NTNU
Norwegian University of
Science and Technology

Trajectory Following for Formula Student Driverless Vehicle

For Revolve NTNU

Sondre Næss Midtskogen

Master of Science in Industrial Cybernetics
Submission date: August 2018
Supervisor: Marta Maria Cabrera Molinas, ITK

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Acknowledgements

I would like to thank all the members of Revolve NTNU Team 2018, in particular the members of the Driverless project. It has been a pleasure to get to know you all, and to work with such a dedicated and skilled group of students.

Thanks to my supervisor, Professor Marta Molinas, and to the Faculty of Information Technology and Electrical Engineering at NTNU for supporting my master's project.

Finally, a thank you to Ingvild, who moved to Alta, which allowed me to spend a ridiculous amount of time working on the Revolve NTNU Driverless project.

Abstract

This masters thesis is written as part of a position in Revolve NTNU Team 2018, as part of the Driverless team. The authors area of responsibility in the project is Trajectory Following. The main objective is to design, implement and test a motion control system for the driverless vehicle, along with any peripherals required to interface with neighbouring systems. Additionally it is the authors responsibility to maintain and adapt the existing driver assistance control systems and the accompanying state estimations, in the transition to a driverless vehicle.

A simple minimum viable prototype system was implemented, consisting of a Line-Of-Sight guidance controller for lateral control and a PI feedback controller with a quadratic drag feedforward term for longitudinal control. This system was very robust at low speeds.

In addition, a high-performance system was implemented in software, but was never tested on the vehicle because the upstream modules never progressed to the point where this was a sensible course of action. This system consists of a nonlinear feedback linearization lateral controller, and a PI longitudinal controller with feedforward terms for aerodynamic drag and reference speed gradient. The interface to the vehicles tractive system control system was augmented to directly accept a reference yaw rate and a longitudinal force request, which are produced by the lateral and longitudinal controllers, respectively. A longitudinal force controller was implemented to handle the latter.

A parameter estimation algorithm for the Magic Formula tire model, using the gradient search method, was implemented in order to estimate the longitudinal and lateral tire-road forces and coefficients of friction, for use in the tractive system control system. This showed good results in simulations, but requires good estimates for the tire slip ratios, which are not readily available on the physical vehicle.

Everything presented in this thesis is the author's own work, unless stated otherwise. No other party was direct involved in the development. The relevant resources at disposal for the project was software licenses for MATLAB and IPG CarMaker, the tractive system control systems already implemented by Revolve NTNU students, including the modelling of Eld and the track used for simulations, in addition to the RC-car and the Gazebo simulation.

Sammendrag

Denne oppgaven er skrevet som en del av en stilling i Revolve NTNU Team 2018, som en del av prosjektet som omhandler den førerløse bilen. Forfatterens ansvarsområde i dette prosjektet er banefølging (Trajectory Following). Hovedoppgaven er å designe, implementere og teste et bevegelseskontrollsystem for den førerløse bilen, sammen med alle systemer nødvendig for å samhandle med nørliggende systemer. Det er i tillegg forfatterens ansvar å vedlikeholde og tilpasse de eksisterende førerassistansereguleringsystemene, og de medfølgende tilstandsestimeringene, i overgangen til en førerløs bil.

En ”enkleste fungerende prototype” ble implementert, bestående av en ”Line-Of-Sight Guidance controller” for lateral regulering, og en PI tilbakekoblet regulator med en kvadratisk luftmotstandsforoverkobling for hastighetsregulering. Dette systemet var veldig robust i lave hastigheter.

Et høyytelsessystem ble også implementert i software, men ble aldri implementert på bil ettersom moduler tidligere i programflyten aldri nådde det nivå at dette var fornuftig å bruke tid på. Dette systemet består av en ”Feedback linearization controller” for lateral kontroll, og en PI tilbakekoblet regulator for hastighetsregulering, med en foroverkoblinger for luftmotstand og gradienten i hastighetprofilen. Grensesnittet til bilens førerassistansereguleringsystemer ble utvidet til å direkte motta en ønsket rotasjonshastighet om bilens vertikalakse, samt en ønsket longitudinal kraft. Dette sendes av den laterale og den longitudinale regulatoren, respektivt. En regulator for longitudinal kraft er også implementert, for å behandle signalet.

En parameterestimeringsalgoritme for dekkmodellen ”Magic Formula”, ved bruk av gradientsøkmetoden, er implementert for å estimere laterale og longitudinale krefter og friksjonskoeffisienter, til bruk i bilens førerassistansereguleringsystemer. Denne algoritmen viste gode resultater i simulering, men krever gode estimater for dekkenes ”slip ratio”, som ikke er direkte tilgjengelig på den fysiske bilen.

Alt som presenteres i denne oppgaven er forfatterens eget arbeid, om ikke annet blir sagt. Ingen andre har vært delaktige i utviklingen. De relevante ressursene tilgjengelig for prosjektet er softwarelisenser for MATLAB og IPG CarMaker, førerassistansereguleringsystemene allerede implementert av tidligere medlemmer av Revolve NTNU, inkludert modelleringen av Eld og banen brukt til simuleringer, i tillegg til RC-bilen og Gazebosimuleringen.

Table of Contents

List of Figures

Nomenclature

α	Sideslip angle
β	Sideslip angle
χ	Course
Δ	Look-ahead distance
δ	Mean wheel steering angle
Δ_s	Spring deflection
$\dot{variable}$	variable differentiated with respect to time
γ	Slip ratio
$\hat{variable}$	Estimated variable
κ	Curvature
μ	Coefficient of friction
ω	Rotational velocity
ϕ	Roll angle
ψ	Yaw angle
ρ	Density of air
σ	Relaxation distance
τ	Torque
$\tilde{variable}$	variable error: $variable_{ref}$ - variable
A	Aerodynamic impact area

a	Acceleration
B	Magic formula constant
C	Magic formula constant
C_d	Drag coefficient
C_f	Front tire cornering stiffness for single-track model
C_r	Rear tire cornering stiffness for single-track model
D	Magic formula constant
F	Tire force
F_red	Torque reduction factor
$F_{z,0}$	Steady-state normal load
g	Gravitational acceleration
GR	Gear Ratio
I	Car rotational inertia
J_ω	Wheel rotational inertia
K	Constant
K_s	Spring stiffness
K_v	Under-steer coefficient
K_{ARB}	ARB stiffness
l	Wheelbase (distance between front and rear axles
l_f	Distance from center of gravity to front axle
l_r	Distance from center of gravity to rear axle
L_w	Distance from car symmetry line to tire (i.e. roll moment arm)
M	Corrective yaw moment produced by Torque Vectoring
P	Power consumption
r	yaw$_rate$
R_{eff}	Effective wheel radius
s	Slip - either slip angle or slip ratio
T	Sum of wheel torques

TPP_{eq}	Throttle Pedal Position equivalent
U	Speed
u	Velocity in vehicle longitudinal (x) direction
v	Velocity in vehicle lateral (y) direction
y	Lateral displacement
^{FL}	Superscript: In the reference frame of - and pertaining to - Front Left tire
^{FR}	Superscript: In the reference frame of - and pertaining to - Front Right tire
^{ij}	Superscript: In the reference frame of - and pertaining to - A given tire
^{RL}	Superscript: In the reference frame of - and pertaining to - Rear Left tire
^{RR}	Superscript: In the reference frame of - and pertaining to - Rear Right tire
^{sf}	In the Seret-Frenet frame [1, 2]
$_{x}$	Subscript: In the x- direction (the driving direction)
$_{y}$	Subscript: In the y- direction (orhogonally left of the driving direction)
$_{z}$	Subscript: In the z- direction (the vertical direction)
$_{acc}$	Subscript: When accelerating
$_{B}$	Subscript: Anti-windup
$_{dec}$	Subscript: When decelerating
$_{D}$	Subscript: Differential
$_{FF}$	Subscript: Feedforward
$_{I}$	Subscript: Integral
$_{k}$	At iteration k
$_{lim}$	Subscript: Limit
$_{m}$	Subscript: Pertaining to Magic formula
$_{PL}$	Subscript: Power Limiter
$_{P}$	Subscript: Proportional
$_{r}$	Subscript: Reference/target value
$_{s-t}$	Subscript: Single-track vehicle model
$_{TC}$	Subscript: Traction Controller
τ_{red}	Torque reduction

Abbreviations

CV	Combustion Vehicle
EV	Electric Vehicle
DV	Driverless Vehicle
FS	Formula Student
FSG	Formula Student Germany
KERS	Kinetic Energy Recovery System (regenerative braking)
ARB	Anti Roll Bar
ROS	Robotic Operating System
TSCS	Tractive System Control Systems
HLC	High-Level Controller
MPCC	Model Predictive Contouring Control
MVP	Minimum Viable Prototype
MVPHLC	Minimum Viable Prototype High-Level Controller
HPHLC	High-performance High-Level Controller
HIL	Hardware-In-the-Loop
VCU	Vehicle Control Unit
TV	Torque Vectoring
LOS	Line-Of-Sight
SR	Slip Ratio
SA	Slip Angle
SLAM	Simultaneous Localization And Mapping
EBS	Emergency Brake System
INS	Inertial Navigation System
QP	Quadratic PRogramming

Chapter 1

Introduction

This master's thesis is written as part of a position in Revolve NTNU Team 2018, as part of the Driverless team. Revolve NTNU is Trondheim's Formula Student team, which has participated in Formula Student competitions every year since 2012. This year, in addition to producing a new electric race car, Revolve NTNU will re-purpose last year's electric race car (Eld) to compete in the Driverless Vehicle class, which was introduced in the 2017 season at Formula Student Germany.

The driverless team will participate in competitions in Hungary and Germany at the end of July and at the beginning of August. The team's primary goal is to complete all the dynamic events at both competitions. Last year, only six out of fifteen teams passed the technical inspection required to participate in the dynamic events, and of those, only one team finished all three events.

The team's secondary goal is to place top five overall, which will require a high-performing autonomous driving system, as well as good performance in the static events.

1.1 Problem definition

The author's area of responsibility in the project is *Trajectory Following*. The main objective is to design, implement and test a motion control system for the driverless vehicle, along with any peripherals required to interface with neighbouring systems. Additionally it is the author's responsibility to maintain and adapt the existing driver assistance control systems and the accompanying state estimations, in the transition to a driverless vehicle.

1.2 Thesis structure

The thesis is structured as follows: In chapter 2, the background for the thesis is presented. This includes an introduction to Formula Student Driverless, to the base vehicle, Eld, including some pre-existing control systems. Finally, the Driverless Team 2018 project plan is presented, and this thesis' contribution to the project is defined. In chapter 3, the thesis objectives are presented. In chapter 4, the development methodology is described, and the mathematical modelling for each system implemented is presented. In chapter 5, the results from simulations are presented and discussed, along with data from the trackdrive event at Formula student Germany. In chapter 6, the thesis is concluded, and further work is suggested.

Chapter 2

Background

This thesis is a contribution to the Revolve NTNU Driverless project for the 2018 Formula Student season. The basis for the project is the Formula Student competition goals and rules [3], as well as the base vehicle, Eld, with all its hardware and software. Most notably the tractive system control systems, which fall under the trajectory following module.

The main goal of this master's project is to contribute towards the main goals of the NTNU Driverless project. This sets a strict precedence for how the master's project is planned and executed. Tasks which are essential for the progression of the NTNU Driverless project, but are not of academic interest, have been prioritized over tasks that are purely of academic interest, but which does not contribute towards the project goals. However, only work that is of academic interest is discussed in this thesis.

2.1 Formula Student Driverless

Formula Student/Formula SAE is the world's largest engineering contest for students, with over 600 teams from universities all over the world [4]. The objective of the competition is to ”conceive, design, fabricate, develop and compete with small, formula style, race cars” [3]. For each competition season, the participating teams build a completely new vehicle in about 8 months, and compete at one or more competition sites around the world.

The competition is divided into three classes: Internal Combustion Engine Vehicle (CV), Electrical Vehicle (EV) and Driverless Vehicle (DV). The DV class was introduced last year at Formula Student Germany (FSG) in Hockenheim, and is this year hosted in Hungary, UK and Italy as well. Revolve NTNU are participating in the competitions in Germany and in Hungary.

The teams compete in a series of static and dynamic events, which are designed to test the engineering skills of the students. The winner of the competition is the team with the highest total score from all the events. See fig. 2.1 for the distribution of available points among the different events. The formulae for determining the scoring in each event can be found in [3].

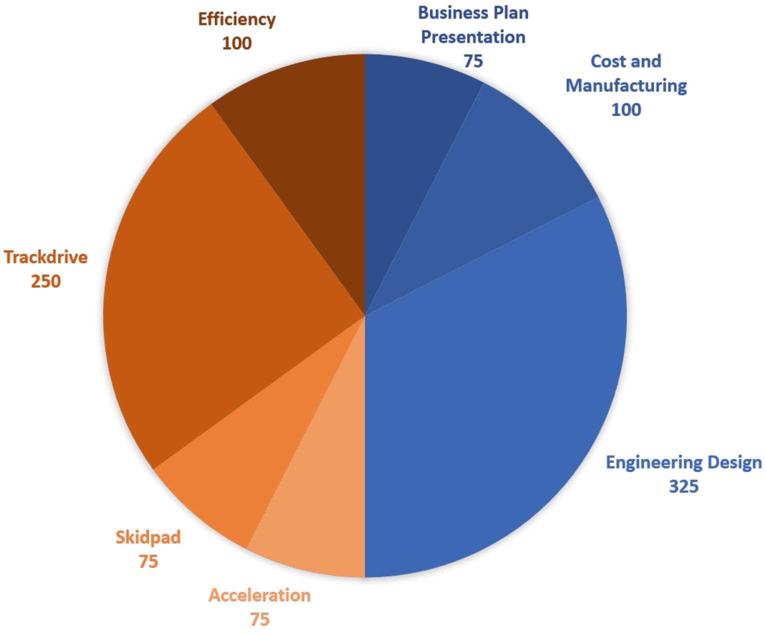

Figure 2.1: Competition Scoring

2.1.1 Static events

Formula student is an engineering competition, not a racing competition. Therefore, the static events are just as important as their dynamic counterparts. For the DV class, an equal number of points are awarded in both types of events. There are four static events for the DV class. In the **Business Plan Presentation** each team must present a business model to convince the judges, acting as potential investors, that their race car prototype is a good investment. The event is held as a presentation, with a following discussion.

In the **Cost and Manufacturing** event, the teams must show their understanding of the manufacturing processes and the cost required to build their prototype race car, as well as a mass production version. The teams are evaluated based on three cost report documents, and a discussion with the judges.

In the **Engineering Design** event, the teams are evaluated on every part of the design process, from the specific design choices, to proper use of the relevant knowledge and tools, and data validation. For the DV class, both the base vehicle and the autonomous system must be covered.

2.1.2 Dynamic events

Even though the static and dynamic events are weighed equally in terms of points in the DV class, the dynamic events are the soul and heart of Formula Student. The teams that have prepared well enough to pass the rigorous technical inspection required to enter the dynamic events, can finally release their car to the tarmac at the Hockenheimring or other

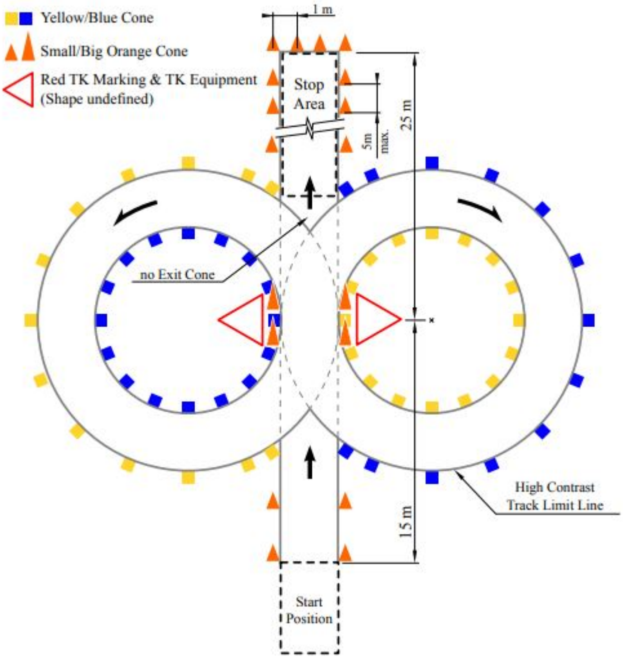

Figure 2.2: The Skidpad track

competition grounds. This is where the race cars are pitted against each other, to burn rubber and push the design to its limit. This is where the hard work of a whole year is measured in lap times and points, and for a select few, in a place on the podium.

