NTNU

Norwegian University of
Science and Technology

Direct Yaw Moment, Traction and Power Limit Control of a Four Wheel Independent Drive Electric (4WID-EV) Formula Student Race Car

Bjørn William Omholt

Master of Science in Cybernetics and Robotics
Submission date: August 2016
Supervisor: Tor Arne Johansen, ITK

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Abstract

This master's thesis describes the development, implementation and testing of a suite of control systems consisting of torque vectoring, traction control and power limit control for a Formula style race car using four electric motors to achieve asynchronous power output to each wheel. The control systems have been designed in Matlab Simulink with the help of a parameterized full car model from IPG CarMaker. Code generation in Matlab Simulink was used to generate C-code, which was exported to a self written embedded C-code project which was programmed onto an Atmel ARM 32-bits microcontroller. Hardware-in-the-loop simulations were then performed to verify that the embedded implementation of the control systems worked as expected. Finally, the control systems were put to the test in the car at Vaernes Airport and at three Formula Student competitions throughout Europe. The control systems have worked as expected, and all the drivers praised the performance of the car.

Sammendrag

Denne masteroppgaven beskriver utvikling, implementasjon og testing av en pakke kontrollsystemer bestående av torque vectoring, traction control og effektkontroll for en Formula type racerbil med fire elekriske motorer for å oppnå asynkron kraftforsyning til hvert hjul. Kontrollsystemene ble designet i Matlab Simulink med hjelp av en parameteriserbar bilmodell fra IPG CarMaker. Kodegenerering i Matlab Simulink ble brukt for å oversette kontrollsystemene til C-kode, som ble eksportert til et selvskrevet C-kode prosjekt som deretter ble lastet opp på en Atmel ARM 32-bits mikrokontroller. Hardware-in-the-loop simuleringer ble deretter gjort for å bekrefte at den innebygde implementasjonen av systemet fungerte som planlagt. Til slutt ble kontrollsystemene satt på prøve gjennom testkjøring på Værnes lufthavn og på tre Formula Student konkurranser i Europa. Kontrollsystemene har fungert som forventet, og sjåførene har alle gitt veldig positive tilbakemeldinger på bilens manøvreringsevne.

Contents

1 Introduction

1.1 Formula Student

Formula Student is a global educational motor sport competition format, which aims to inspire and develop enterprising and innovative engineering students. University students world-wide are challenged to design and build an open-cockpit, single-seat and open-wheel formula style racing car in order to compete in static and dynamic events. Through this the Formula Student concept assists universities in producing more complete engineers by providing highly motivating and very difficult practical challenges that have to be solved in a team context, something which is not included in most engineering educations. The competitions include the following events.

Business Presentation
Two presenters must pitch a business idea based on the premise of manufacturing 1000 units of the car each year.

Cost
Both the manufacturing method and cost of every part in car the car must be documented in a cost document. The cost event involves an assessment of this document, as well as a test of the team's knowledge of manufacturing processes and associated sustainability issues.

Design
In front of a jury consisting of world class engineers each team has to give a 40 minutes presentation focused on explaining the theory and rationale behind the design and development of the car.

Acceleration
This event is a 75 meter run to demonstrate the acceleration capabilities of the car.

Skid pad
The steady state cornering performance of the car is tested in this event. This is done by taking the average time of driving the car in a full right hand circle and then a full left hand circle.

Autocross
Here, one is asked to finish one lap on a technical track in order to test both the per-

formance of the car and the skills of the driver.

Endurance

The endurance race is a 22 km race with a driver change at 11 km. The sheer length of this race tests the reliability of the car, as well as the fuel management and race strategy of the team. Finishing the endurance race is required for doing well in the overall ranking.

Efficiency

Teams are given an efficiency score based on a combination of energy consumption and track pace in the endurance event. Track pace is included in the calculation to not undermine the racing focus of the competition.

The maximum scoring for each event is shown in Figure 1.

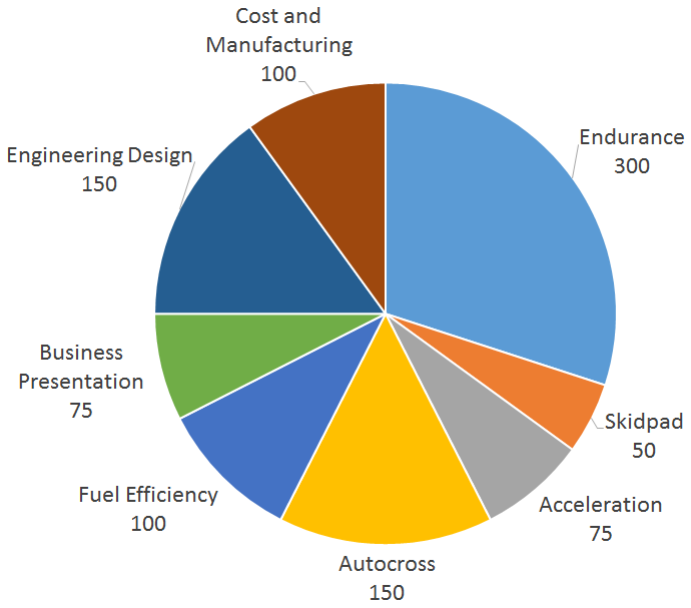


Figure 1: Scoring distribution for the events.

