 3.27- 3.28.

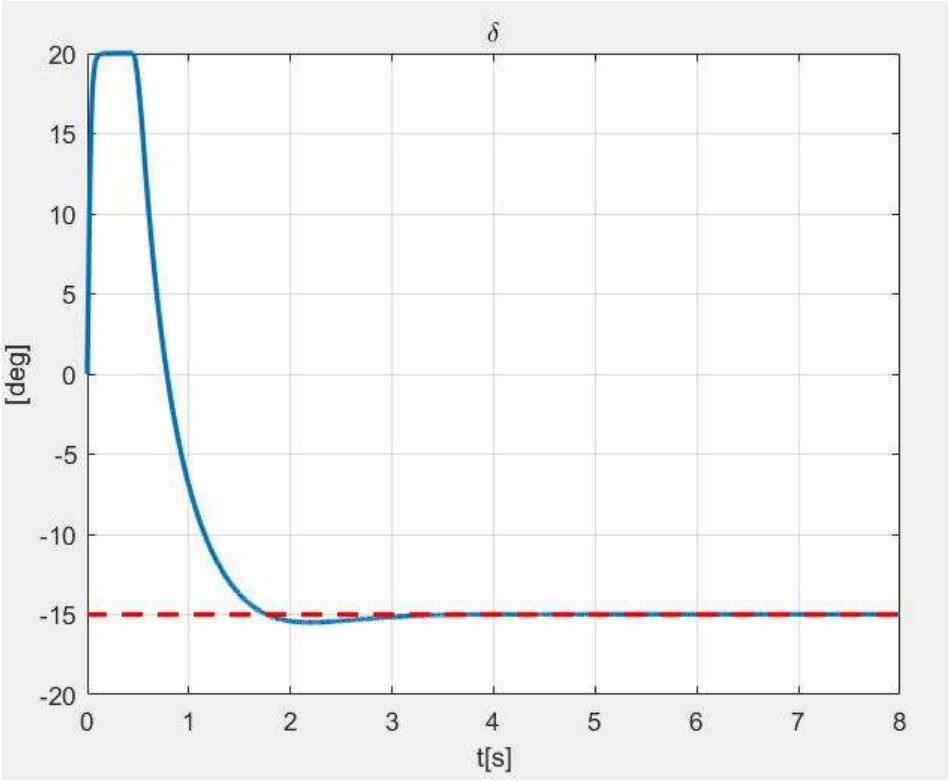

Figure 3. 27. Changes of steer angle in the function of time

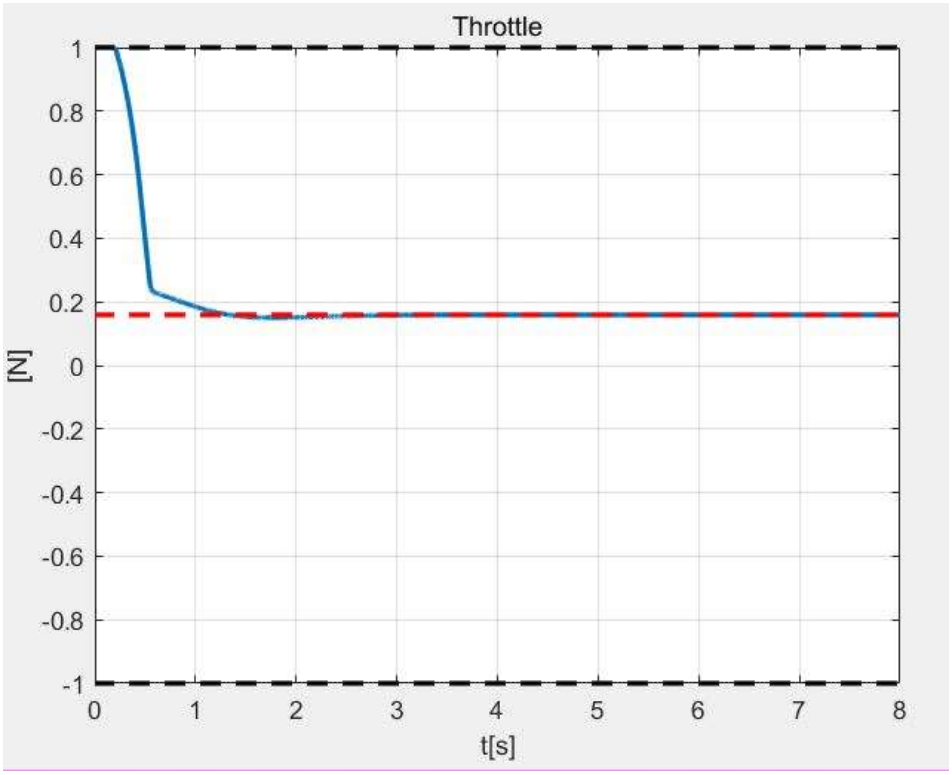

Figure 3. 28. Changes of throttle value in the function of time

At the beginning, the car accelerates for a short time due to the peak of throttle. The peak of throttle makes the rear tire saturated and reduces the capacity of affording force very much. At this moment, the car is over-steering by the steer angle and saturated rear tire. Hence it has a sharp increasing of yaw rate. When the yaw rate increases, the side slip angle goes to a negative value. If considering the lateral velocity at this time, it can be found that the peak of yaw rate makes a dramatic decreasing of V_y in figure 3.29.

At this time, the stabilization process is activated. Negative steer angles are applied to balance the moment of car's body. The throttle is also released to recover a part of capacity of tire. When the car arrives at the equilibrium point, both the throttle and steer commands converges to the equilibrium value so as to maintain the car at the drifting point.