In the DV class, the course for each dynamic event is marked by colour-coded traffic cones: blue cones for the left hand border of the track, yellow for the right hand border, and orange cones for the start and stop areas. A time penalty is added for each cone that is pushed over or knocked out of place during a run.

Acceleration tests the vehicle's longitudinal performance. The course is a 75 m straight line, and the objective is to cross the finish line in the shortest time possible.

Skidpad tests the vehicle's lateral performance. The course consists of two circles in a figure eight pattern, see fig. 2.2, with an inner diameter of 15.25 meters and a track width of 3 meters. The vehicle takes two consecutive laps of each circle, and the average time of the final lap for each circle determines the final score.

Trackdrive is the most challenging of the dynamic events for the DV class. In FSG 2017, only one team were able to complete this event. The vehicle must complete 10 laps on an unfamiliar closed circuit, and stop within a designated area. The length of the circuit is between 200 and 500 meters, and the track width is slightly over 3 meters.

The **Efficiency** score is based on the vehicle's power consumption during the trackdrive event.

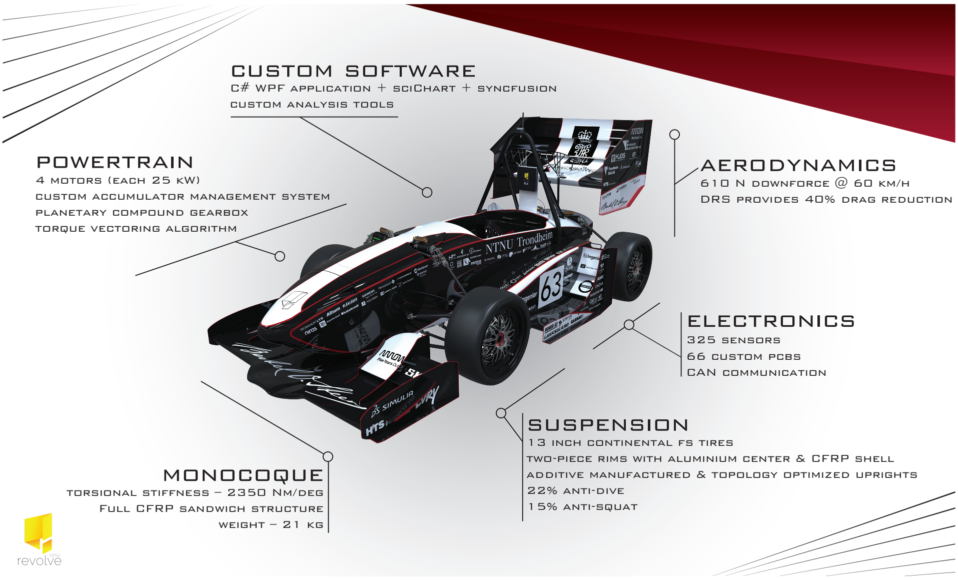
CUSTOM SOFTWARE
C# WPF APPLICATION + SCICHART + SYNCFUSION
CUSTOM ANALYSIS TOOLS
POWERTRAIN
4 MOTORS (EACH 25 KW)
CUSTOM ACCUMULATOR MANAGEMENT SYSTEM
PLANETARY COMPOUND GEARBOX
TORQUE VECTORING ALGORITHM
AERODYNAMICS
610 N DOWNFORCE @ 60 KM/H
DRS PROVIDES 40% DRAG REDUCTION
ELECTRONICS
325 SENSORS
66 CUSTOM PCBS
CAN COMMUNICATION
SUSPENSION
13 INCH CONTINENTAL FS TIRES
TWO-PIECE RIMS WITH ALUMINIUM CENTER & CFRP SHELL
ADDITIVE MANUFACTURED & TOPOLOGY OPTIMIZED UPRIGHTS
22% ANTI-DIVE
15% ANTI-SQUAT
MONOCOQUE
TORSIONAL STIFFNESS – 2350 NM/DEG
FULL CFRP SANDWICH STRUCTURE
WEIGHT – 21 KG
NTNU Trondheim
63
revolve

Figure 2.3: Eld as an Electric Vehicle

2.1.3 Eld - a Formula Student race car

Eld was Revolve NTNU's entry into the EV class in the 2017 season, see fig. 2.3, and is being repurposed as a DV class vehicle for the 2018 season, see fig. 2.4. Eld is an individual-wheel drive electric race car, with one motor mounted in the upright of each wheel, which can be individually controlled. Each motor can output a maximum of 21 Nm driving torque, and 18 Nm braking torque through its *kinetic energy recovery system* (KERS). The motors are connected to the Continental 13 inch tires through a compund planetary gearbox with a gear ratio of 15.58:1. They draw power from a litium-cobolt battery pack, which can output 80kWh.

Eld weighs in at 176 kg without a driver, much thanks to the chassis, which is made out of a carbon fibre and aluminum honeycomb sandwich structure. The aerodynamic package produces 610 N downforce at 60 km/h.

Eld boasts an impressive 325 sensors, most of which are used for battery, temperature and power management. One of the more interesting sensor units for analyzing vehicle dynamics, is the high-performance GPS-aided INS. It comes with a integrated Kalman filter, which outputs good estimates of linear velocities and accelerations, orientation and angular velocities. It also provides position, with a root-mean square error of 2.5 meters [5].

Encoders measure the rotation of each motor, as well as that of the steering wheel. Finally, the suspension includes four linear potentiometers to measure the compression of each spring/damper. As part of the conversion to a DV class vehicle, Eld has been fitted with electrically actuated steering and braking systems, as well as a remotely triggered emergency brake system. Two cameras and a 360 deg lidar are used to perceive the surroundings, and a DGPS is added, which gives a much higher positional accuracy than the

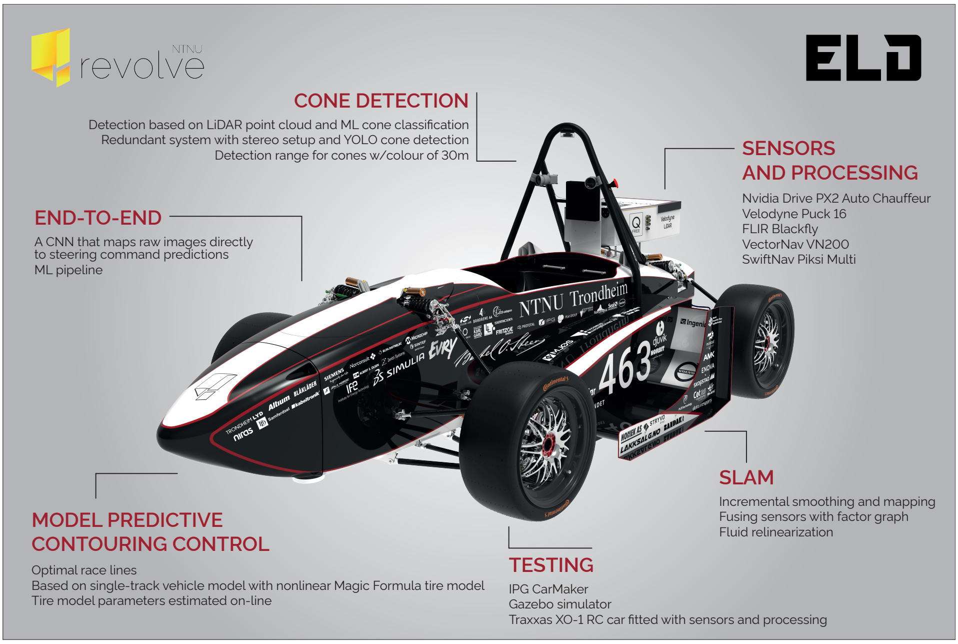
revolve NTNU
ELD
CONE DETECTION
Detection based on LiDAR point cloud and ML cone classification
Redundant system with stereo setup and YOLO cone detection
Detection range for cones w/colour of 30m
SENSORS AND PROCESSING
Nvidia Drive PX2 Auto Chauffeur
Velodyne Puck 16
FLIR Blackfly
VectorNav VN200
SwiftNav Piksi Multi
END-TO-END
A CNN that maps raw images directly to steering command predictions
ML pipeline
NTNU Trondheim
463
SLAM
Incremental smoothing and mapping
Fusing sensors with factor graph
Fluid relinearization
MODEL PREDICTIVE CONTOURING CONTROL
Optimal race lines
Based on single-track vehicle model with nonlinear Magic Formula tire model
Tire model parameters estimated on-line
TESTING
IPG CarMaker
Gazebo simulator
Traxxas XO-1 RC car fitted with sensors and processing

Figure 2.4: Eld as a Driverless Vehicle

Vectornav GPS. An Nvidia Drive PX2 is used as the main computer unit.

In order to fit the autonomous system, and because it is deemed unnecessary at the performance level expected for the DV class vehicle, the aerodynamic package is removed. As a result of the higher demand for low voltage power, the high voltage battery pack has been reduced in size to accommodate several DC-DC converters. This has reduced the maximum power output of the battery pack to 60 kWh.

2.2 Revolve NTNU team 2017 tractive system control systems

The Revolve NTNU tractive system control systems (TSCS) consists of several driver assist modules, see fig. 2.5. The purpose of these systems, apart from following the FS rules, is to optimize utilization of the available tire-road grip to produce the highest possible acceleration in the appropriate direction. In short, to help get the car around the track as fast as possible. The control systems have been developed since the 2016 season, and have been the subject of two master's theses [6][7].

The two theses give an extensive description of the control systems, as well as provide experimental data for their performance. In this sub-chapter, a brief overview of the current control systems at the beginning of the 2018 season is presented, as it serves as the foundation for the autonomous control systems developed in this thesis.

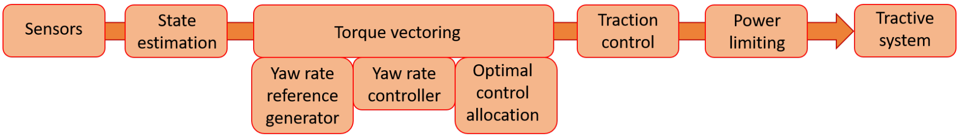

Figure 2.5: Driver-assistance tractive control system structure

2.2.1 State estimation

The control system relies on knowledge about the tires' interactions with the road surface, which are not readily measurable. Some state estimation is therefore necessary.

The normal tire force estimations are used in eq. (2.11) to estimate available grip, which in turn is used to calculate the maximum torque output for each motor.

$$F_z^{ij} = F_{z,0}^{ij} + K_s^{ij}\Delta_s^{ij} \pm K_{ARB}^{ij}\phi L_w ij \tag{2.1}$$

The longitudinal tire force estimations are used to calculate the corrective yaw moment produced by the torque allocation algorithm, which is used as a feedback term to the yaw rate controller. It is calculated using Newton's second law of rotation, relying on the differentiated measurements from the wheel encoders.

$$F_x^{ij} = \frac{\tau^{ij} GR - J_w \dot{\omega}^{ij}}{R_{eff}} \tag{2.2}$$

The friction coefficient is assumed constant, and is guesstimated.

2.2.2 Torque vectoring

Revole NTNU built its first independent four-wheel drive vehicle in 2016. The main advantage of this type of drivetrain is that the torque output of each wheel can be individually controlled to serve a master plan. For example, by applying more torque to the outer wheels than to the inner wheels during cornering, a moment about the z-axis is created, which will rotate the vehicle independent of the steering angle, in the manner of a continuous track vehicle. This is known as *torque vectoring*. Coupled with conventional steering, torque vectoring results in very quick cornering and good utilization of available grip.

The torque vectoring algorithm consists of three modules, as seen in fig. 2.5: Yaw rate reference generator, yaw rate controller and torque allocation.

The reference yaw rate is generated from the following equation, which is derived from the idealized bicycle model driving in a circle of constant radius [6].

$$r_r = \tan(\delta)\frac{u}{l + K_v u^2} \tag{2.3}$$

K_v is the vehicle's under-steer coefficient, defined as

$$K_v = \frac{m(l_r C_r - l_f C_f)}{2C_f C_r l}. \tag{2.4}$$

The reference yaw rate is upper bounded by the theoretical maximum yaw rate, given by

$$r_{r,max} = \frac{a_{y,max}}{u} = \frac{\mu g}{u} \tag{2.5}$$

The yaw rate controller is implemented as a PI feedback controller with gain scheduling for different velocity ranges, and anti-windup. The output of the yaw rate controller is a corrective yaw moment, M_z, which should be produced by the torque allocation.

The torque allocation module aims to produce both the longitudinal acceleration requested by the driver and the corrective yaw moment requested by the yaw rate controller. With four independent motors, this becomes an over-actuated system. A quadratic programming problem is formulated to calculate the optimal torque allocation:

$$\begin{aligned} \min_{\boldsymbol{\tau}} \quad & q(\boldsymbol{\tau}) = \frac{1}{2}\boldsymbol{\tau}^\top \boldsymbol{G}\boldsymbol{\tau} + \boldsymbol{\tau}^\top \boldsymbol{c} \\ \text{s.t.} \quad & \boldsymbol{A}_{eq}\boldsymbol{\tau} = b_{eq} \\ & \boldsymbol{A}_{ineq}\boldsymbol{\tau} \geq \boldsymbol{b}_{ineq}, \end{aligned} \tag{2.6}$$

where $\boldsymbol{\tau} = \begin{bmatrix}\tau_{FL} & \tau_{FR} & \tau_{RL} & \tau_{RR}\end{bmatrix}^\top$ is the torque output for each motor.

The cost function consists of three terms. The first term minimizes the error in the generated yaw moment given by

$$\begin{aligned} q_1(\boldsymbol{\tau}) &= (M_{z,tot} - M_{z,r})^2 \\ &= \left(\begin{bmatrix} h_{FL} & h_{FR} & h_{RL} & h_{RR} \end{bmatrix} \begin{bmatrix} \tau^{FL} \\ \tau^{FR} \\ \tau^{RL} \\ \tau^{RR} \end{bmatrix} - M_{z,r} \right)^2, \end{aligned} \tag{2.7}$$

where $M_z^{ij} = h_{ij}\tau^{ij}$ is the yaw moment generated by tire *ij* when applying a motor torque τ^{ij} on motor *ij* (see [7] for details). Formulated as a quadratic function of $\boldsymbol{\tau}$, this becomes

$$q_1(\boldsymbol{\tau}) = \frac{1}{2}\boldsymbol{\tau}^\top \boldsymbol{G_1}\boldsymbol{\tau} + \boldsymbol{\tau}^\top \boldsymbol{c}, \tag{2.8}$$

where

$$\boldsymbol{G_1} = \begin{bmatrix} 2h_{FL}^2 & h_{FL}h_{FR} & h_{FL}h_{RL} & h_{FL}h_{RR} \\ h_{FR}h_{FL} & 2h_{FR}^2 & h_{FR}h_{RL} & h_{FR}h_{RR} \\ h_{RL}h_{FL} & h_{RL}h_{FR} & 2h_{RL}^2 & h_{RL}h_{RR} \\ h_{RR}h_{FL} & h_{RR}h_{FR} & h_{RR}h_{RL} & 2h_{RR}^2 \end{bmatrix} \tag{2.9a}$$

$$\boldsymbol{c} = -2M_{z,r} \begin{bmatrix} h_{FL} \\ h_{FR} \\ h_{RL} \\ h_{RR} \end{bmatrix} \tag{2.9b}$$

The term $M_{z,r}^2$ is omitted from the cost function because it is constant.

The second term aims to avoid tire saturation by applying torque to each wheel in proportion to the available grip.

$$q_2(\boldsymbol{\tau}) = \left(\frac{\tau^{FL}}{\tau^{FL}_{sat}}\right)^2 + \left(\frac{\tau^{FR}}{\tau^{FR}_{sat}}\right)^2 + \left(\frac{\tau^{RL}}{\tau^{RL}_{sat}}\right)^2 + \left(\frac{\tau^{RR}}{\tau^{RR}_{sat}}\right)^2, \tag{2.10}$$

where

$$\tau^{ij}_{sat} = \frac{F_{z,ij} R \mu}{GR} \tag{2.11}$$

is the saturation torque for wheel *ij*. Formulated as a quadratic function, we get

$$q_2(\boldsymbol{\tau}) = \frac{1}{2}\boldsymbol{\tau}^\top \boldsymbol{G_2} \boldsymbol{\tau}, \tag{2.12}$$

where

$$\boldsymbol{G_1} = \begin{bmatrix} \left(\frac{1}{\tau^{FL}_{sat}}\right)^2 & 0 & 0 & 0 \\ 0 & \left(\frac{1}{\tau^{FR}_{sat}}\right)^2 & 0 & 0 \\ 0 & 0 & \left(\frac{1}{\tau^{RL}_{sat}}\right)^2 & 0 \\ 0 & 0 & 0 & \left(\frac{1}{\tau^{RR}_{sat}}\right)^2 \end{bmatrix} \tag{2.13}$$

The third term is a 4x4 identity matrix, which is added to ensure that $\boldsymbol{G}$ is positive definite even when one or more wheels lift from the ground, which makes the problem strictly convex [8]. In addition, this terms helps to evenly distribute the requested torque among the four wheels. Thus, the final cost function is given by

$$\begin{aligned} q(\boldsymbol{\tau}) &= k_1 q_1(\boldsymbol{\tau}) + k_2 q_2(\boldsymbol{\tau}) + k_3 q_3(\boldsymbol{\tau}) \\ &= \frac{1}{2}\boldsymbol{\tau}^\top \big(k_1 \boldsymbol{G_1} + k_2 \boldsymbol{G_2} + k_3 \boldsymbol{I_4}\big)\boldsymbol{\tau} + \boldsymbol{\tau}^\top k_1 \boldsymbol{c}, \end{aligned} \tag{2.14}$$

where the constants, k_i weigh the different terms against each other.

The equality constraint, $\boldsymbol{A}_{eq}\boldsymbol{\tau} = b_{eq}$, ensures that the total torque satisfies the torque requested by the driver, T_{req}. The inequality constraint, $\boldsymbol{A}_{ineq}\boldsymbol{\tau} \geq \boldsymbol{b}_{ineq}$, ensures that the requested torque is within the capacity of each motor, both in terms of available grip, τ_{sat} and maximum/minimum torque output, $\tau_{max,motor}$, $\tau_{min,motor}$. Thus $\tau^{ij}_{max} = min(\tau^{ij}_{sat}, \tau_{max,motor})$, $\tau^{ij}_{min} = max(-\tau^{ij}_{sat}, \tau_{min,motor})$. Both constraints are formulated differently depending on whether the vehicle is accelerating or decelerating. The matrices are:

$$
\begin{aligned}
&A_{eq} = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}, \\
&b_{eq,acc} = min(T_{req}, \sum \tau^{ij}_{max}), \quad b_{eq,dec} = max(T_{req}, \sum \tau^{ij}_{min}) \\
&\boldsymbol{A}_{ineq} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
&\boldsymbol{b}_{ineq,acc} = \begin{bmatrix} -\tau^{FL}_{max} \\ -\tau^{FR}_{max} \\ -\tau^{RL}_{max} \\ -\tau^{RR}_{max} \\ -\tau^{FL}_{min} \\ -\tau^{FR}_{min} \\ -\tau^{RL}_{min} \\ -\tau^{RR}_{min} \end{bmatrix}, \quad \boldsymbol{b}_{ineq,dec} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -\tau^{FL}_{min} \\ -\tau^{FR}_{min} \\ -\tau^{RL}_{min} \\ -\tau^{RR}_{min} \end{bmatrix}
\end{aligned}
\tag{2.15}
$$

The QP problem is solved using an active set solver, which is described at length in [7].

2.2.3 Traction control

If a tire experiences excessive slip despite the efforts of the torque vectoring module, the traction control module overrules the allocated torque request. A torque reduction is calculated by a PI controller, and added to the allocated torque.

$$\tau^{ij}_{red} = K_{P,TC}\tilde{SR}^{ij} + K_{I,TC}\int_0^t \tilde{SR}^{ij} dt, \tag{2.16}$$

where $\tilde{SR}_{acc} = min(0, SR - SR_{max})$, $\tilde{SR}_{dec} = max(0, SR + SR_{max})$ SR_{max} is a constant obtained by adding a margin of error to the optimal slip ratio, SR_{opt} , defined by the peak of the tire model, see fig. 4.8. The error margin is added because the generated tire force decrease much faster when $abs(SR) < SR_{opt}$ than when $abs(SR) > SR_{opt}$. Thus a controller working around the optimal slip ratio would perform worse than one operation at a slightly higher slip. This is illustrated in fig. 2.6.

2.2.4 Power limiting

The power limiting module serves two purposes. It ensures that the tractive system does not overload the battery pack by drawing too much power, or by producing too much power during regenerative braking, and it ensures that the vehicle upholds Formula Student

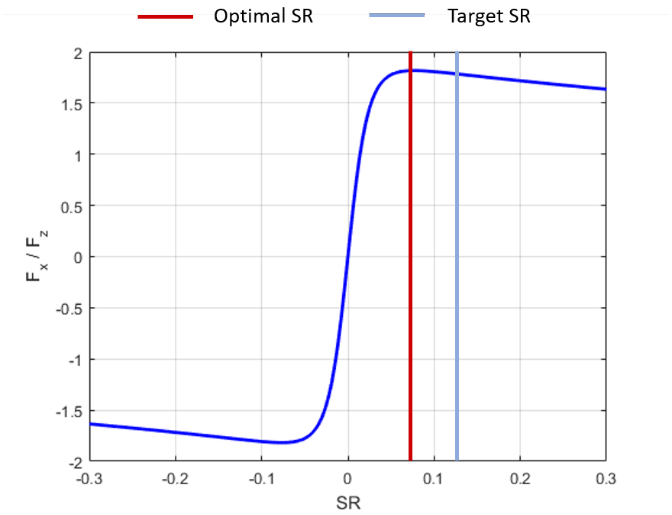

Figure 2.6: Target slip ratio for Traction control vs. optimal slip ratio.

regulation D9.4.1: ”A violation is defined as using more than the maximum power [...] or exceeding the specified voltage, [...] after a moving average over 500 ms [...]” [3].

The power limit module is implemented in much the same way as the traction control module. A PI controller calculates a reduction factor, F_{red}, based on the power violation, $\tilde{P}$. The torque request to each motor is multiplied by this factor.

$$F_{red} = K_{P,PL}\tilde{P} + K_{I,PL}\int_0^t \tilde{P}dt, \tag{2.17}$$

where $\tilde{P}_{acc} = min(0, P - P_{lim})$ and $\tilde{P}_{dec} = min(0, P_{KERS} - P_{lim,KERS})$.

2.2.5 Upgrades for team 2018

Several major improvements are made to the TSCS for this season, by members of the EV team. The normal force (F_z^{ij}) estimation is extended to use a complete model of the suspension, including the anti-roll-bar. A launch control module is added specifically for the acceleration event, which aims to optimize the slip ratio for each wheel, thus maximizing the longitudinal acceleration. The power-limiting module is also completely reworked.

2.3 Revolve NTNU Driverless Project plan

The project is organized with the two overall objectives in mind. The primary objective takes precedence over the secondary, which means that until the primary objective is under control, work on the secondary objective will not be prioritized.

The primary objective for the project is to finish all the dynamic events at the two competitions. To fulfill this objective, a minimum viable prototype (MVP) of the autonomous

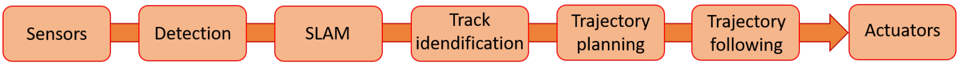

Figure 2.7: The driving pipeline

software will be developed, which must be able to detect and follow an unfamiliar track without stopping or running outside the track boundaries.

The secondary objective is to place top 5 overall at both competitions. In 2017, this would be implied by the primary goal, as only one of the 15 attending teams managed to complete all dynamic events. However, with several teams now having had two years to develop their autonomous systems, this might no longer be the case. Therefore, a high-performance version of the autonomous software will be developed, which aims to make the vehicle complete the dynamic events as quickly as possible.

The autonomous system is organized in several modules, in what we call the *driving pipeline*, see fig. 2.7. My area of responsibility is the trajectory following module, including the interface to the path planning- and actuation modules. The remaining modules will not be discussed further in this thesis.

The autonomous software is written in the Robotic Operating System (ROS) framework [9]. ROS is an open-source software framework for robotics applications, which is used extensively in research and industry. It was chosen primarily because a large collection of drivers, programs and libraries useful for robotics application exists for ROS, as well as several useful tools for visualization and configuration. Each piece of software is written as an autonomous node, and ROS handles inter-node communication and scheduling. This makes it easy to make a modular and well-organized software architecture, where subsystems can be changed or replaced without affecting the rest of the system.

On the downside, ROS is not a real-time framework, and is therefore not well-suited for tasks with strict real-time demands, such as low- and mid-level control. These tasks will therefore be carried out on the vehicle's embedded microcontrollers.

2.4 Trajectory Following

The trajectory following module consists of the motion control systems required to make the vehicle follow a feasible trajectory given by the path-planning module. This includes the tractive system control systems described in section 2.2, in addition to the state estimation required for these systems, beyond the vehicle's position, velocity and acceleration in 6 dof.

The trajectory following module can be divided into three levels: High-level, mid-level and low-level control systems, see fig. 2.8.

The high-level control systems are those that replace the driver, and decides, at a minimum, the desired steering angle and throttle input to the vehicle. Without the mechanical interface imposed by a human driver, these systems can pass on any number of additional control signals to the mid-level control systems, such as the yaw rate reference.

The low-level control systems are the actuator controllers: Torque control for the driving motors, position control for the steering system and velocity control for the service

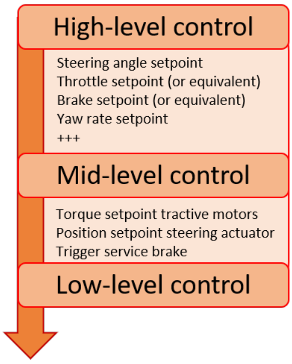
High-level control
Steering angle setpoint
Throttle setpoint (or equivalent)
Brake setpoint (or equivalent)
Yaw rate setpoint
+++
Mid-level control
Torque setpoint tractive motors
Position setpoint steering actuator
Trigger service brake
Low-level control

Figure 2.8: Modular separation of trajectory following

brake system. These are embedded in the proprietary motor controllers, and will not be discussed in great detail.

The mid-level control systems encompass everything that happens between the high-level and the low-level control systems, most notably the TSCS, described in section 2.2. They provide the reference values for the low-level control systems.

It is convenient to distinguish between high- and mid level control systems in this project for several reasons. The systems are separated in hardware, as high-level control systems are run on the main processing unit, while the mid-level control systems are run on the vehicle's embedded systems. The design becomes more modular, which lets the team use different high-level controllers on different vehicles without modification. Finally, the mid-(and low-)level control systems for the tractive system are shared with the EV team, where two team members are working exclusively on these systems. Therefore, by keeping these as a separate sub-module, these systems can remain EV compatible, and any improvements will benefit both projects.

2.4.1 Upstream interface

Two different strategies for trajectory planning will be used, which require different interfaces between the trajectory planning and -following modules. In the first, a global trajectory will be planned based on the mapped sections of the track. The trajectory following module must analyze its state relative to the desired trajectory, and calculate control inputs to make the vehicle follow the planned trajectory as closely as possible. In the second strategy, model predictive contouring control (MPCC) is used to calculate an optimal trajectory on-line, using a vehicle model and finding the sequence of control inputs which maximize the progression along the center line of the track. In order to solve for a sufficiently long prediction horizon, the MPCC algorithm runs at a moderately low frequency and uses a simplified vehicle model. The mid-level control systems are run at a high rate, and can be used to support the MPCC by forcing the vehicle to follow the trajectory predicted by the MPCC.

Chapter 3

Thesis Objectives

The overall objective of the thesis is to develop and implement the necessary trajectory following systems required to fulfill the two main goals for Revolve NTNU Driverless Team 2018 described in section 2.3: To complete all the dynamic events, and to place in top 5 at the competitions in Germany and Hungary.

Towards the primary goal, and in accordance with the overall project plan, an MVP high-level controller (as defined in section 2.4) must be implemented. The purpose of the MVP high-level controller is to have a solution for trajectory following which is easy to implement and configure and is robust against irregular state estimates and reference trajectory. Computational efficiency and high-speed performance is not required. The controller should be simple and intuitive to tune on-line to a decent performance, either by a transparent design where the relation between behaviour and each tuning parameter is apparent, or by a self-tuning controller.

The mid-level control systems for the steering system and the service brakes must also be implemented, as well as an interface to the TSCS. However, the MVP does not need to make use of the torque-vectoring module in the TSCS.

Towards the secondary goal, one must assume that the vehicle is required to drive at speeds beyond the grasp of the MVP. Therefore, a high-performance high-level controller must be implemented. The high-performance high-level controller must be able to follow a feasible trajectory much more closely, and at higher speeds than the MVP high-level controller. It must also be able to closely follow a feasible velocity profile. These are soft requirements, as the performance guaranteed by the controller dictates the margin of error used in the path-planning node, which means that a better controller can follow a quicker trajectory. The high-performance high-level controller should also make full use of the TSCS.

The TSCS must be modified to provide an interface tailored to the autonomous system, in parallel with the human interface. The requirements for this interface will be determined by the high-level control algorithm(s) implemented.

The remaining work capacity should be used to improve the TSCS, as they play a key role in high-speed performance, both for the DV and the EV. An important aspect of the

TSCS that is not being addressed by the members of the EV team is the tire-road force and friction estimation. In order to fully utilize the available road grip, it is crucial to know how much road grip is available, and how much is being utilized at a given time. Currently, only the longitudinal force components are being estimated, and only using a simple open-loop estimator.

This thesis contribution to the TSCS will therefore be to implement a method to estimate longitudinal and lateral tire-road forces and the coefficient of friction. The longitudinal force estimate should be more precise than the current version. The system should be EV compatible, which means it should not make use of the extra sensors introduced in the autonomous systems. It must also be run on the embedded system, meaning computing power is limited.

To summarize, the sub-objectives of this thesis are to:

1. Implement a minimum viable prototype high-level controller.
2. Implement the necessary mid-level controllers for the steering system and the service brakes, and provide the MVP with an interface to the TSCS.
3. Implement a high-performance high-level controller.
4. Modify the mid-level tractive control systems as required by the high-performance high-level controllers.
5. Implement an estimator for longitudinal and lateral tire forces, the coefficient of friction.

Chapter 4

Methodology

For each of the systems developed in this thesis, the overall development process has been:

- Mathematical modelling
- Software implementation
- Testing in one or more testing platform
- Implementation on Eld

The mathematical modelling for each system is presented in sections 4.2 to 4.5. The two high-level controllers and several peripherals are programmed as ROS nodes in C++, to be run on the main processing unit. The TSCS modules are programmed in MATLAB/Simulink, and included in the embedded software as auto-generated C code. The actual software implementation is not presented in this thesis, as Revolve NTNU wants to keep its software private - it is a competition after all. The testing platforms are discussed in section 4.1.