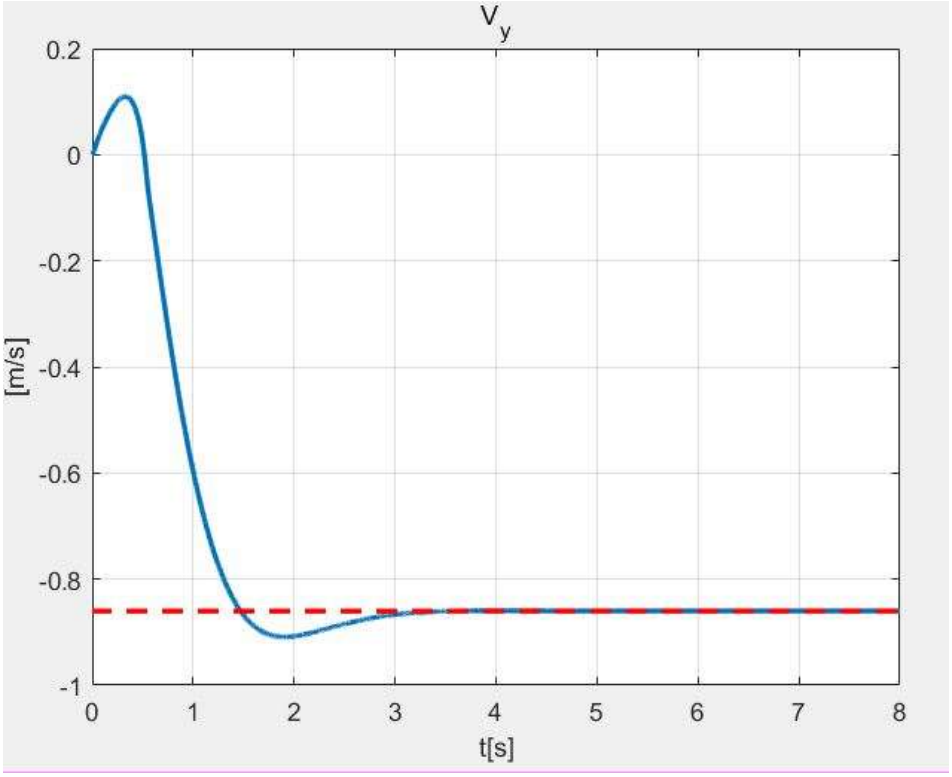

Figure 3. 29. Changes of lateral velocity in the function of time

The trajectory of the car during the whole process is plotted in figure 3.30. A smooth circle trajectory of the car can be viewed in this figure.

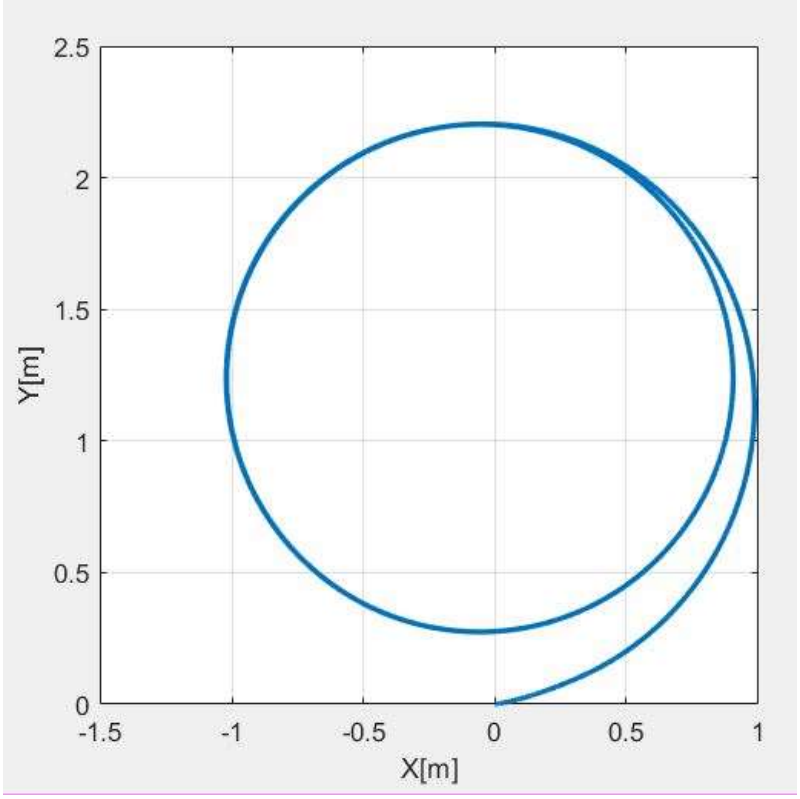

Figure 3. 30. Trajectory of the whole process

The simulation results prove that the feedforward-feedback LQR controller with optimized parameters stabilizes a car around a desired drifting equilibrium even if it starts from a standing still state.

Since the results is under Simulink environment which has many ideal assumptions, further evaluation are needed in Robotic Operation System.

Chapter 4 ROS simulation

The whole scheme mentioned above is simulated in ROS framework in the present chapter. Before starting the simulation, the scheme is divided into several parts. Every part is compiled to ROS node individually. More nodes are also added to complete the system, like the clock, the IMU and the Optitrack Motive. Some tests with simple input commands in ROS are done before simulation in order to validate the generated nodes. In the end of this chapter, the ROS simulation results are evaluated.

4.1 Nodes generating

4.1.1 Environment

The Robotic Operating System (ROS) runs on Linux system. The kinetic version of ROS is applied in the present study. In order not to get into any compatibility problem, the Ubuntu 16LTS is used as the operating system of ROS, which is the most distributed Linux version.

4.1.2 Nodes generating

Nodes are the basic units in ROS, and all the messages are exchanged between nodes. In practice there are two ways to generate ROS nodes[3]. The first one is to write C++ or python code to program. This is the most common way but it requires the skill of programming. The other way is to compile the Simulink model to C++ programs and then build it as a node. This is a quite straightforward way and is presented in this study.

The first step to generate nodes from Simulink model is to set a proper solver for the model. In this study a fixed step Heun solver is chosen with sample time equaling to 1e-03 seconds. As for the simulation time, the ROS cannot recognize the simulation clock directly. Instead, it uses the clock of PC which has large values and is hard to evaluate. So, there should be one command in order to set a simulation time for the system. All the simulation processes should follow this time.

```
<param name="use_sim_time" value="true" />
```

Figure 4. 1. Command of setting simulation time

In Simulink model, the process of generating ROS nodes is in the following steps:

1. Going to Hardware Implementation setting of simulation model and choosing the hardware board as Robotic Operating System (ROS).

2. Configuring the target hardware resources to set the device parameters as the working PC's.
3. Editing the ROS folder and establishing a folder of 'catkin_ws' in Linux which providing a platform for all the nodes, programs and files.

```
sunyao@sunyao-virtual-machine:~/ROS$ catkin_make
```

Figure 4. 2. Command of establishing the catkin folder

4. Going back to Simulink model and compiling the model by 'deploy to hardware'.
5. Generating the ROS node with the compiling file.

```
sunyao@sunyao-virtual-machine:/mnt/hgfs/ros/Simulink Model$ ./build_ros_model.sh
 DriftingController.tgz ~/ROS/catkin_ws
```

Figure 4. 3. Command of generating nodes

4.1.3 Nodes

Several nodes are needed to complete the whole system. A controller to drive and maintain the car in drifting condition; a node of Optitrack Motive to capture the position and orientation of car and estimate the velocity and side slip angle; a node of the IMU to measure the the yaw rate of the car. When the estimation is finished, state variables are published to the controller. A clock to generate simulation time, then, is required. The last but not the least, a node of the vehicle model is necessary for simulation. By the way, a communication node to connect all the hardware parts, like the radio controller and the RC car, is important in the future work of experiments. Below, a list of nodes which are established in this study:

-Controller

Each node has three main parts: a subscriber to obtain input messages from another node, a publisher to deliver output messages for the next node, and a main part to implement input and compute output. In controller node, the side slip angle, yaw rate and velocity of the car are required as the receiving messages. Those three values are obtained by the IMU and Optitrack motive, so they are delivered to the subscriber of controller node. As to the publishing messages which are quite straightforward, the throttle value and steer angle value are transmitted to the vehicle model by the publisher.