4.1 Testing platforms

Three main testing platforms were used to test software before implementation on Eld: An RC-car)fig. 4.2), a Gazebo simulation (section 4.1.2) and IPG CarMaker (section 4.1.3). The different platforms are described in the following sub-sections. The three platforms have different properties, and have been used for different purposes at different stages in the development process. An overview is presented in fig. 4.1. Unfortunately, due to problems with the detection module, the project did not progress to the point where high-performance trajectory following was necessary. As soon as this became evident, all work on these systems was halted, apart from documentation. This was done in accordance with the project plan described in section 2.3, in order to direct manpower towards more pressing tasks. This is marked by orange background color in fig. 4.1.

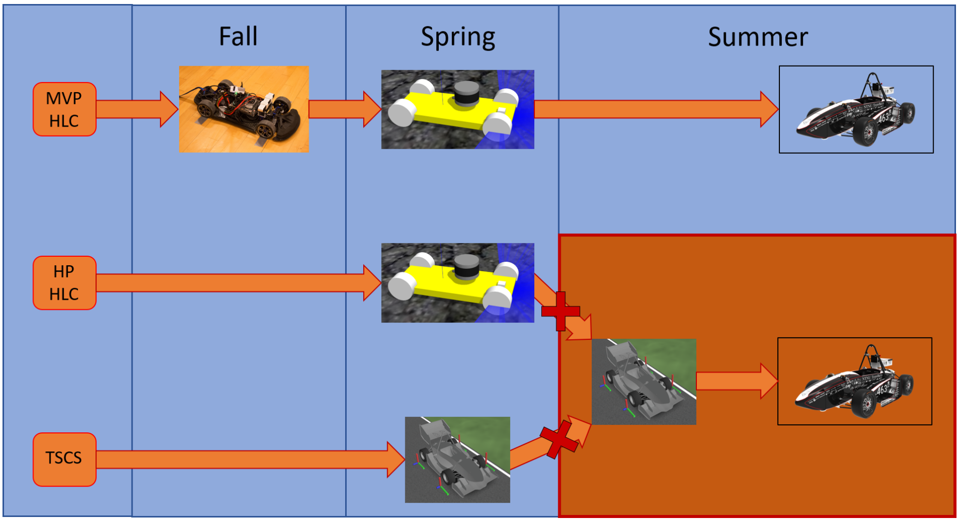

Figure 4.1: Use of development tools throughout the project. The section with red background contains work that was abandoned in favour of more pressing tasks.

4.1.1 RC-car

A 1:7 scale RC-car, see fig. 4.2, is used as a prototype platform for the autonomous sensors and software system in the early stages of development, before either the simulator or the autonomous hardware system on Eld are functional. This allows for iterative development and hardware-in-the-loop (HIL) testing from the very beginning of the project. The RC-car is used for integration testing and qualitative testing of the MVPHLC.

4.1.2 Gazebo

Gazebo [10] is a 3D robot- and environment simulator, which is integrated with ROS. In addition to a physics engine, it provides virtual sensors, which give realistic data. A vehicle

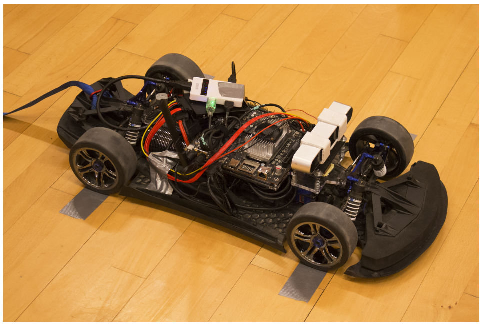

Figure 4.2: RC-car used as a prototype platform in the early stages of the project.

model was created by one of the DV team members, fitted with a virtual LiDAR and INS, and different race tracks were created using 3D modelled traffic cones, see section 4.1.2. This lets us test the whole autonomous pipeline, apart from the camera detection module, up to the high-level controllers. This simulator is used for qualitative testing and for integration testing for both HLCs, but the vehicle dynamics are not accurate enough to use it for bench-marking of estimators or control systems.

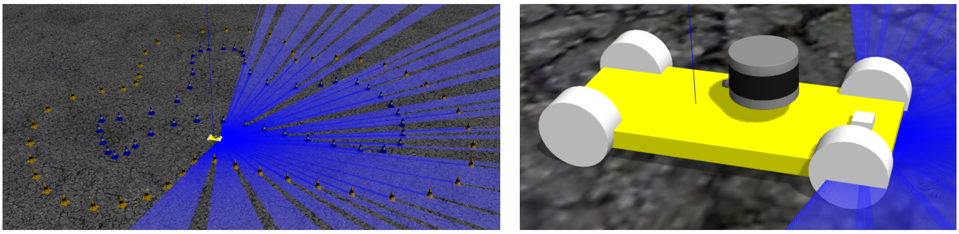

Figure 4.3: Gazebo simulator.

4.1.3 IPG CarMaker

IPG CarMaker [11] is a commercial car simulator which is used by a lot of Formula Student teams, and has been used to develop EV control systems and vehicle dynamics since 2016. It provides a high-fidelity vehicle model of a formula student-type race car, which Revolve has customized to match Eld in both appearance and dynamics. It also comes with an automated driver. Control systems and state estimation algorithms are programmed in Simulink, and fed into the CarMaker vehicle model through CarMaker's Simulink interface. C-code is automatically generated from the Simulink code and uploaded to the custom-made vehicle control unit (VCU), which is then HIL-tested using the same interface.

IPG CarMaker is used for quantitative testing of the TSCS. The plan was to also use IPG CarMaker to test the HPHLC as well as the interaction between the HPHLC or the MPCC and the TSCS. However, replacing the automated driver module in IPG CarMaker with a custom high-level controller is nontrivial, and was not prioritized when it became evident that these systems would not be implemented on Eld this season.

A replica of the Autocross track from FSG 2006, see section 4.1.3, is used for simulations for the tire-ground force and friction estimation algorithms. This track has a large variation in bends and straights, which gives very dynamic driving and is a good representation for a Formula Student track. Each simulation makes one lap of the track.

4.2 MVP high-level controller

In the MVP implementation, trajectory planning will simply find the center-line of the track, and set a low constant target velocity. The controller must stay within the bounds of the track, i.e. not deviate from the center-line by more than approximately 0.7 m (minimum

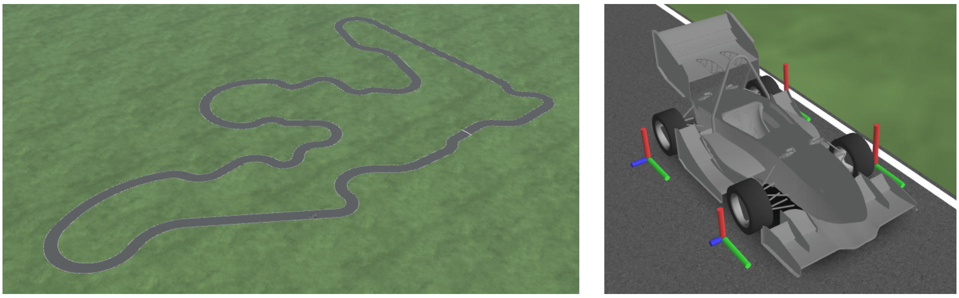

Figure 4.4: IPG CarMaker vehicle simulator.

track width - vehicle width)/2. The vehicle speed must be able to maintain a walking-pace, between 1 and 3 m/s, but no precise speed control is required.

The trajectory provided by the MVP path-planning module is a list of densely spaced points in 2D space, the path-tangential angle, ψ_t, and the reference speed, U_r, for each point. The trajectory interpreter finds the point closest to the vehicle by calculating the distance between the vehicle and each point on the trajectory, and selecting the nearest one. A look-ahead point is then found by iterating forward through the list of points until the distance between the closest point and the lookahead-point is greater than the lookahead-distance, Δ, see fig. 4.5. The reference heading is then found by drawing a line between the vehicle and the look-ahead point.

A PD controller is used to regulate the heading to the reference heading:

$$\delta_r = K_{P,\psi}\tilde{\psi} + K_{D,\psi}\frac{d}{dt}\dot{\tilde{\psi}} \tag{4.1}$$

Initially, a PD controller from cross-track error directly to steering angle was used, but this controller was discarded because it was too vulnerable to irregular values for $\tilde{y}$.

The longitudinal controller is a PI feedback controller with a feed-forward term for aerodynamic resistance, which outputs an equivalent throttle pedal position, TPP_{eq}:

$$TPP_{eq} = K_{P,U}\tilde{U}K_{I,U}\int_0^t \tilde{U}dt + K_{FF,U}U_r^2 \tag{4.2}$$

The MVP has 6 tuneable parameters, of which only 3 (Δ, $K_{P,\psi}$ and $F_{FF,U}$) are absolutely necessary to fulfill the requirements. The outputs of the two controllers are on the same form as that of a human driver, and the effect of each tuneable parameter is intuitively relateable to a physical reaction. Therefore, the MVP controller can quickly and easily be empirically tuned for a good performance.

4.3 High-performance high-level controller

The controller in this section is a feedback linearization controller built in three levels, see fig. 4.6. The control design is based on [12], and a single-track vehicle model, see

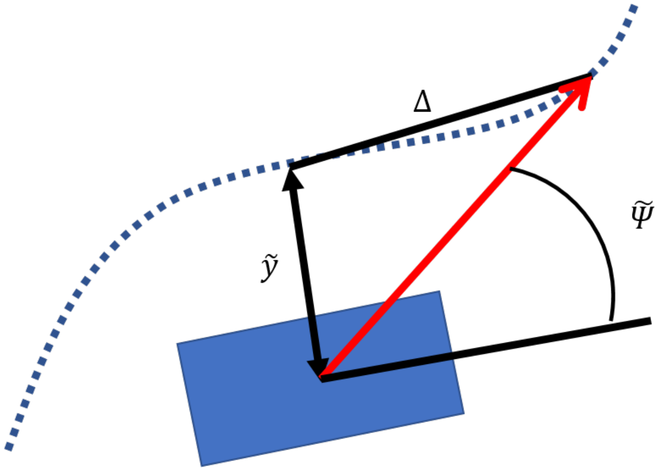

Figure 4.5: LOS guidance

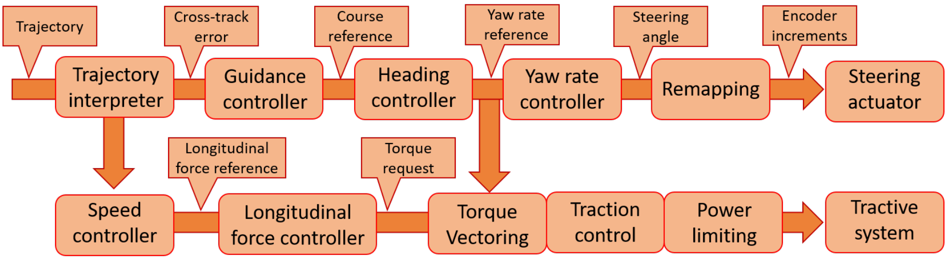

Figure 4.6: High-performance high-level trajectory following

fig. 4.7, using a linear tire model. The controller was chosen in part because it outputs a reference yaw rate, which can be sent directly to the torque vectoring module, taking full advantage of the independent four-wheel drive. Additionally, a feedback linearization controller is a good choice if the model parameters are well known. Using the on-line parameter estimation scheme described in section 4.5, this is ensured. The controller was developed in its basic form in [13], but is presented here for a better overview.

The trajectory passed on from the high-performance trajectory planning module consists of three cubic splines which denote the x and y position as well as the target velocity as a function of arc length. The closest point to the vehicle's center of gravity is found using a combination of Newton's method and the Bisection method, solved locally from the previous closest point.

The cross-track error, the path tangent and the signed curvature of the path are evaluated at this point, in addition to the value and the gradient of the target velocity.

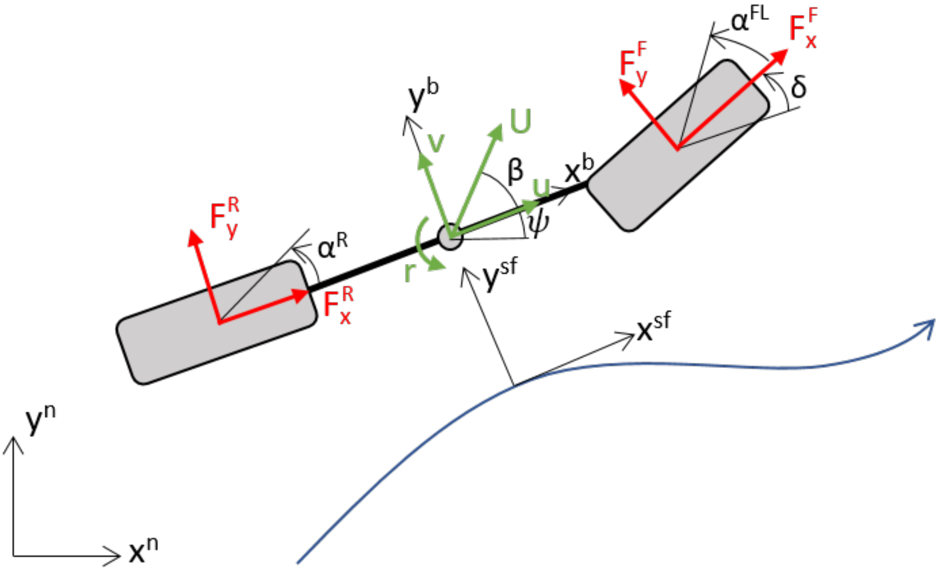

Figure 4.7: Single-track vehicle model and the coordinate frames used in the controller design.

4.3.1 Controller design

The first level is a LOS guidance controller with control law given by:

$$\chi_r^{sf} = atan\left(\frac{-y^{sf}}{\Delta}\right) \tag{4.3}$$

where Δ is a design parameter describing the path-tangential distance to the intersect point, assuming a straight path.

The second level is a feedback linearization course controller which controls $\chi^{sf} \rightarrow \chi_r^{sf}$, with the yaw rate reference as control input. The yaw dynamics in the sf frame are given by

$$\dot{\chi}^{sf} = r + \dot{\beta} - \kappa u^{sf}, \tag{4.4}$$

where

$$\dot{\beta} = \frac{d}{dt}\left(atan\left(\frac{v}{u}\right)\right) = \frac{\dot{v}u - \dot{u}v}{U^2} \tag{4.5}$$

The reference yaw rate is chosen as

$$r_r = \dot{\chi}_r^{sf} - \dot{\beta} + \kappa u^{sf} + K_{P,r}\tilde{\chi}^{sf} \tag{4.6}$$

From here, the yaw rate reference is passed down to the TV module, as well as being used to calculate the steering angle in the third level. The yaw dynamics for the single-track vehicle model with a linear tire model are given by

$$\dot{r} = \frac{1}{I_z}(-B_{y,s-t}\alpha^F l_f + B_{y,s-t}\alpha^R l_r), \tag{4.7}$$

where

$$\alpha^F = atan\left(\frac{v + rl_f}{u}\right) - \delta \tag{4.8a}$$

$$\alpha^R = atan\left(\frac{v - rl_r}{u}\right) \tag{4.8b}$$

and the tire model parameters are found using the parameter estimation algorithm described in section 4.5.2. Solving eq. (4.7) for δ we get

$$\delta = \frac{I_z}{B_{y,s-t}l_f}\dot{r} - atan\left(\frac{v + rl_f}{u}\right) + atan\left(\frac{v - rl_r}{u}\right)\frac{l_r}{l_f} \tag{4.9}$$

The reference steering angle is therefore given by

$$\delta_r = \frac{I_z}{B_{y,s-t}l_f}\dot{r}_d - atan\left(\frac{v + rl_f}{u}\right) + atan\left(\frac{v - rl_r}{u}\right)\frac{l_r}{l_f} + K_{P,\delta}\tilde{r} + K_{I,\delta}\int_0^t \tilde{r}dt \tag{4.10}$$

When the torque vectoring module is enabled, it replaces the integral.

The longitudinal controller is the same as eq. (4.2), with an added feed-forward term to account for changes in the reference speed. In addition, the output is a reference longitudinal force, instead of an equivalent throttle position, which means that the coefficients can be found by calculation instead of trial-and-error.

$$F_{x,r} = \frac{1}{2}\rho AC_d U_r^2 + m\dot{U}_r - K_{p,U}\tilde{U} - K_{i,U}\int_0^t \tilde{U}dt \tag{4.11}$$

4.4 DV mid-level control systems

The autonomous system requires a different interface to the car than a human driver. At the very least, this includes an interface to the steering system- and service brake actuators, as well as a torque request to the tractive system.

Additional interface channels can be implemented as required by the more advanced high-level controllers. A reference yaw rate is requested by the HPHLC as well as the MPCC, directly supplied to the yaw rate controller, bypassing the yaw rate reference generator entirely. For longitudinal control, both controllers output a required longitudinal force in Newtons, demanding the implementation for a longitudinal force controller in the tractive control system.

4.4.1 Steering system

The steering system is powered by a brushless DC motor with a gear ratio of 66:1, coupled with the steering shaft by a 1:1 spur gear transmission. The motor is position controlled by a proprietary control card, using an encoder on the motor shaft as feedback, and a hall-effect sensor as redundancy. The encoder has 1000 increments per revolution. This gives a linear mapping between the number of encoder increments away from the neutral position and the steering wheel angle in radians. The mapping from the mean steering angle, δ, and the steering wheel position, is found in a look-up table from [14].

$$encoder\_increments = 1000 * 66/(2\pi) * lookup\_steering\_wheel\_pos(\delta) \tag{4.12}$$

4.4.2 Service brakes

Regenerative braking has several advantages over mechanical brakes. It is easier to incorporate into the longitudinal force control described in section 4.4.3, gives a higher efficiency score during the trackdrive event, and does not require external actuation. Therefore, the car will rely solely on regenerative braking during a race. The service brakes are only used to stop the car at the end of a race, and to act as redundancy to the emergency brake system, in compliance with [3]. This makes the design process for the service brakes much simpler, as they can simply be on/off. The service brakes will not be discussed further in this thesis.

4.4.3 Tractive system longitudinal force controller

The Torque Allocation module uses an open-loop estimate for the longitudinal forces:

$$F_x^{ij} = \frac{GR\tau_r^{ij}}{R_{eff}^{ij}} \tag{4.13}$$

From section 2.2 we know that the Torque Allocation module accepts a torque request, T_r, which is used as an equality constraint when not limited by available grip. Including the tire force estimates in the TV optimization problem is beyond the scope of this thesis. Instead, the requested longitudinal force is translated into a torque request using the open-loop estimate as the feedforward, and a PI feedback controller based on the estimated forces from section 4.5. Integral windup protection is added for the cases when the torque requested is higher than that allocated by the torque allocation module.

$$T_r = \frac{F_{x,r}\bar{R}_{eff}}{GR} + K_{P,T}\tilde{F}_x - K_{I,T}\int_0^t \tilde{F}_x dt - K_{B,T}T_{unallocated}, \tag{4.14}$$

where $\tilde{F}_x = F_{x,r} - \sum_{ij} F_x^{ij}$.

The longitudinal force controller is tested in IPG CarMaker. A straight-line trajectory is set, and a varying longitudinal force reference profile is defined.

4.5 Tire-ground force and friction estimation

The tires are the only medium with which a race car can generate acceleration. When a tire moves relative to the ground, known as slip, it is deformed due to friction forces. It

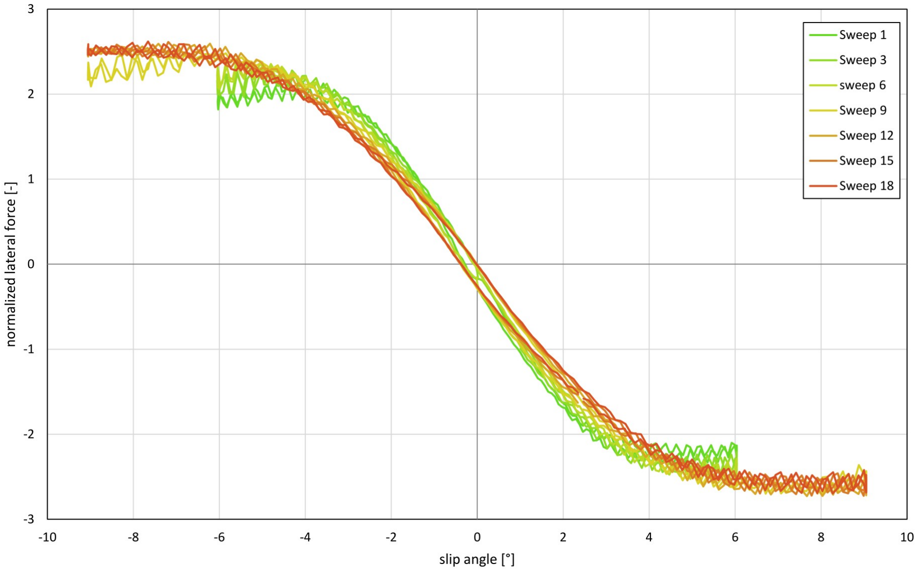

Figure 4.8: Tire force vs slip angle from tire test for the tires used on Eld.

coils up like a spring, creating force as it tries to regain its shape [15]. For longitudinal movement, the relative movement is expressed by the slip ratio, given by

$$\gamma = \frac{\omega R^{ij} - u^{ij}}{u^{ij}}, \tag{4.15}$$

and for lateral movement, it is expressed as the slip angle

$$\alpha = atan\left(\frac{v^{ij}}{u^{ij}}\right). \tag{4.16}$$

The force-slip curves are divided into three distinct regions, see fig. 4.8. For small slip values, the force grows linearly with increased slip. For larger slip values, the force grows nonlinearly, until it reaches the peak force, which defines the coefficient of friction. From here, the force decreases with increased slip, henceforth referred to as excessive slip. Different tire assemblies and different conditions give different variations of this curve. Notice, for example, how the peak force in fig. 4.8 increases with each sweep, as the tires are warmed up from the friction forces.

In a racing situation, it is desirable to be at peak force slip ratio, γ_{peak}, as much of the time as possible. However, since the force decreases as slip goes beyond γ_{peak}, a constant torque corresponding to peak force would result in rapidly increasing slip. Not only does this produce less force, but it consumes more energy because $P = \tau\omega$.

It is therefore important to know exactly how much force the tire is exerting, and what the maximum attainable force is. The maximum longitudinal and lateral forces are given

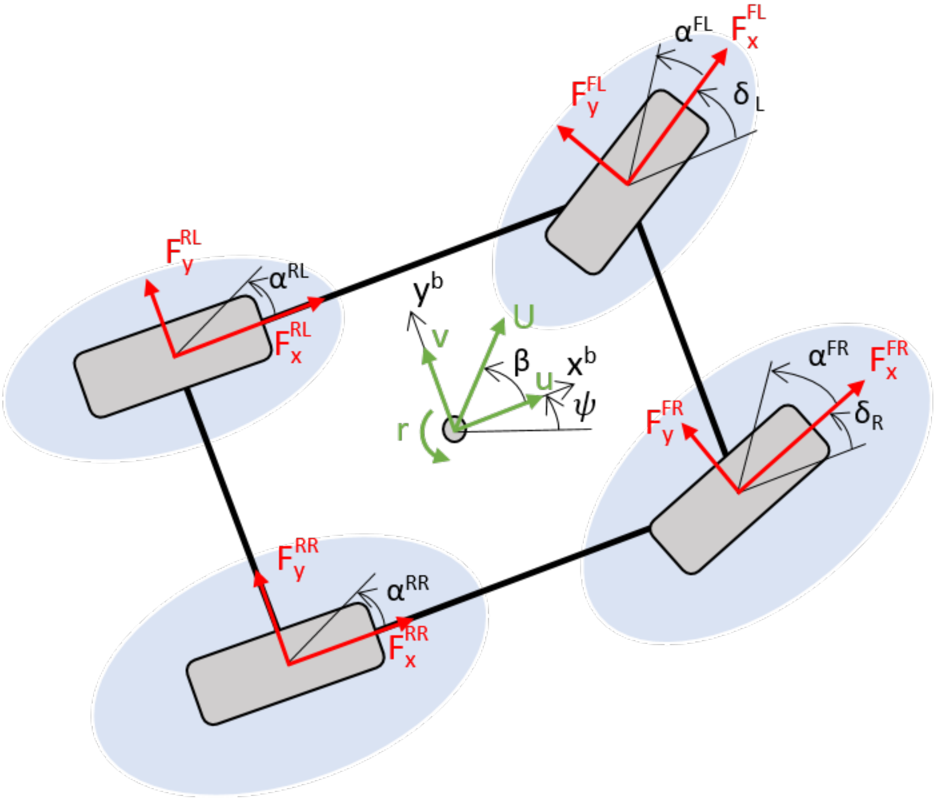

Figure 4.9: Tire-road forces and friction ellipses for a four-wheel vehicle.

by the friction ellipse:

$$\frac{{F_x^{ij}}^2}{(F_z^{ij}\mu_x)^2} + \frac{{F_y^{ij}}^2}{(F_z^{ij}\mu_y)^2} = 1, \tag{4.17}$$

It is an ellipse rather than a circle, because the tire has different properties in the longitudinal and the lateral direction, which results in two coefficients of friction, μ_x and μ_y. The friction ellipses are illustrated in fig. 4.9.

For a driverless vehicle, this is doubly important, as there is no driver who can reduce the throttle when they feel the wheels start to slip.

4.5.1 Tire-ground force and friction estimation using Kalman Filtering

The first approach to tire force and friction estimation was by using Kalman filtering. The method described in [16] and [17] was implemented in Simulink and tested in CarMaker. Several misprint errors in the model were corrected, and the Kalman filter was adjusted to conform with [18] and [12]. Despite this, the implementation was unsuccessful.

The leading theory is that only two equations of motion affect the four lateral tire forces in any significant way. The yaw dynamics distinguish the front forces from the rear, and sway and roll both determine the sum of forces, making one of them obsolete. This makes it impossible to distinguish F^{FL} from F^{FR} and F^{RL} from F^{RR}, resulting in the forces in each pair diverging in opposite directions.

In order to rule out sources of errors, the problem was simplified by providing measurements of all states which are estimated in [16], except the longitudinal and lateral tire forces and the effective rolling radius. However, the problem remained.

[19] uses input from a tire model as a measurement in the filter. The method was implemented using the nonlinear Magic Formula [20], which is valid for larger slip values. This turned out to work quite well, as long as the parameters for the tire model were good. However, since the input signals are already filtered, the Kalman filter becomes obsolete, and the tire model can be used directly. With this method, the problem is reduced to finding good parameters for the tire model.

4.5.2 On-line parameter estimation of parameters for Pacejka model

The *Magic Formula*, [20] is an empirical tire-road contact force model which is widely used in the automotive industry. The model gives longitudinal and lateral tire force as a function of slip ratio and slip angle, respectively, based on a large set of variables determined by the tire used and the suspension design. Ultimately, the input is boiled down to an expression with four non-physical coefficients, which fit a vast array of tires. The normalized Magic Formula with constant parameters [21] is given by

$$\frac{F}{F_z} = \mu sin(Ctan^{-1}(Bs - E(Bs - tan^{-1}(Bs)))) \tag{4.18}$$

Here, s is the slip ratio when calculating F_x, and slip angle when calculating F_y. In this section, it is assumed that the difference in the suspension design for each tire is negligible, so that we can use one set of parameters to model the longitudinal forces, and one set to model the lateral forces. Let the longitudinal and lateral forces on tire ij estimated by the Magic Formula be given by

$$\begin{aligned} F^{ij}_{x,m} &= F^{ij}_z \mu_x sin(C_x tan^{-1}(B_x\gamma^{ij} - E_x(B_x\gamma^{ij} - tan^{-1}(B_x\gamma^{ij})))) \\ F^{ij}_{y,m} &= -F^{ij}_z \mu_y sin(C_y tan^{-1}(B_y\alpha^{ij} - E_y(B_y\alpha^{ij} - tan^{-1}(B_y\alpha^{ij})))) \end{aligned} \tag{4.19}$$

The minus sign on the lateral force estimates is due to the coordinate systems used by Revolve NTNU. An on-line parameter estimation estimation scheme using the gradient approach method [8, 22], is suggested as a way of finding these parameters. The advantage of this method is that it is fairly simple to implement, and does not rely on extensive knowledge of the suspension and tires used, tire temperature measurements and road surface condition, and other input required for data-driven tire modelling. The aim of the algorithm is to find the set of parameters that minimizes the fitness functions

$$\begin{aligned} f_x &= \left|\sum F_x - ma_x\right| \\ f_y &= \left|\sum F_y - ma_y\right| \end{aligned} \tag{4.20}$$

over time. At each iteration, the model parameters are marginally changed to improve eq. (4.20).

Because the fitness functions are highly nonlinear, convergence to a global optimum can not be guaranteed. This can be remedied by selecting good initial parameters, or by running several models in parallel, and select the one which gives the best fitness. Eventually, a library can be built by saving a set of parameters and tagging it with the conditions of the test, i.e. "Dry asphalt, 18 degrees, cold slick tires in new condition". Once a complete library has been built this way, operators can select a set of parameters with a tag matching the current conditions, which will be close to the optimal set. The tire model is only valid for steady-state slip, when the tire has deformed to its steady-state shape. This effect is significant in the lateral dynamics, but not in the longitudinal dynamics, which are much quicker. To catch the transient behaviour of the tire, a relaxation distance, σ, can be used [19]. The lateral forces are modelled as following:

$$\dot{\hat{F}}_y^{ij} = \frac{U}{\sigma}(F_{y,m}^{ij} - \hat{F}_y^{ij}) \tag{4.21}$$

In the nonlinear tire model parameter estimation, coupled forces are taken into account. That gives the following terms for the fitness functions, eq. (4.20):

$$\begin{aligned}\sum F_x =& F_{x,m}^{FL}\cos(\delta_L) + F_{x,m}^{FR}\cos(\delta_R) - \hat{F}_y^{FL}\sin(\delta_L) + \hat{F}_y^{FR}\sin(\delta_R)\\ &+ F_{x,m}^{RL} + F_{x,m}^{RR} - \frac{1}{2}\rho C_d A u^2\\ \sum F_y =& \hat{F}_y^{FL}\cos(\delta_L) + \hat{F}_y^{FR}\cos(\delta_R) + F_{x,m}^{FL}\sin(\delta_L) + F_{x,m}^{FR}\sin(\delta_R)\\ &+ \hat{F}_y^{RL} + \hat{F}_y^{RR}\end{aligned} \tag{4.22}$$

Let $\mathbf{p} = \begin{bmatrix} B_x & C_x & \mu_x & E_x & B_y & C_y & \mu_y & E_y \end{bmatrix}$ be the parameter vector. The Jacobian for the fitness functions with respect to the parameter vector is given by

$$\mathbf{J}_m = \begin{bmatrix} \frac{\partial f_x}{\partial \mathbf{p}} \\ \frac{\partial f_y}{\partial \mathbf{p}} \end{bmatrix} \tag{4.23}$$

This is calculated offline using the MATLAB Symbolic Math Toolbox, and evaluated at each iteration. Finally, let
$\mathbf{K}_m = diag\big(\begin{bmatrix} K_{B_x} & K_{C_x} & K_{\mu_x} & K_{E_x} & K_{B_y} & K_{C_y} & K_{\mu_y} & K_{E_y} \end{bmatrix}\big)$ be the diagonal update gain matrix. The parameter update term is given by

$$\mathbf{p}_{k+1} = \mathbf{p}_k + \begin{bmatrix} f_x & f_y \end{bmatrix} \mathbf{J}_m \mathbf{K}_m \tag{4.24}$$

A method was tried in which different slip ranges were reserved for altering certain parameters, similar to what was done in the linear version. However, this gave, at best, similar results to eq. (4.24).

Single-track model parameter estimation

The single-track vehicle model, see fig. 4.7, is used in the design of high-level controllers. The high-performance high-level lateral controller described in section 4.3 uses the single-track vehicle model with a linear tire model, whereas [23] uses one with a nonlinear tire model.