The main part is similar to the simulation model. The only difference is about the feedback matrix K. In simulation model, this K is computed by LQR around a chosen equilibrium point. Instead in ROS, the equilibrium point and the feedback matrix are chosen and computed a priori. They are putted in a file named 'Callbacks' associated with the controller node. Whenever this node is run, the file is called and all the values in it are defined at the beginning.

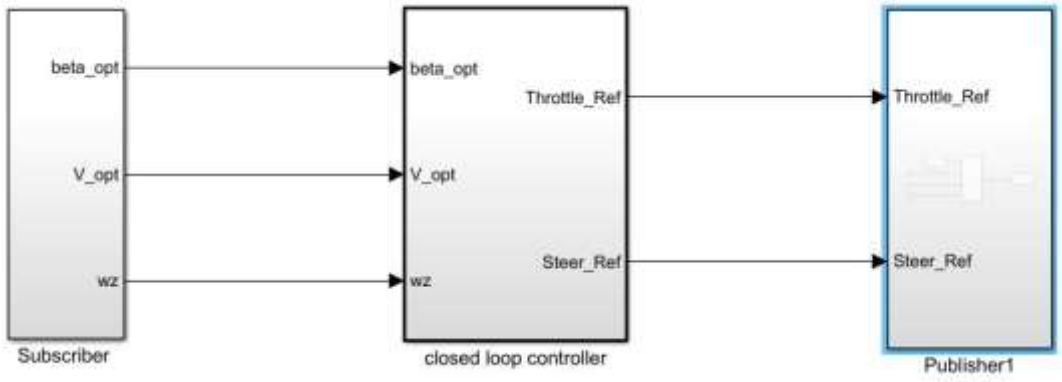

Figure 4. 4. Controller node

-Estimator

Estimator is another node which plays a part of the role of Optitrack Motive. It is provided to compute state variables during car's moving for the controller.

In this node, it receives the position and the orientation of the vehicle model. With the coordinates of the position and the time of clock, the velocity V of the model is computed. However, this is not enough for the controller since the state variable to be regulated is the longitudinal velocity. It still requires finding out the side slip angle, which is also another state variable as well.

The side slip angle is provided with the help of model's orientation. The Optitrack records the orientation of the model with the form of quaternions. After a transformation from quaternions to Euler angle by Simulink, the quaternions are transformed to a form of rotation angles. Combining the yaw angle from rotation angles and the direction of velocity, the side slip angle is found.

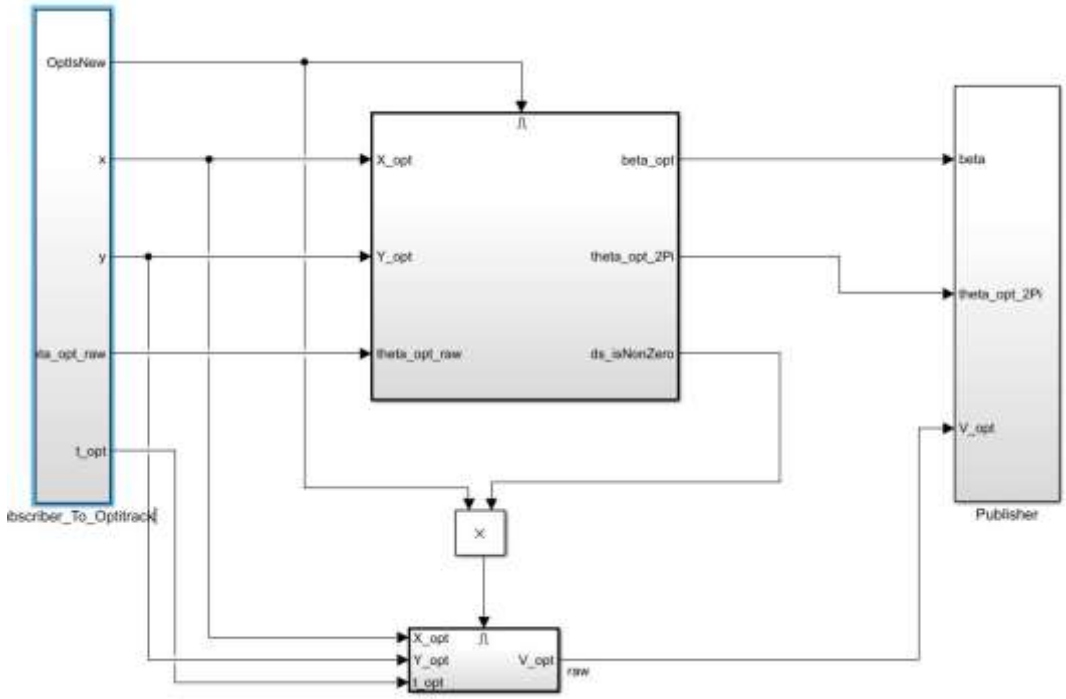

Figure 4. 5. Estimator node

-Clock

The ROS is expected to use simulation time. A clock node to generate time in simulation is required. In Simulink, it is established directly by using a block named Digital Clock with a desired sample time.

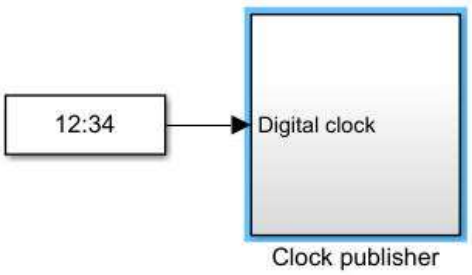

Figure 4. 6. Clock node

-Vehicle model

Generating nodes directly from Simulink loses the information inside the nodes. It is necessary to validate whether the nodes work well before running them in simulation procedure.

Just as the model in chapter 2, the model subscribes commands from controller node. Then it transfers them to the servo commands and regulates them within the proper region. The vehicle model works with the servo commands and generate a series of messages about its positions and orientations. The coordinates of position and the quaternions of orientation are delivered to Optitrack Motive and the IMU.

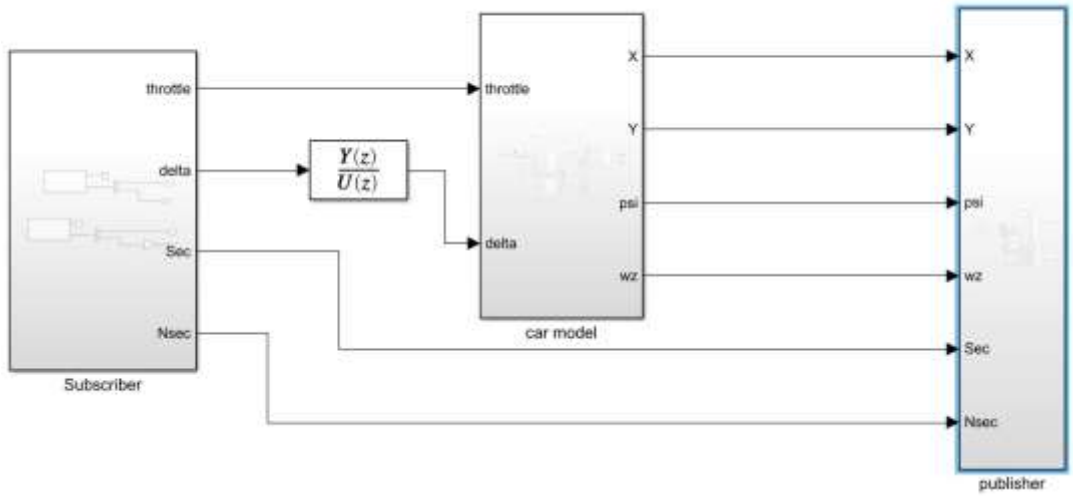

Figure 4. 7. Model node

4.1.4 Launch file

Running all the nodes by typing command line by line is not a clever idea. There is a method named launch file being used to run all the nodes and set required parameters at the same time.