Both controllers only use the lateral tire model. This is because $\alpha^{FL} \approx \alpha^{FR}$ and $\alpha^{RL} \approx \alpha^{RR}$, which makes the combined lateral force for each pair of wheels roughly fit the Magic Formula. But with four wheel independent drive, all four wheels may be spinning at different speeds, making a similar assumption for slip ratios highly inaccurate. Therefore, only the lateral tire model parameters are estimated.

Equation (4.8) is used to calculate the slip angles for the front and rear combined tires, and the term used in the fitness function is

$$\sum F_y = F_y^F \cos(\delta) + F_y^R, \tag{4.25}$$

where

$$\begin{aligned} F_y^F &= (F_z^{FL} + F_z^{FR})\mu_{s-t} sin(C_{s-t} tan^{-1}(B_{s-t}\alpha^F - E_{s-t}(B_{s-t}\alpha^F - tan^{-1}(B_{s-t}\alpha^F)))) \\ F_y^R &= (F_z^{RL} + F_z^{RR})\mu_{s-t} sin(C_{s-t} tan^{-1}(B_{s-t}\alpha^R - E_{s-t}(B_{s-t}\alpha^R - tan^{-1}(B_{s-t}\alpha^R)))) \end{aligned} \tag{4.26}$$

Other than that, the methods for estimating the linear and the nonlinear tire model parameters are the same as for the four-wheel model.

Chapter 5

Results and analysis

In this chapter, the testing results from the different systems implemented in this thesis are presented and discussed. Unfurtunately, due to the reallocation of manpower towards the main overall project goal described in section 2.3, no quantitative results were produced for the HPHLC. Nor was the interaction between the HPHLC and the TSCS tested.

5.1 MVP Trajectory Following at FSG

Due to problems passing the rigorous mechanical and electrical technical inspections, the team was not allowed to enter any of the dynamic events at the competition in Hungary, and only made it to the final event in Germany, which was the trackdrive event. At this event, the car ran smoothly for one lap, at a steady but quick walking pace, without knocking over any cones on the way. After one lap, the car stopped due to problems with the SLAM module, which caused the trajectory to end. The lap can be seen by scanning the barcode in fig. 5.1 with a smartphone or using this link:

`https://youtu.be/HDqbehdr0o4`

The car also drove autonomously during a successful EBS test. The track for the EBS test is 3.5 meters wide, and the car has to accelerate to a velocity exceeding 40 km/h within 20 meters. The EBS is then triggered, and the car must come to a controlled stop. The EBS test is failed if the car knocks over any cones. The car did this in a controlled manner, but unfortunately this was not documented.

This counts as a flawless execution by the MVP version of trajectory following, according to the requirements set. As the requirements were of a qualitative nature, rather than a quantitative, no quantitative analysis will be performed.

As a side note, since many other teams also had similar issues, the team placed 2^{nd} in Hungary and 7^{th} in Germany due to good performance in the static events. This was the top placement among first-year teams in both competitions, and Revolve NTNU was also the only first-year team to complete a lap in the trackdrive event.

Figure 5.1: Scan this QR code with a smartphone to view Eld driving the trackdrive event. Alternately, this link can be used: https://youtu.be/HDqbehdr0o4

5.2 Longitudinal force control

Figure 5.2 shows the reference tracking for the longitudinal force controller for a predefined force profile on a straight trajectory. The controller is able to follow the force profile with high precision, suggesting that it can be used as an interface for the high-level controllers.

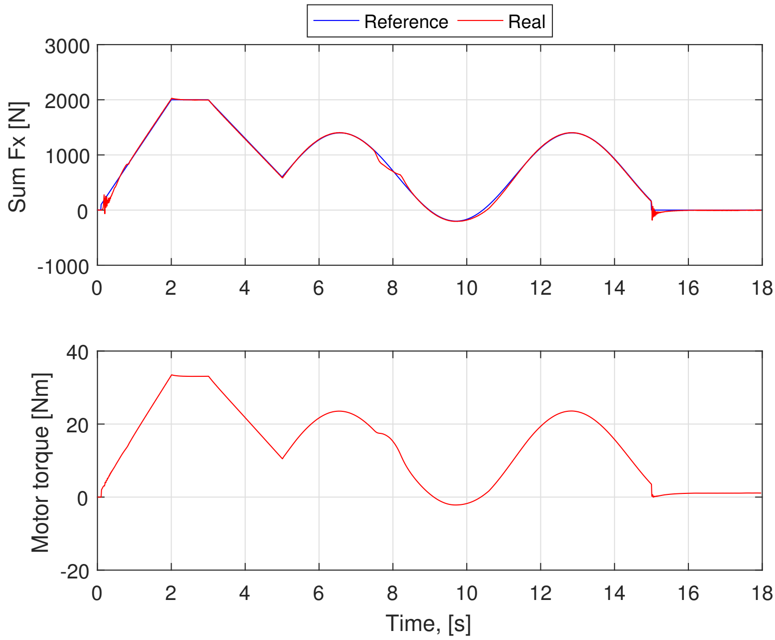

Figure 5.2: Reference tracking for longitudinal force controller for a predefined force profile.

5.3 Tire-ground force and friction estimation using nonlinear Pacejka parameter estimation

The results from a simulation run in IPG CarMaker are shown in section 5.3 and section 5.3. IPG CarMaker uses an automated driver algorithm, which has a much less agressive driving style than Revolve NTNU's drivers. This means that the simulated vehicle spends much less time in the high slip conditions required to get the correct shape for the nonlinear parts of the tire model, than the physical vehicle would. It can therefore be expected that the algorithm has better convergence on the physical vehicle than on the simulation.

Looking at fig. 5.3 and fig. 5.4 we see that the coefficients of friction, μ_x and μ_y, converge to a rough estimate quite quickly. The other parameters seem to have little or no convergence, but by looking at fig. 5.5 and fig. 5.6, we can see that the tire model curves do indeed approach a good fit. This is probably due to the strong nonlinearity of the tire model, which cause coupling effects between the different parameters. This way, the curves can be quite similar even though the parameters are quite different.

Figure 5.7 and fig. 5.8 show that the fitness function does not really converge to zero, but fluctuates in the near vicinity. This can be explained by looking at the real normalized tire forces depicted by red o's in fig. 5.9 and fig. 5.10. This suggests that the model is imperfect, and is unable to account for certain dynamics. For the lateral dynamics, this is in part because the model assumes a steady-state slip condition, which is handled by using

-	RMS error	Peak error	Peak value	RMS error % of peak value	Peak error % of peak value
F_x^{FL}	23.71	195.25	1788.10	1.3	10.9
$\sum F_x$	43.31	68.27	5917.95	0.7	1.2
F_y^{FL}	184.81	1417.90	2035.89	9.1	69.6
$\sum F_y$	107.51	580.68	4673.56	2.3	12.4

Table 5.1: Error characteristics of tire force estimation

a relaxation model, eq. (4.21). This is not evident in fig. 5.10, but is more clearly shown in fig. 5.12 in that the force estimates are much better for high-slip conditions than fig. 5.10 would suggest.

Figures fig. 5.11 and fig. 5.12 show the estimated forces vs. the real forces for each tire. From the figures we can see that the estimate is quite good for both longitudinal and lateral forces. From fig. 5.11 we can also clearly see that the new longitudinal force estimate is much better than the old. This is not applicable for the lateral force estimate, as no estimator existed before.

Table 5.1 further shows that the longitudinal force estimate is much better than the lateral one. This makes sense when comparing the model fit fig. 5.9 and fig. 5.10. Note that the peak value in F_y^{FL} is caused by the spike in slip angle at t=23, when the vehicle almost swerved out of control.

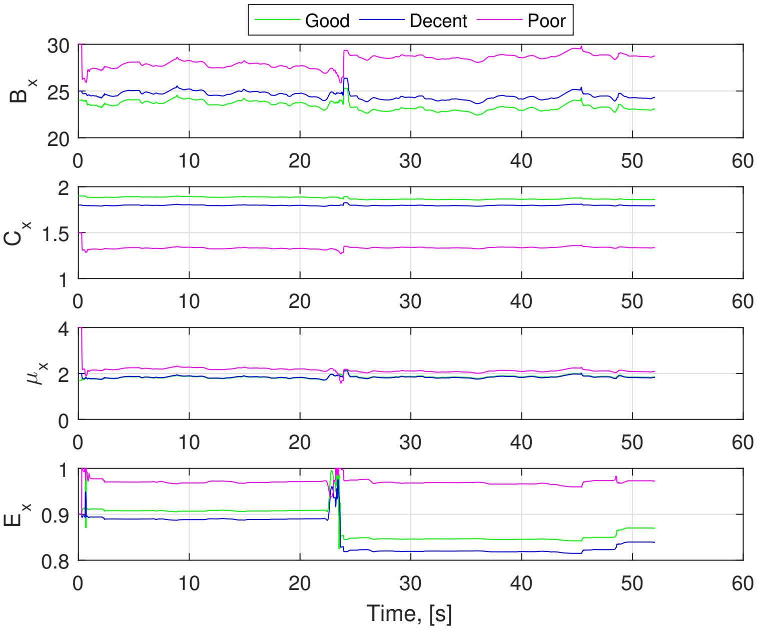

Figure 5.3: Longitudinal tire model parameters over time, with good, decent and poor starting conditions.

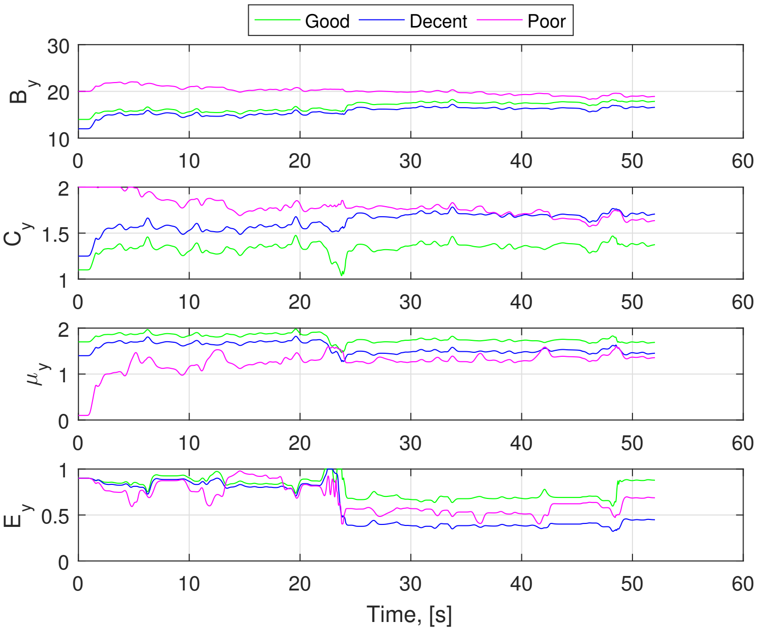

Figure 5.4: lateral tire model parameters over time, with good, decent and poor starting conditions.

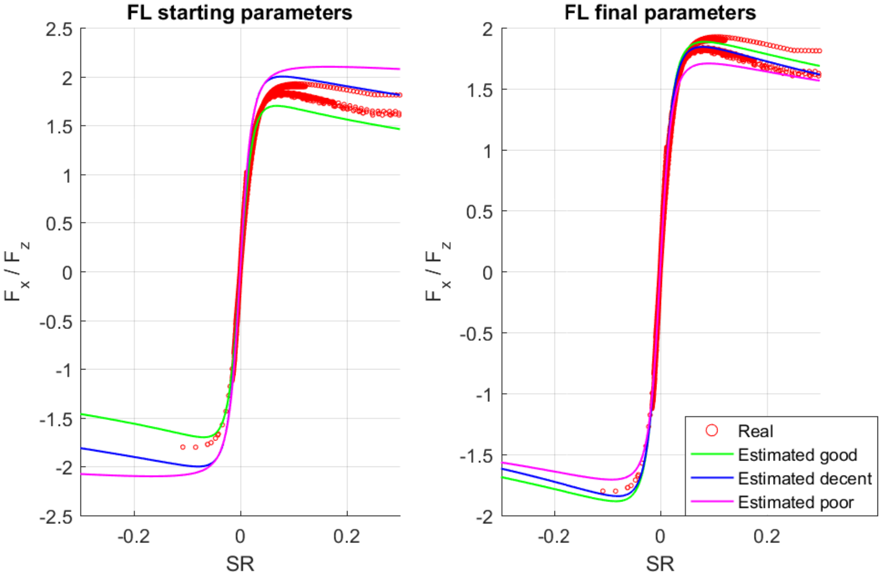

Figure 5.5: Longitudinal tire model at start and end of simulation for front left tire, with good, decent and poor starting conditions.

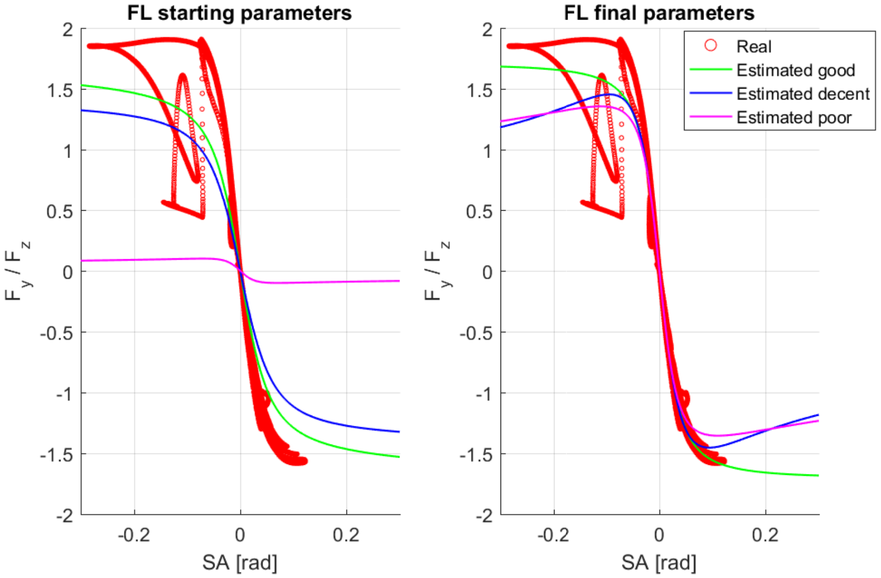

Figure 5.6: Lateral tire model at start and end of simulation for front left tire, with good, decent and poor starting conditions.

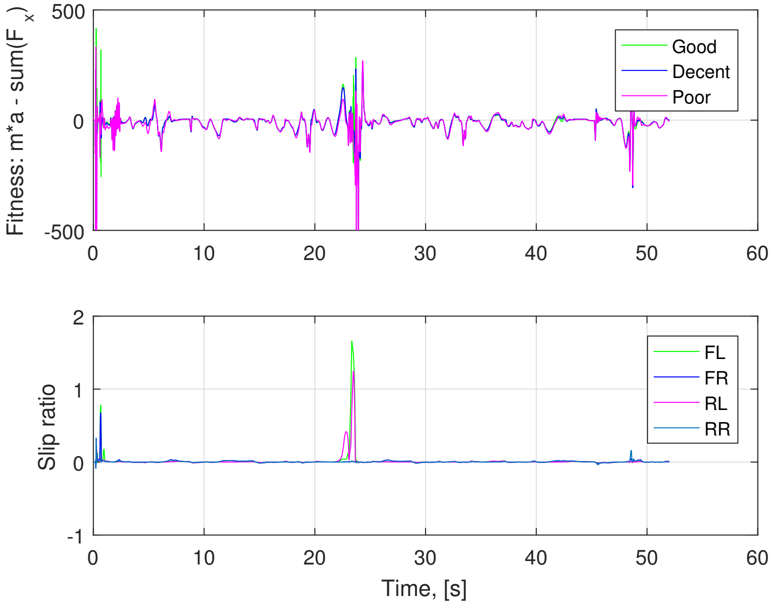

Figure 5.7: Top: Fitness of the longitudinal tire model over time, with good, decent and poor starting conditions. Bottom: Slip ratio over time for each tire.