Launch file is a file which includes all the nodes and parameters. In launch file, the clock problem is solved by setting the parameter 'use_sim_time' to be true. With this setting, the simulation is under the simulation clock time instead of the time in the operating system.

There are still some nodes which are not discussed. They are briefly introduced here in order to have an integrated structure of the whole system.

-serial_comm

The node serial_comm is a node for the serial communication among the parts in the RC car. The Odroid-XU4 plays as an on-board computer to manipulate all the data within the system. Ardruino takes the microcontroller role. It exchanges data among the other parts and delivers the commands to the actuators. The node serial_comm node is a bridge for all the data communications in the system.

-myahrs_driver

This node is a driver for the myAHRS+ inertia measurement unit. It publishes all the data measured by the IMU, like the orientation, angular rate, acceleration and magnetic field to Ardruino. The most important thing in this part is setting the port and baud rate to be a proper value.

-drifting_optitrack

The drifting_optitrack is a node to establish the connection between the RC car and Optitrack Motive. Since Motive is used to capture the motion of the car, this node should be set

properly. The most important thing in this node is to guarantee the IP address of Motive and Odroid. Without a correct IP address, the communication between Motive and Odroid will fail.

-rosbag

The rosbag recorder is another important node. It is used to save everything obtained during the simulation, for example the positions and orientations of the car. This node has already been established in a package of ROS named ‘rosbag’. It records the data and saves them to a desired file. It is convenient for a further evaluation of the results.

```
<launch>

<param name="use_sim_time" value="true" />

  <!--- clock_example -->
  <node pkg="clock_example"
        type="clock_example_node"
        respawn="false"
        required="true"
        name="clock_example"
        launch-prefix="">
  </node>

<!-- Run simulation_nonlinear -->
  <node pkg="simulation_nonlinear"
        type="simulation_nonlinear_node"
        name="simulation_nonlinear"
        respawn="false"
        launch-prefix=""
        required="true"
        output="screen">

  </node>

  <!--- Run driftingcontroller -->
  <node pkg="driftingcontroller"
        type="driftingcontroller_node"
        respawn="false"
        name="driftingcontroller"
        launch-prefix="">
  </node>

<!-- Run state_estimator node -->
  <node pkg="state_estimator"
        type="state_estimator_node"
        name="state_estimator"
        respawn="false"
        launch-prefix="">
  </node>
```

```
<arg name="user_home_folder"
        default="/home/sunyao" />
<!-- Run rosbag record -->

  <node pkg="rosbag"
        type="record"
        name="recorder"
        output="screen"
        respawn="false"
        launch-prefix=""
        required="true"
        args="-a -O $(arg user_home_folder)/ROS/bagfile/test2.bag"/>

</launch>
```

Figure 4. 8. An example of launch file

4.1.5 Topics

In ROS, every node manipulates data and achieves their functions by itself. However, if they are used to establish a whole system, they need to communicate with each other and exchange messages. Topic is the unit with a certain structure and is used to exchange messages among nodes.

In ROS, topics are used for unidirectional and streaming communication. They are subscribed and published by different nodes. When the nodes need to receive messages, they require subscribing messages from a topic. On the contrary, when the nodes generate messages, they need to publish messages to a topic.

Every topic has a structure with respect to the message type it carried. Before working in ROS, the matching of type is very important, since a mismatch type message cannot be created by the subscribers. Thus, a file with all the messages' types named 'car_msg' is provided in folder catkin_ws. When a type of topic is required to communicate, it is established directly due to the message type it needs.

Here is a list of some topics which play an indispensable role in the simulation:

-/clock

Topic /clock carries the simulation clock time message. It provides the clock for the whole system during the simluation. The message is composed of two parts, one is the value of second and the other is the value of Nanosecond. When they are added after unifying the unit, the simulation time of clock is obtained.

-/controller_cmd

This topic is the main point of the control scheme. The controller subscribes all the state variables in it and publishes this topic with steer and throttle commands based on the control law. Just as what the radio controller does, this topic is composed of three channels of data. The first and second channels are steer command and throttle command messages. In order to distinguish the automatic command and manual command which occurs in the real RC car, it requires a tag message named 'state'. When Ardruino receives this topic with state equaling to 2, it regards the topic as the autonomous controller command and send them to actuators. In the

simulation process, the state message is always set to be 2.

-/state_estimator_opt_V & /state_estimator_opt_beta

Those two topics carry the estimation results of the velocity and side slip angle of the model. They are from the state estimator node and are the main reference data for the controller.

-/imu/data

This topic has all the variables measured by the IMU. It contains the orientation of the model and the angular rate around three axes with quaternion form. The yaw rate is obtained from this topic after transforming the quaternions to Euler angles.

-/car/pose

This topic comes from Optitrack Motive. In this topic, all the position and orientation messages of the model are found. This topic and topic /imu/data are two important topics for estimating state variables.

4.2 Validation of ROS nodes

4.2.1 Controller of simple inputs

The validation procedure of ROS nodes can be achieved by applying simple inputs as the steer angle and throttle value to the model node (figure 4.9). The simple inputs have only two parts. The first one is a simple acceleration command where the throttle is high and the steer angle is zero to see if the throttle command is achieved properly. The other one has a sinusoid steer angle command and low throttle, which can test whether the steer input works well (figure 4.10-4.11).

Since a fidelity model is required, the servo delay of car is considered in the model. This delay is between the desired steer command and the servo input. From some identifications, this delay is simulated with a low pass filter which has a cut-off frequency of 8 Hz, and a time delay with 0.09 second.

In this test, the sample time of digital clock is set to be $1e^{-03}$ second as it is in the Simulink model. As a reference, the model in Simulink is also modified to make it consistent with the validation procedure. Two results in ROS and Simulink are compared and seen if the ROS nodes work well.

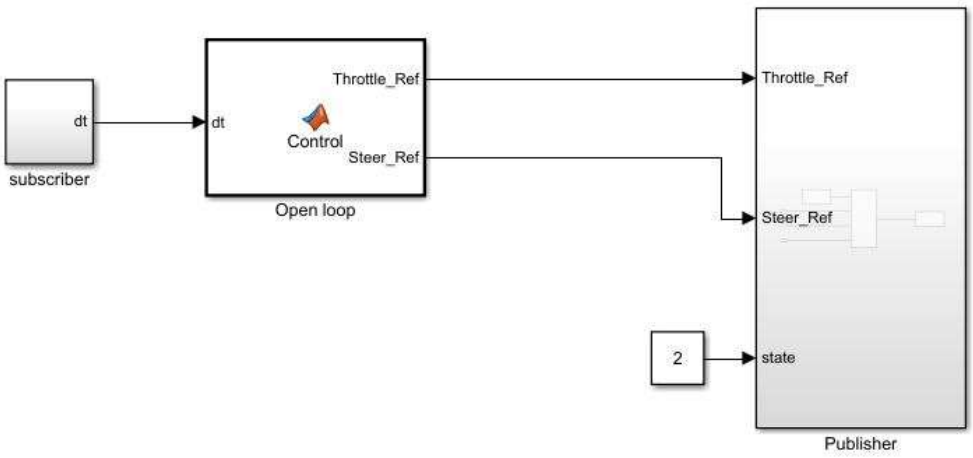

Figure 4. 9. Controller of validation procedure

```
function [Throttle_Ref, Steer_Ref] = Control(dt)

if (dt>=1 && dt<2)
    Throttle_Ref=0.8;
    Steer_Ref=0;
else
     if (dt>=2)
        Throttle_Ref=0.01;
        Steer_Ref=10*pi/180*sin(0.5*pi*(dt-2));
     else
        Throttle_Ref=0;
        Steer_Ref=0;
    end
end
return;
```

Figure 4. 10. Control commands

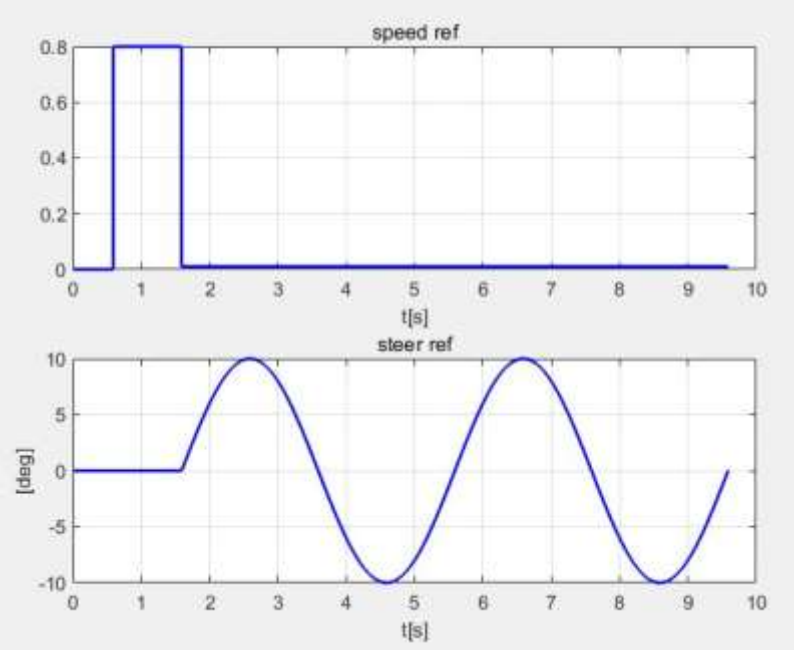

Figure 4. 11. Simple inputs in the function of time

4.2.2 Validation results

In order to have a total and detailed comparison, the ROS results and Simulink results are compared in two aspects. One is about the trajectory and the other is about the state variables, which are the longitudinal velocity, side slip angle and yaw rate.

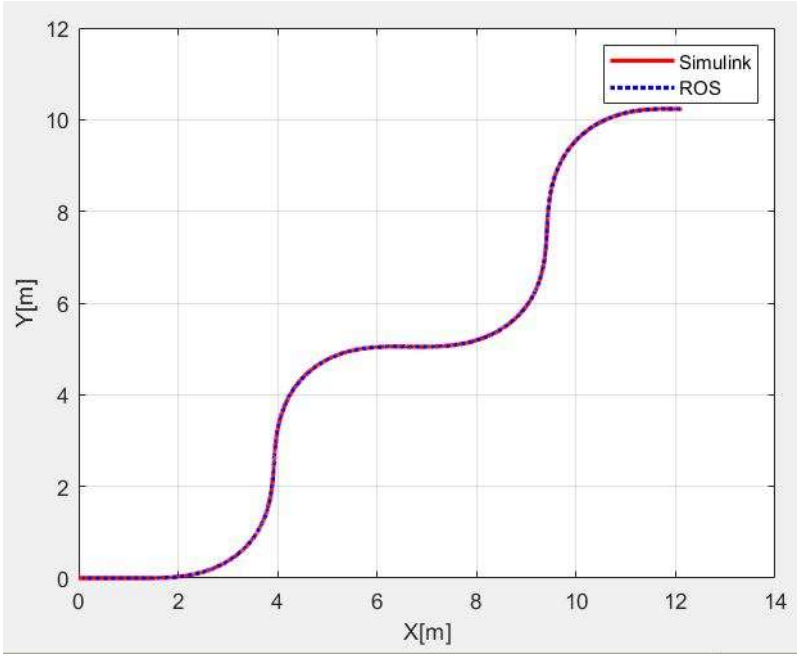

Figure 4. 12. Trajectories of validation procedure

The figure 4.12 is about the trajectories of ROS and Simulink model within 10 seconds.

In figure 4.12, two trajectories are consistent with each other perfectly. Thus, in terms of the trajectory, the ROS nodes work properly as the Simulink model does.