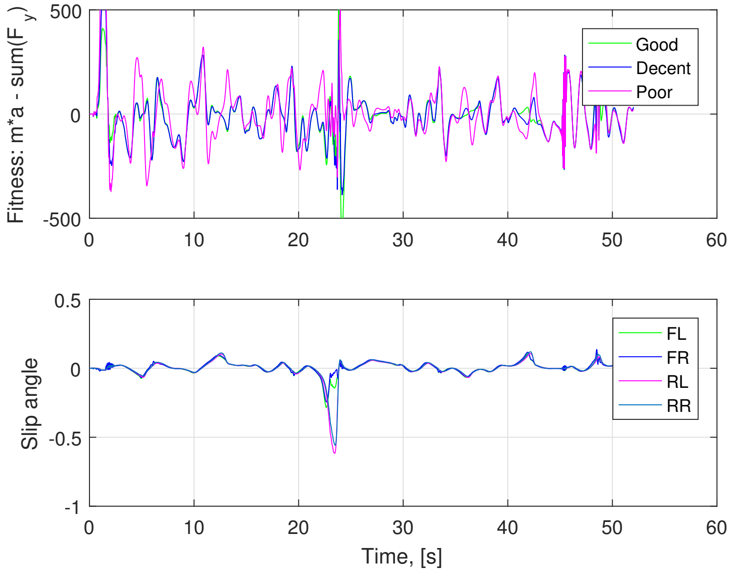

Figure 5.8: Top: Fitness of the lateral tire model over time, with good, decent and poor starting conditions. Bottom: Slip angle over time for each tire.

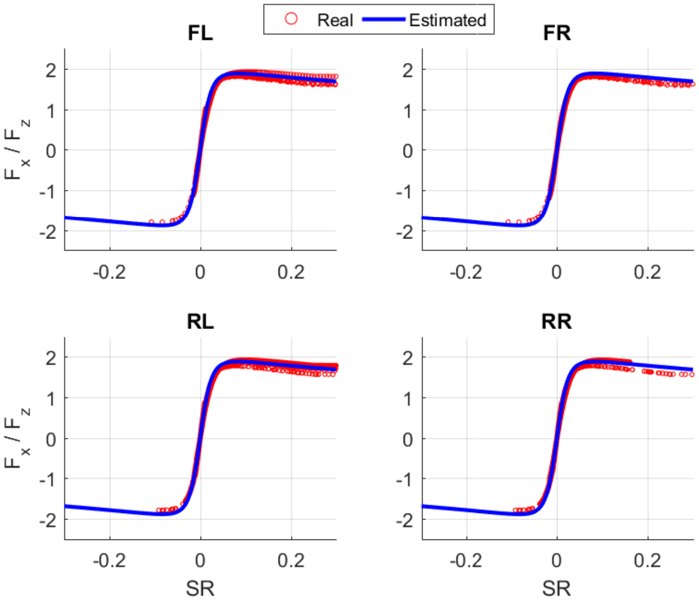

Figure 5.9: Longitudinal tire model vs. normalized tire forces.

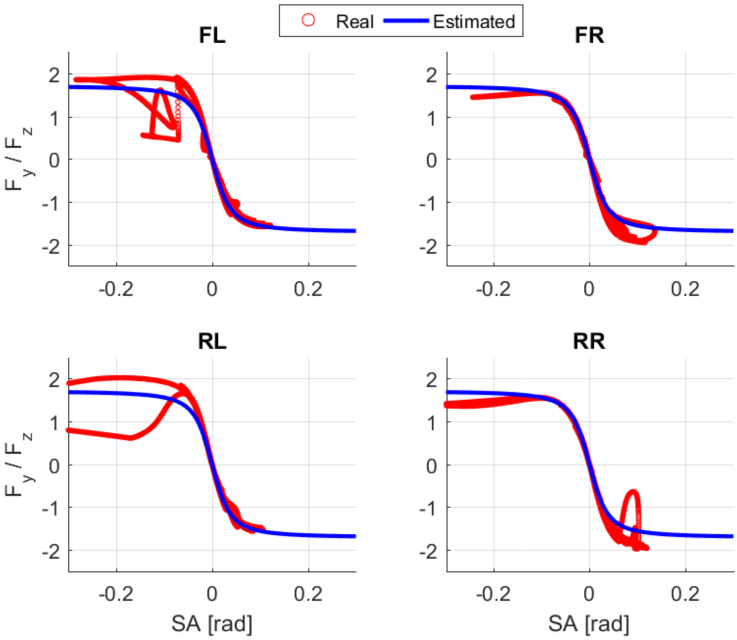

Figure 5.10: lateral tire model vs. normalized tire forces.

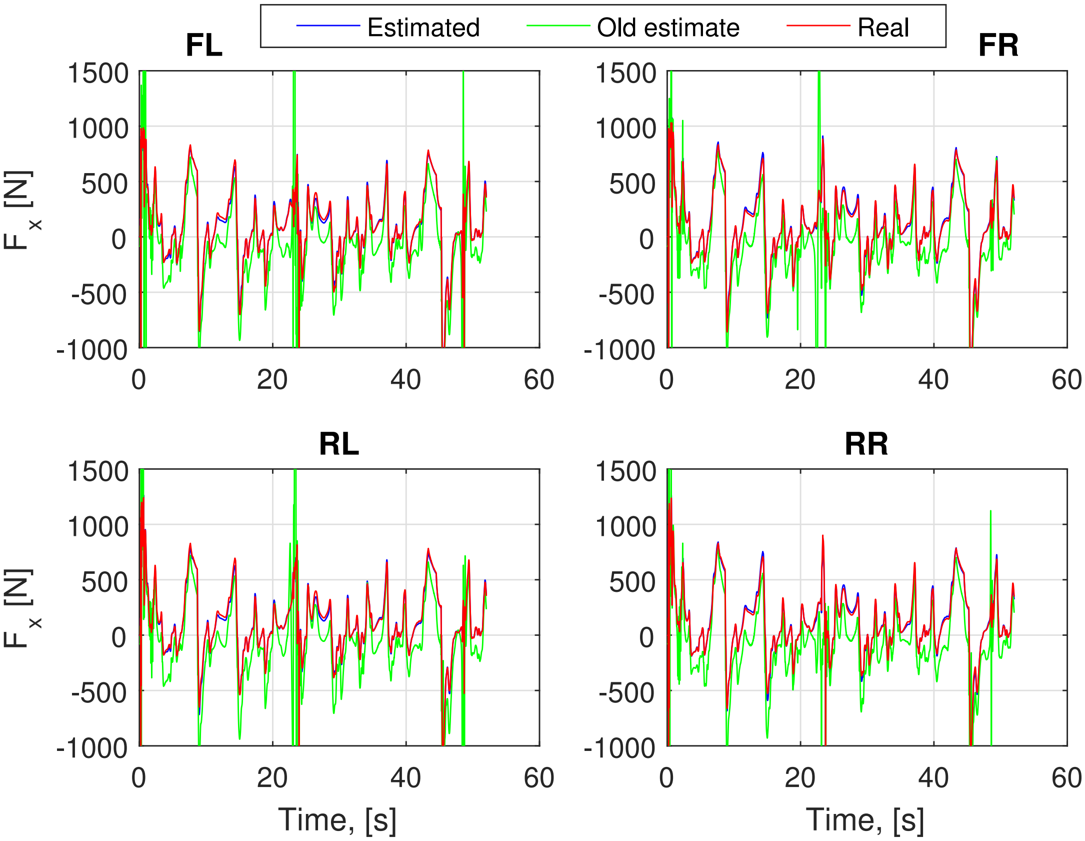

Figure 5.11: Estimated vs real (simulated) longitudinal tire forces for each wheel, from the tire model with good starting parameters.

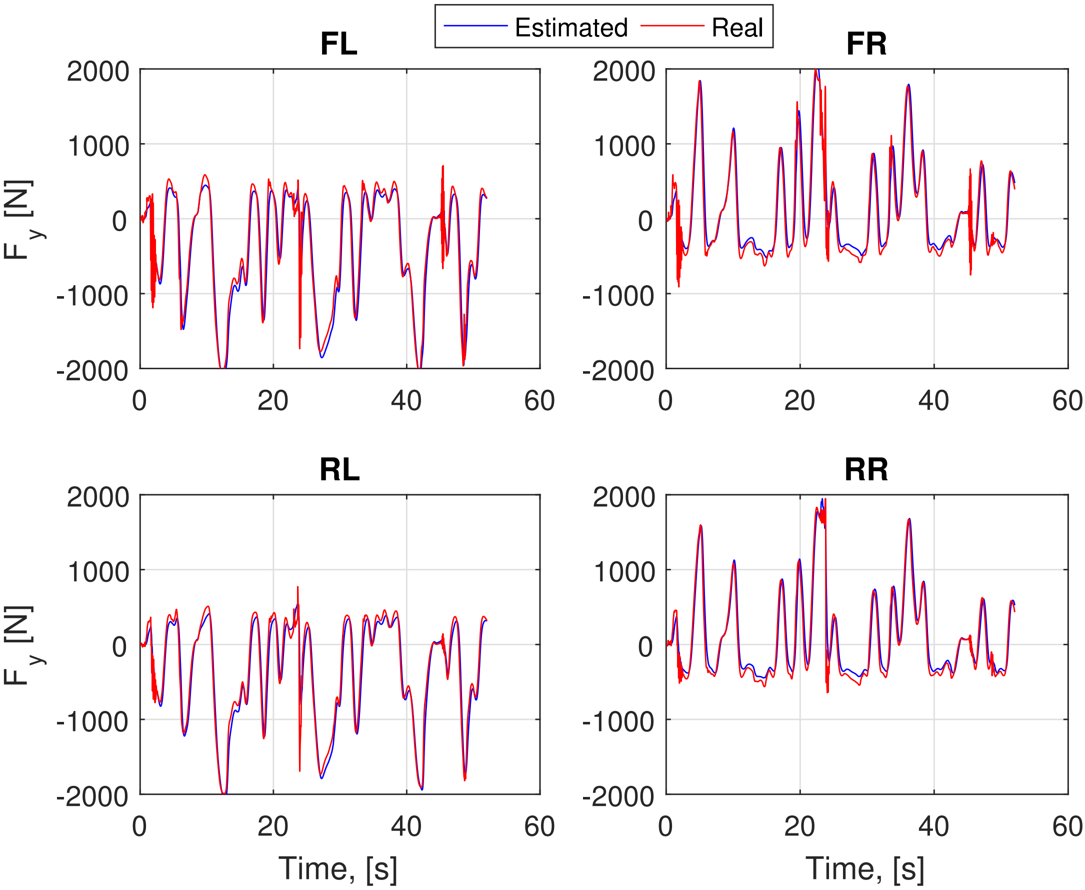

Figure 5.12: Estimated vs real (simulated) lateral tire forces for each wheel, from the tire model with good starting parameters.

5.3.1 Using estimated values for SR, SA and Fz

The results from section 5.3 are made using ground-truth values for SR, SA and Fz. This is done in order to isolate the parameter estimation problem. However, on the physical vehicle, these quantities are not readily available, and must be estimated. From figs. 5.13 to 5.15 we can see that the estimates for F_z and SA are quite good, but the estimates for SR are not. This results in a good fit for the lateral tire model, see fig. 5.18, and good lateral force estimates, see fig. 5.19. But the longitudinal tire model, fig. 5.16, and therefore the longitudinal force estimates, fig. 5.17, are completely useless.

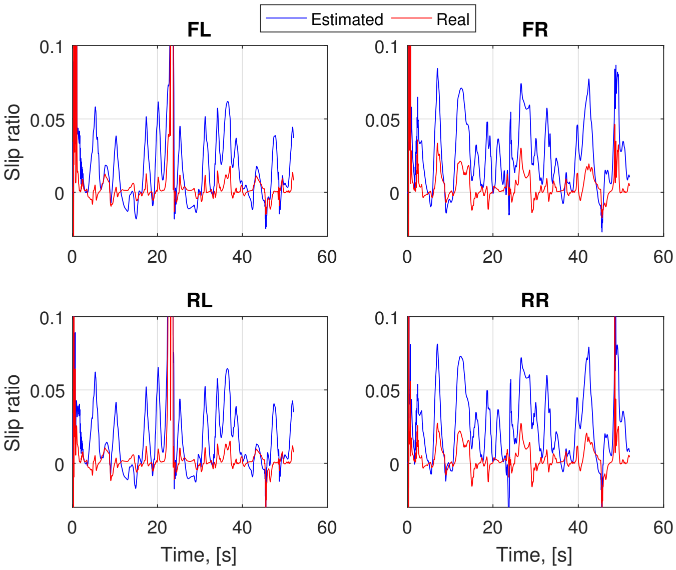

Figure 5.13: Estimated vs real slip ratios

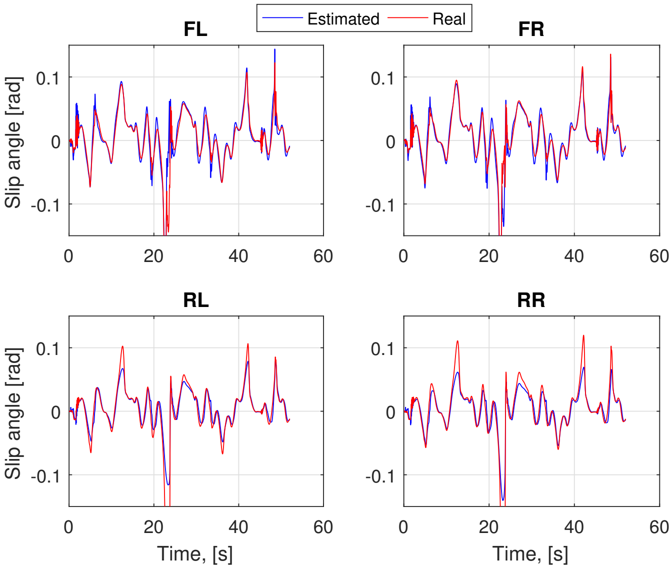

Figure 5.14: Estimated vs real slip angles

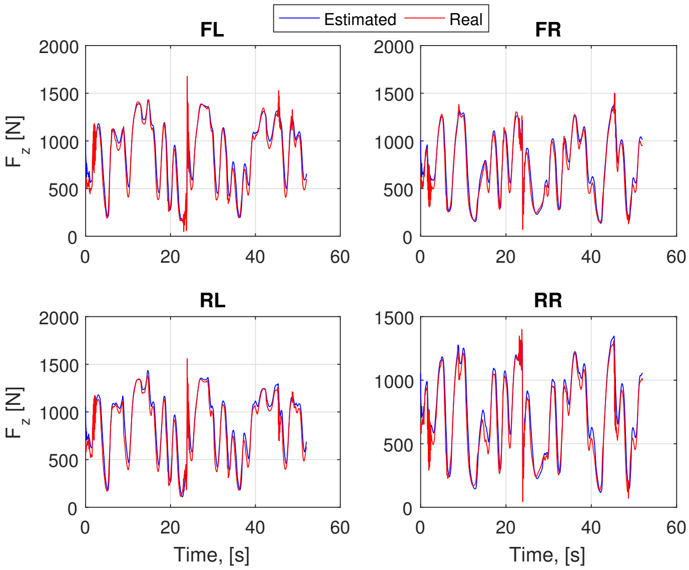

Figure 5.15: Estimated vs real normal tire forces

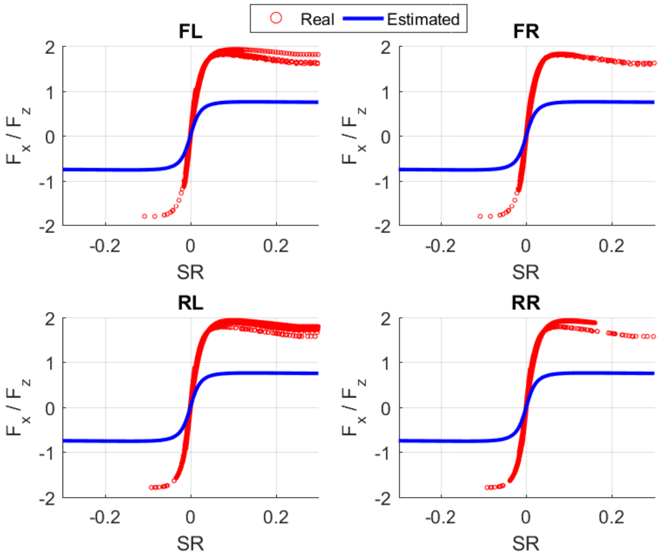

Figure 5.16: Longitudinal tire model vs. normalized tire forces using estimated input values.