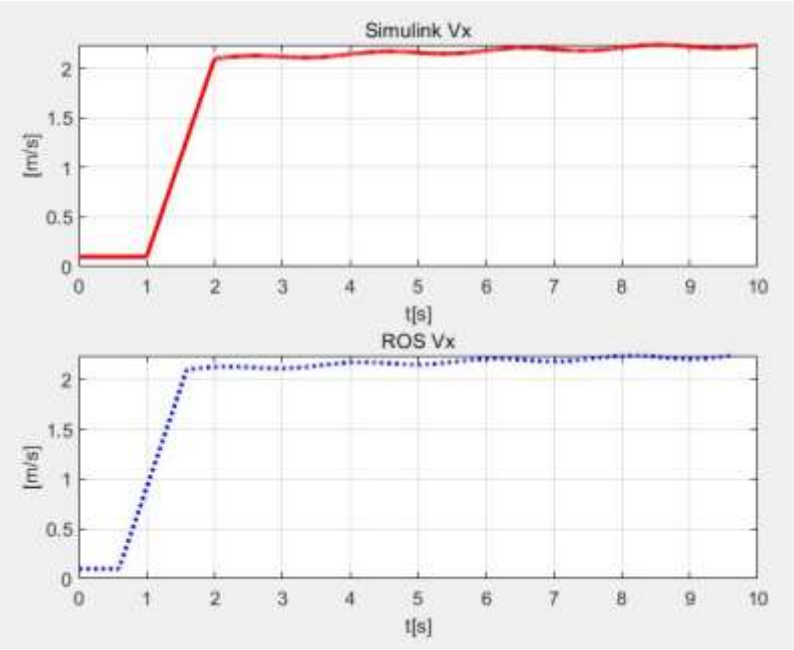

Figure 4. 13. Longitudinal velocities

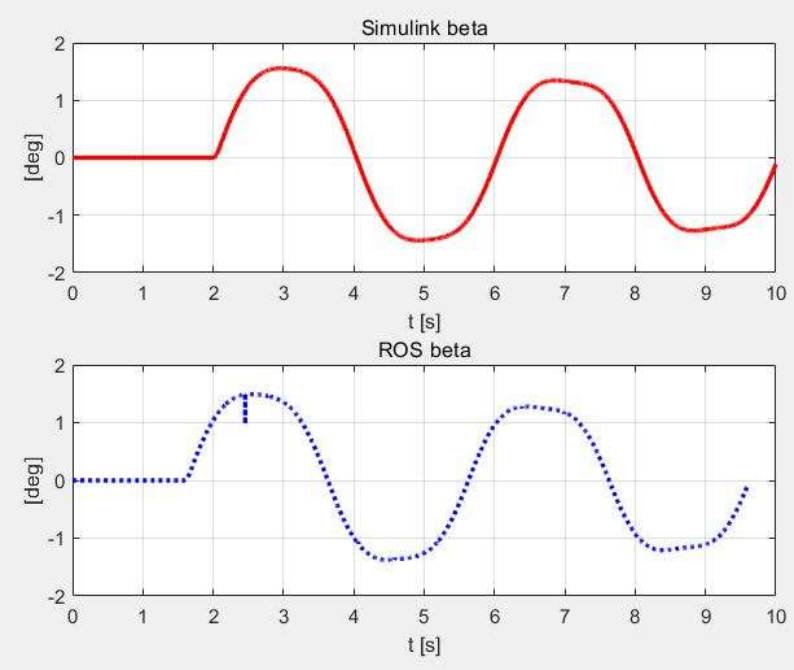

Figure 4. 14. Side slip angles

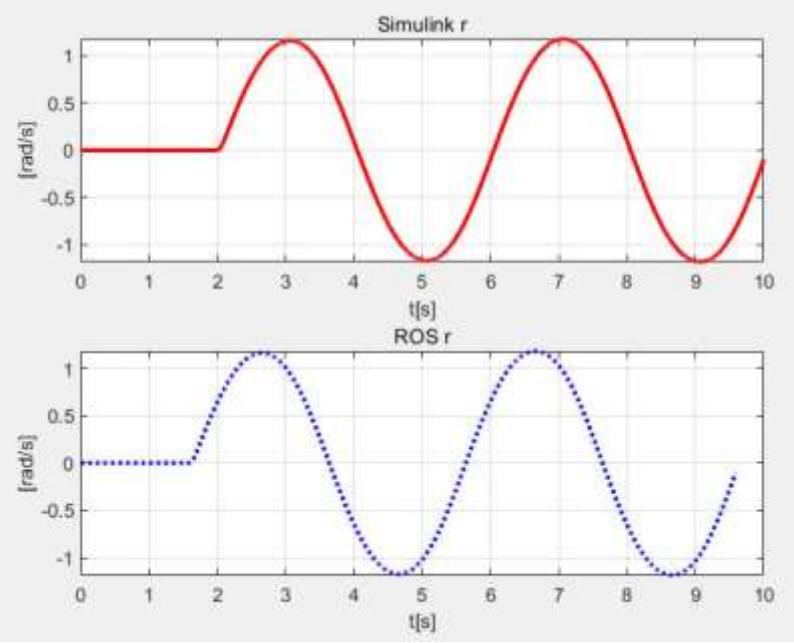

Figure 4. 15. Yaw rates

Those three figures (figure 4.13- 4.15) are the comparison of longitudinal velocity, side slip angle and yaw rate. Like the trajectories, the ROS results and the Simulink results are almost the same only except a time shift at the beginning. This time shift always exists because the rosbag file recorder does not work simultaneously at the beginning of clock. In this test, the time shift is 0.4115s. If this shift is taken into account in the ROS result, two results are believed to be consistent. Therefore, the generated ROS nodes can work as well as the Simulink model does.

4.3 Simulation of LQR controller in ROS

After the validation with simple input commands, the controller based on ROS is simulated. In this simulation, the whole system is a closed loop. The work to collect the angular rates and coordinates of model of the IMU and Optitrack are simulated. The node 'state estimator' is used to compute the velocity and side slip angle during the process, and the topic /imu/data is used to record the yaw rate of the model. Those three variables are published to the drifting controller in order to compute the control commands for the servo of the model

Here are the results of the simulation process and some evaluations about the performance of the system. In order to avoid losing information at the beginning which results from the time shift of ROS, a very small time period is left before activating the system.

Among three state variables, the most direct one is the yaw rate r. This state is record in the topic /imu/data. In fact, this topic is composed of not only the angular rate, but the orientation and acceleration as well. In this simulation, the yaw rate is mainly considered.

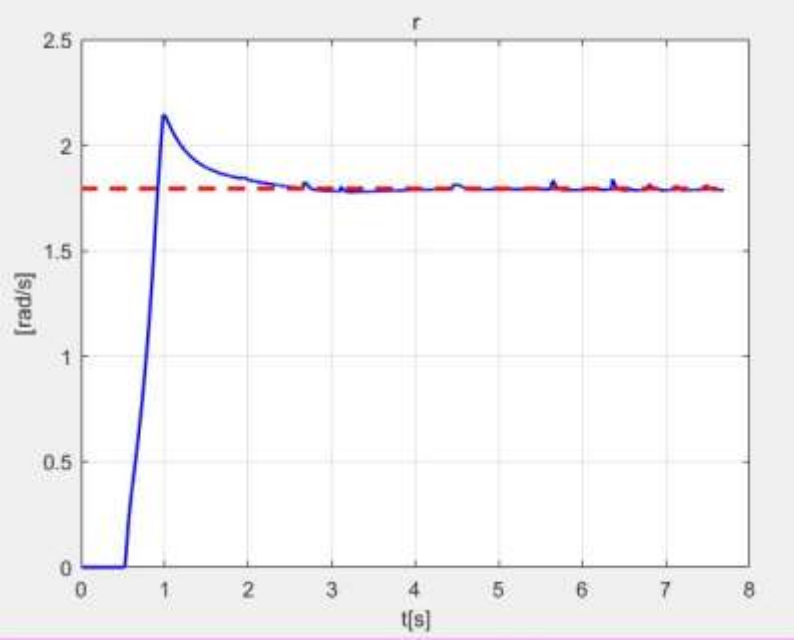

Figure 4. 16. Yaw rate result in ROS simulation

The trend of yaw rate is similar to the MatLab results. After the small time period at the beginning, it increases dramatically and then moves to a steady state value with the help of LQR. It can be noticed that there are some disturbances in the results. The disturbances may come from signal transferring among the nodes.