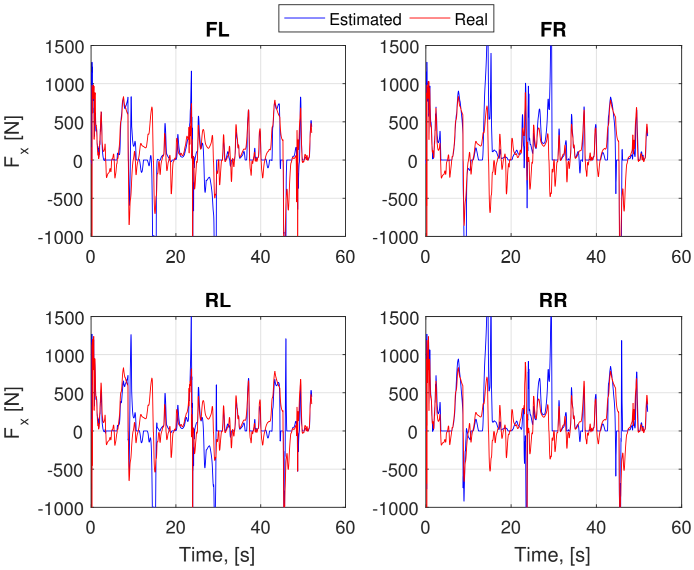

Figure 5.17: Longitudinal tire model vs. normalized tire forces using estimated input values.

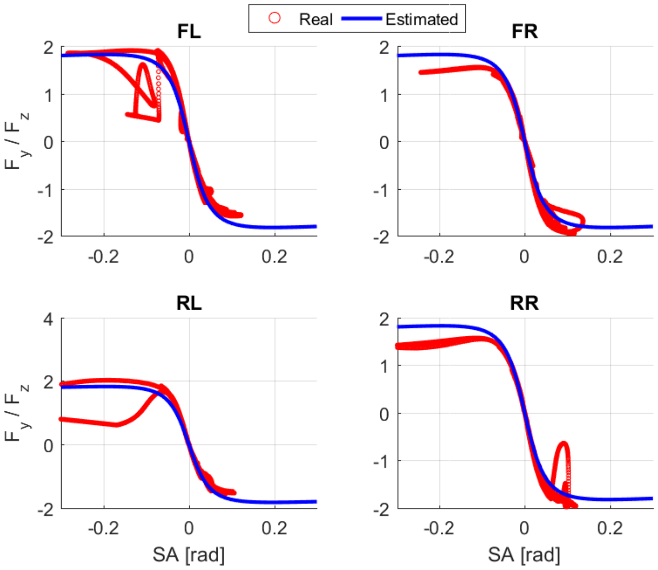

Figure 5.18: Lateral tire model vs. normalized tire forces using estimated input values.

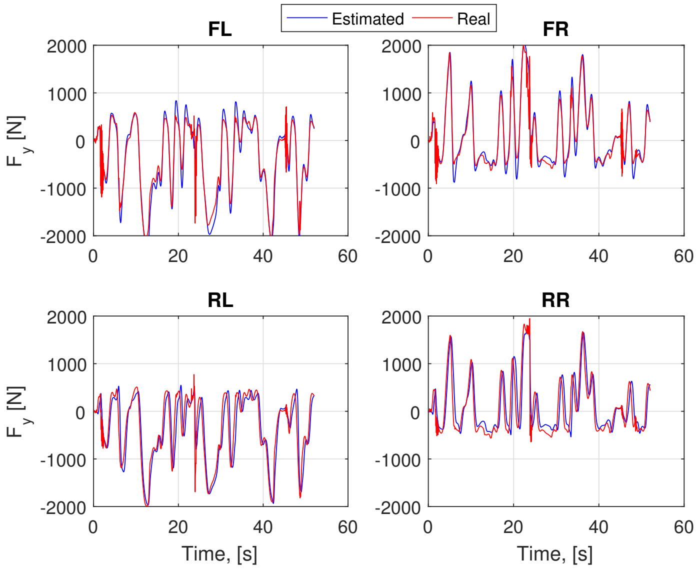

Figure 5.19: Lateral tire model vs. normalized tire forces using estimated input values.

5.4 Pacejka tire model parameter estimation for single-track vehicle model used in control algorithms

The algorithm for determining the parameters for the normalized Pacejka tire model for the single-track vehicle model is the same as for the four-wheel model. Therefore, only the final results are discussed. Figure 5.20 shows the normalized combined lateral forces for the front and rear tires. We can clearly see that the assumption that the same tire model can be used for both combined tires is not completely valid. Further, the normalized forces for each of the combined tires follow a much less uniform curve than those of the individual tires in section 5.3, which automatically makes any tire model curve a worse fit.

The algorithm finds the model that makes the best compromise in terms of the cost function. Figure 5.21 shows that the force estimate is quite good for the combined front tires, but not so good for the combined rear tires. Even though the graphs look pretty closely matched, the values are often quite different at any given time. This impacts the yaw acceleration estimate in much the same way, and we can see that, while the yaw acceleration estimated by the single-track model follows the same general curve as the true value, it is nowhere near perfect.

Wether of not the single-track model is sufficiently accurate to be used in the two high-performance high-level controllers remains to be worked out, as neither controllers have been tested in a high-fidelity simulator or on the physical car.

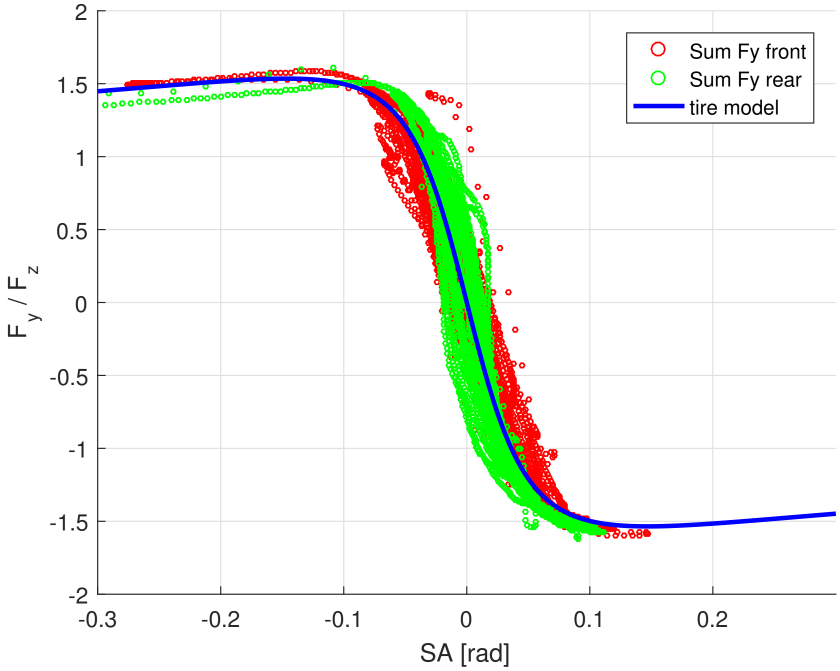

Figure 5.20: Pacejka tire model vs. normalized front and rear combined tire forces.

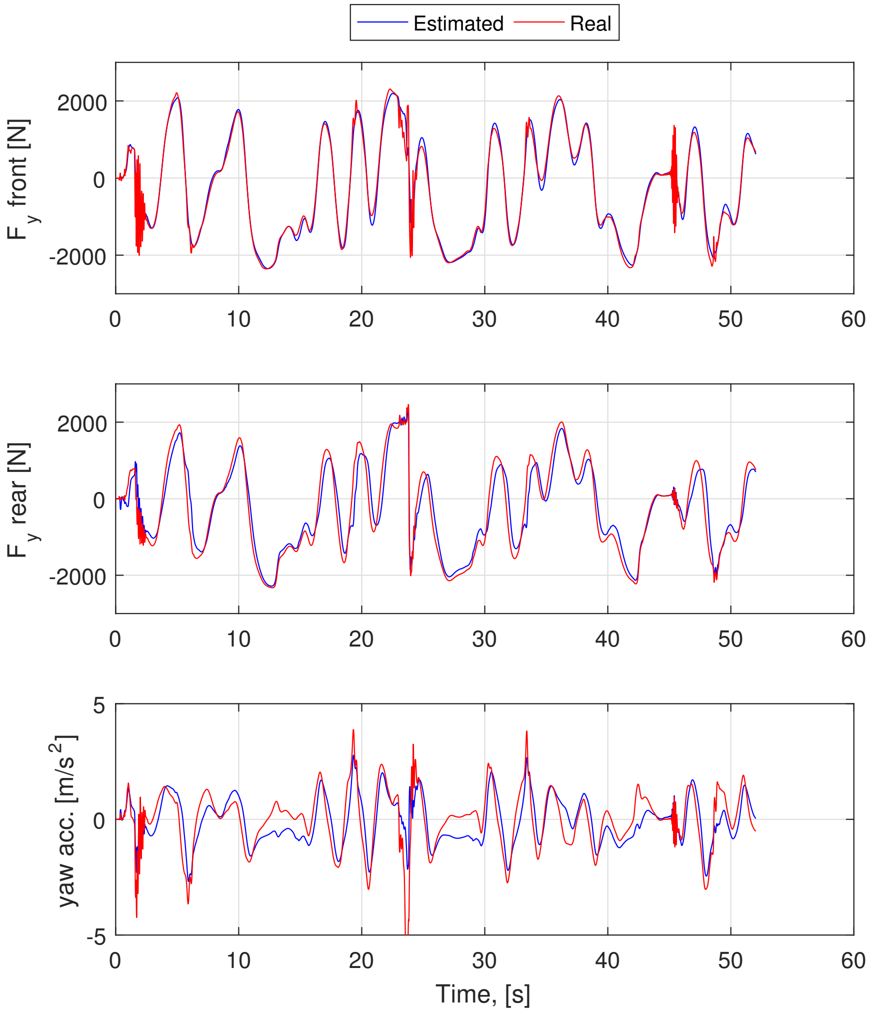

Figure 5.21: Front (top) and rear (middle) combined tire forces, true vs. estimated using single-track model. Yaw acceleration calculated from single-track model vs. real (bottom)

Chapter 6

Conclusion and Further work

The MVP version of trajectory following, which was covered by the first two sub-objectives for this thesis, performed flawlessly during the competition, as seen in fig. 5.1. It was, in fact the most robust module in the autonomous driving pipeline throughout the summer.

The MVP was the only system developed in this thesis which made it onto Eld this season. The remaining three sub-objectives were implemented in software, which counts as a success for the thesis objectives. The author has limited control of the overall project, and the high-performance systems were only implemented for the eventuality that the upstream modules would be robust at higher speeds, which was not the case

The HPHLC was implemented in ROS, and qualitatively tested in Gazebo. The TSCS was modified to accept a longitudinal force request as well as a yaw rate request, which was the common downstream interface for the HPHLC and the MPCC module.

The Pacejka tire model parameter estimation algorithm was implemented in MATLAB/Simulink. It provided quite good tire-road force and friction estimation in both the lateral and longitudinal direction, given good values for F_z, SR and SA. The estimated values for F_z and SA are sufficiently accurate, but the SR estimate is not, which makes the longitudinal estimates very bad. Thus, this sub-objective was only partly successful.

6.0.1 Further work

For the next season, a good SR estimator should be implemented. This is important for several sub-modules in the TSCS, not only the tire model parameter estimation algorithm.

When deemed sufficiently accurate, the tire-road force and friction estimates should be used in the torque vectoring algorithm, to allow the vehicle to perform near the grip limit. The next logical step is to make a complete data-driven tire model. This requires much more effort, and additional sensors, than the parameter estimation algorithm, but should give better results. The new tire model can be benchmarked against the old.

Qualitative testing of the HPHLC and the MPCC module, and/or any other high-level controllers implemented, should be carried out in a high-fidelity simulator, in isolation as well as in sequence with the TSCS, before being implemented on the Team 2019 driverless vehicle.

Bibliography

[1] J.-A. Serret, “Sur quelques formules relatives à la théorie des courbes à double courbure.” *Journal de Mathématiques pures et appliquées*, pp. 193–207, 1851.

[2] F. Frenet, “Sur les courbes a double courbure.” *Journal de Mathématiques pures et appliquées*, pp. 437–447, 1852.

[3] F. S. Germany, “Formula student rules 2018, version: 1.1, rev-541.”

[4] D. Mazur, “Formula student - world ranking lists,” online; accessed 23-May-2018. [Online]. Available: https://mazur-events.de/fs-world/

[5] Vectornav, “Industrial series product brochure,” online; accessed 9-August-2018. [Online]. Available: https://www.vectornav.com/docs/default-source/product-brochures/industrial-series-product-brochure-(12-0009).pdf

[6] B. W. Omholt, “Direct yaw moment, traction and powerlimit control of a four wheelindependent drive electric (4wid-ev)formula student race car,” Master’s thesis, Norwegian University of Science and Technology, 2016.

[7] O. Øren, “Optimal direct yaw control for four wheel independent drive electric vehicle,” Master’s thesis, Norwegian University of Science and Technology, 2017.

[8] J. Nocedal and S. J. Wright, “Numerical optimization 2nd,” 2006.

[9] O. S. R. Foundation, “Ros,” online; accessed 8-August-2018. [Online]. Available: http://www.ros.org/

[10] ——, “Gazebo,” online; accessed 8-August-2018. [Online]. Available: http://gazebosim.org/

[11] I. A. GmbH, “Ipg-automotive,” online; accessed 8-August-2018. [Online]. Available: https://ipg-automotive.com/

[12] T. I. Fossen, *Handbook of marine craft hydrodynamics and motion control.* John Wiley & Sons, 2011.

[13] S. N. Midtskogen, "Evaluation of different controlal gorithms for implementation in revolve ntnus 2018 driverless vehicle," 2018, unpublished.

[14] J. M. Haaland, "Vehicle dynamics simulation forsetting up a four-wheelindependent drive electric racecar," Master's thesis, Norwegian University of Science and Technology, 2017.

[15] W. F. Milliken, D. L. Milliken *et al.*, *Race car vehicle dynamics*. Society of Automotive Engineers Warrendale, 1995, vol. 400.

[16] A. Turnip, K.-S. Hong *et al.*, "Detection of critical driving situations based on wheel-ground contact normal forces," in *ICCAS-SICE, 2009*. IEEE, 2009, pp. 2519–2524.

[17] L. R. Ray, "Nonlinear state and tire force estimation for advanced vehicle control," *IEEE Transactions on Control Systems Technology*, vol. 3, no. 1, pp. 117–124, 1995.

[18] R. G. Brown, P. Y. Hwang *et al.*, *Introduction to random signals and applied Kalman filtering*. Wiley New York, 1992, vol. 3.

[19] X. Jin and G. Yin, "Estimation of lateral tire–road forces and sideslip angle for electric vehicles using interacting multiple model filter approach," *Journal of the Franklin Institute*, vol. 352, no. 2, pp. 686–707, 2015.

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

[21] Mathworks, "Tire-road interaction (magic formula)," online; accessed 9-August-2018. [Online]. Available: https://se.mathworks.com/help/physmod/sdl/ref/tireroadinteractionmagicformula.html

[22] I. D. Landau, R. Lozano, M. M'Saad, and A. Karimi, *Adaptive control*. Springer New York, 1998, vol. 51.

[23] Ø. Hovind, "Model predictive contouring control for formula student racecar," Master's thesis, Norwegian University of Science and Technology, 2018.