As for the side slip angle, it cannot be measured directly by sensors. In this simulation it is provided by the topic /car/pose and the state estimator node. The side slip angle is defined as the angle between the velocity of COG and the orientation of the car's body, so it can be derived with the help of the car's position.

$$\operatorname{atan}\left(\frac{v}{u}\right) - \theta = \beta \tag{4-1}$$

Here v and u are the velocity along the y-axis and x-axis of the world frame reference, θ is the angle that the center axis of car rotating from the x-axis of the world frame.

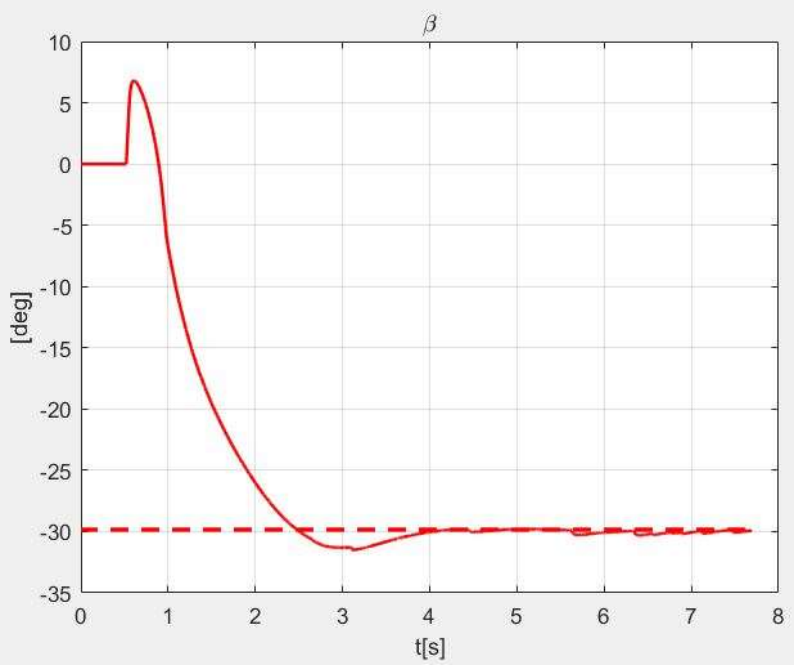

Figure 4. 17. Side slip angle in ROS simulation

The longitudinal velocity is computing by the resultant velocity and side slip angle. When the topic /car/pose collects the coordinates of positions in the world frame, the state estimator receives them and does differential on the positions both on the x axis and y axis. Then, the velocity v and u of the world frame are obtained. The resultant velocity is computed by a simple superposition. Since the side slip angle β is obtained by equation (4-1), the longitudinal velocity is provided by a product between resultant velocity and cosine β.

In figure 4.18, we can notice that there are several points far from the main trend and seems like an impulse. They are produced by the differential computation in the state estimator node.

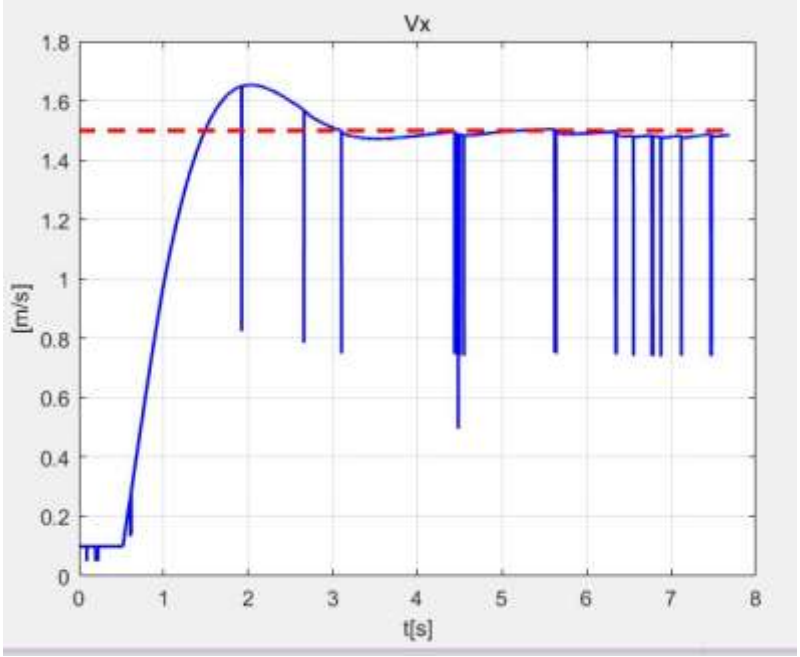

Figure 4. 18. Longitudinal velocity in ROS simulation

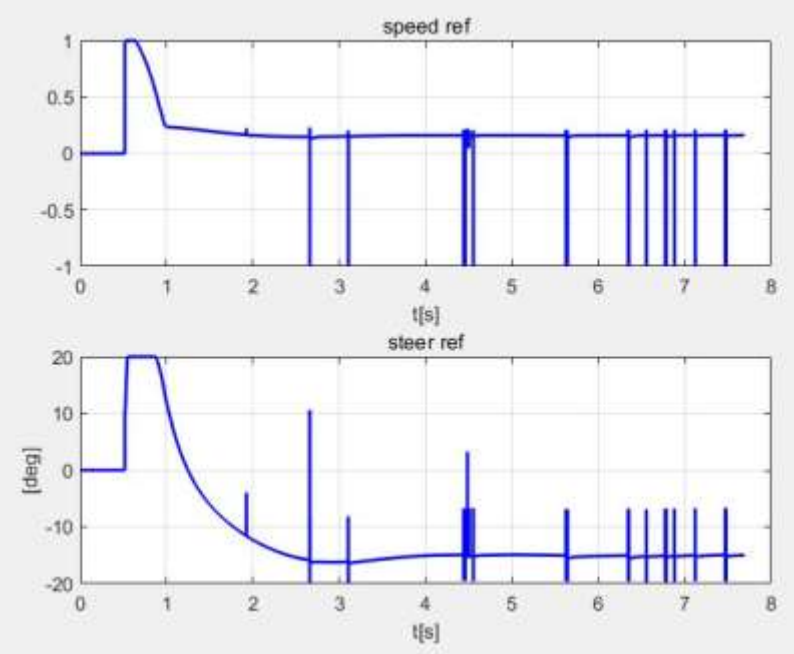

Figure 4. 19. Controller commands in ROS simulation

In figure 4.19 of the controller commands in this simulation, the throttle and steer commands in the topic /controller_cmd are shown. As the behavior in Simulink results, the throttle reaches a peak value at the beginning to destabilize the model from the stable equilibrium. The large value of steer angles also has contribution on this process. When the large yaw rate leads a negative side slip angle, the steer command is reduced and becomes a counter-steering command. In the meanwhile, the throttle is also reduced to recover part of capacity of the rear wheel. During this process, the longitudinal velocity is increasing with

respect to the throttle. When the model is near the desired drifting equilibrium, the throttle and steer commands reach the equilibrium values to maintain the model under the desired equilibrium. After the yaw rate and side slip angle finish their stabilization, the longitudinal velocity reaches its equilibrium value. The simulation results present that the LQR controller with a feedforward-feedback control scheme succeeds to stabilize an RWD vehicle model to a chosen drifting equilibrium.

Conclusion and future work

In the present study, an LQR drifting controller with a feedforward-feedback scheme has been designed for an RWD RC car to stabilize it to a drifting equilibrium. The controller is developed based on a single track model and is simulated in Simulink and ROS framework.

This controller provides throttle and steer commands and regulates the longitudinal velocity, side slip angle and yaw rate of the model to a chosen drifting equilibrium. The LQR control strategy is applied to a linearized system around the drifting equilibrium. A feedforward-feedback control scheme is applied to the LQR controller in order to achieve the control target for the non-linear single track model. By evaluating the simulation results in Simulink, the parameters of LQR are carefully optimized for a proper performance of the system from an initial state with standing still. All the designs are in Simulink and are compiled to ROS with the function of MatLab. Thus, a validation procedure in ROS is proposed to evaluate the consistence between the Simulink model and the generated ROS nodes. This evaluation procedure applies a series of simple input commands to the model and compares the results of the Simulink model and the ROS nodes. After validation, the system in ROS is modified by adding a few nodes and blocks. The state variables of the system are estimated in the new nodes as if in the real hardware. The control commands are also considered with a delay block of servo as if in the real car to increase the fidelity of the simulation. The simulation in ROS executes the developed system with the time of simulation clock. The variables to be evaluated from the simulation are recorded in a rosbag file. The final simulation results prove that the LQR controller with feedforward-feedback control scheme can stabilize the model from a standing still condition by the optimized LQR parameters.

In the future, the system will be applied for different conditions of tracks, and developed by taking into consideration of the aerodynamic drag force. The model can also be extended to be a more complex one for a deeper study and an experiment of a real RC car.

Bibliography

[1]. Hindiyeh, R.Y. and J.C. Gerdes, *A controller framework for autonomous drifting: Design, stability, and experimental validation.* Journal of Dynamic Systems, Measurement, and Control, 2014. **136**(5): p. 051015.

[2]. Voser, C., R.Y. Hindiyeh, and J.C. Gerdes, *Analysis and control of high sideslip manoeuvres.* Vehicle System Dynamics, 2010. **48**(1): p. 317-336.

[3]. *ROS tutorial*. Available from: http://wiki.ros.org/ROS/Tutorials.

[4]. Quigley, M., et al. *ROS: an open-source Robot Operating System*. in *ICRA workshop on open source software*. 2009. Kobe, Japan.

[5]. Abdulrahim, M., *On the Dynamics of Automobile Drifting*. 2006, SAE International.

[6]. Chakraborty, I., P. Tsiotras, and R.S. Diaz. *Time-optimal vehicle posture control to mitigate unavoidable collisions using conventional control inputs*. in *2013 American control conference*. 2013. IEEE.

[7]. Chakraborty, I., P. Tsiotras, and J. Lu. *Vehicle posture control through aggressive maneuvering for mitigation of T-bone collisions*. in *2011 50th IEEE Conference on Decision and Control and European Control Conference*. 2011. IEEE.

[8]. Velenis, E. and P. Tsiotras. *Minimum time vs maximum exit velocity path optimization during cornering*. in *2005 IEEE international symposium on industrial electronics*. 2005. Citeseer.

[9]. Clover, C. and J. Bernard, *Longitudinal tire dynamics.* Vehicle System Dynamics, 1998. **29**(4): p. 231-260.

[10]. Beckman, B., *The physics of racing.* Burbank, CA, 1991. **91503**: p. 1991-2002.

[11]. Velenis, E., et al., *Steady-state drifting stabilization of RWD vehicles.* Control Engineering Practice, 2011. **19**(11): p. 1363-1376.

[12]. Velenis, E. *FWD vehicle drifting control: The handbrake-cornering technique*. in *2011 50th IEEE Conference on Decision and Control and European Control Conference*. 2011. IEEE.

[13]. Velenis, E., E. Frazzoli, and P. Tsiotras. *On steady-state cornering equilibria for wheeled vehicles with drift*. in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*. 2009. IEEE.

[14]. Velenis, E., E. Frazzoli, and P. Tsiotras, *Steady-state cornering equilibria and stabilisation for a vehicle during extreme operating conditions.* International Journal of Vehicle Autonomous Systems, 2010. **8**(2-4): p. 217-241.

[15]. *Odroid*. Available from: https://www.hardkernel.com/.

[16]. *Ardruino*. Available from: https://www.sintron.co.uk/.

[17]. *myAHRS+*. Available from: http://withrobot.com/en/sensor/myahrsplus/.

[18]. *OptiTrack Motive*. Available from: https://optitrack.com/products/motive/.

[19]. *Sakura D4 RWD 1/10 Drift car*. Available from: https://hobbyking.com/en_us/3racing-sakura-d4-rwd-1-10-drift-car-kit.html.

[20]. Hac, A. and M.D. Simpson, *Estimation of Vehicle Side Slip Angle and Yaw Rate*. 2000, SAE International.

[21]. Chindamo, D., B. Lenzo, and M. Gadola, *On the Vehicle Sideslip Angle Estimation: A Literature Review of Methods, Models, and Innovations*. Vol. 8. 2018. 355.

[22]. Grip, H.F., et al., *Vehicle sideslip estimation.* IEEE control systems magazine, 2009. **29**(5): p. 36-52.

[23]. Burhaumudin, M.S., et al. *Modeling and validation of magic formula tire model*. in *International Conference on the automotive industry, mechanical and materials science (ICAMME'2012), Penang*. 2012.

[24]. Pacejka, H., *Tire and vehicle dynamics*. 2005: Elsevier.

[25]. *Linear Control System*. Available from: https://www.academia.edu/27993639/Linear_Control_Systems_Lecture_14_Linear_Quadratic_Regulator_LQR.

[26]. Gonzales, J., et al. *Autonomous drifting with onboard sensors*. in *Proc. 13th Int. Symp. Advanced Vehicle Control (AVEC'16)*. 2016.

Acknowledgements

First of all, I would like to express my deepest gratitude to Professor Luca Bascetta for giving me the opportunity to do this study. His simulating suggestions and encouragement helped me during all the time of this study.

Then I would like to thank Marco Baur, who kindly shared his knowledge and experience to me. I would not make such progress without his patient guidance and valuable advice.

My thanks also go to all my colleagues in AIRLab. Their friendship and company encouraged me to solve every problem I met in the project.

I finally thank my beloved parents who have always believed in me and support me out of difficulties in my life.