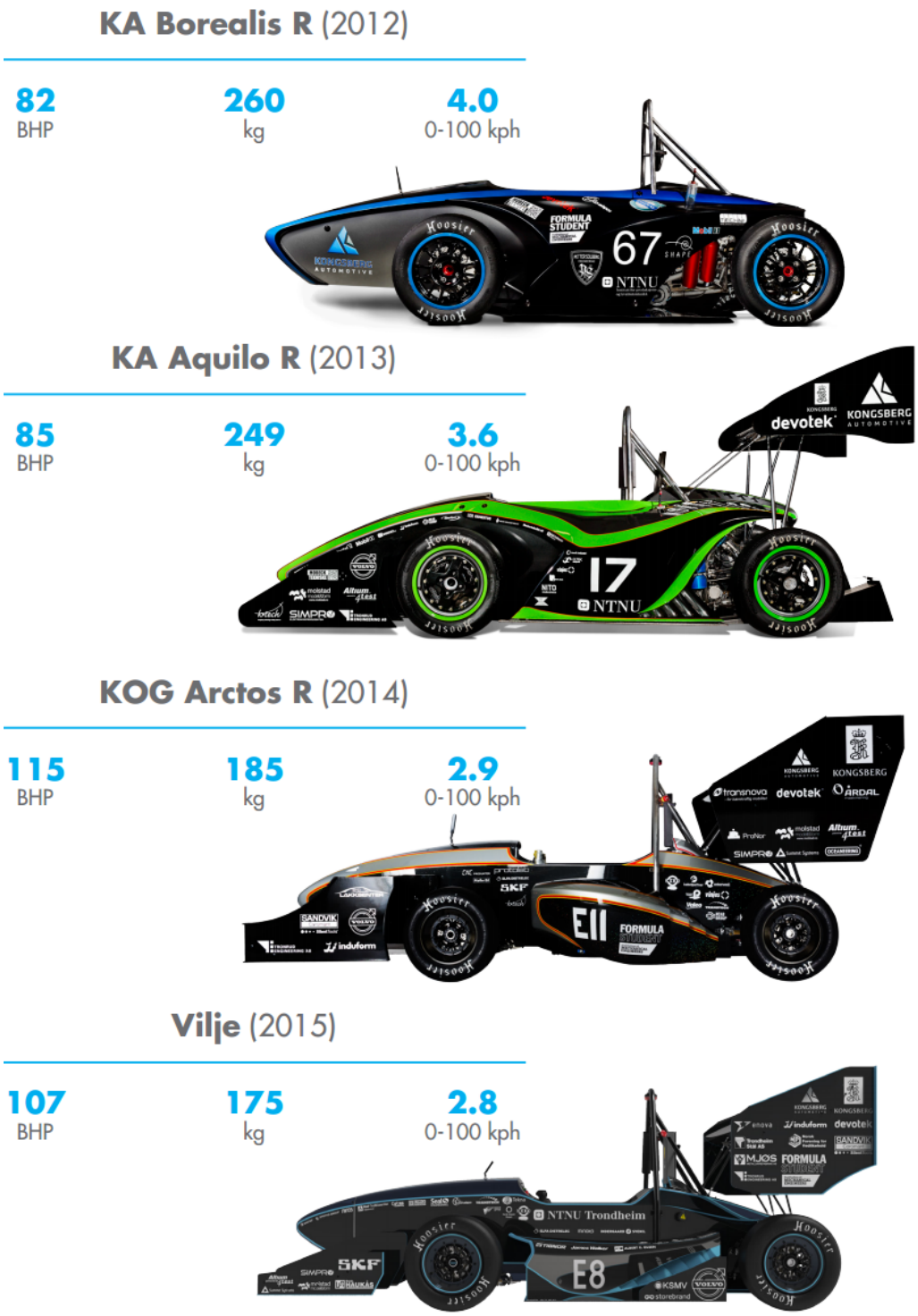


Figure 2: The Revolve cars, kindly provided by Peder Aune.

1.2 Revolve NTNU

Revolve NTNU is an independent organization controlled and operated by students at NTNU in Trondheim. Revolve was founded in 2010 by four mechanical engineering students who wanted to bridge the gap between theory and practice in an exciting way. The first team consisted of 23 members who produced the first Revolve car, KA Borealis R, which was finished in 2012. Since 2012, Revolve has produced a race car every year (Figure 2).

Robust and relatively simple design choices were made for Borealis to ensure that the vehicle would be able to accomplish the endurance test. Borealis had a combustion engine and a steel tube chassis. Borealis was described as the best first year car ever in the history of the competition by the Chief Design Judge, Pat Clarke. The Revolve 2013 team consisted of 46 members who made another combustion vehicle, KA Aquilo R, which did very well for a second year car. In 2014 however, Revolve decided to go all in and make Norway's first electric race car. In addition to tackling the challenges of designing an electric power train, the team made a full carbon fiber monocoque and carbon fiber rims. In 2015 Revolve made another electric vehicle, Vilje, which can be seen as a continuation of the main design choices of Arctos, with the exception of making an in-house developed motor controller (inverter).

1.3 Revolve 2016 Car: Gnist

This year (2016), however, Revolve NTNU made again a crucial redesign of the car by producing Scandinavia's first four-wheel-independent-drive electric race car (4WID-EV), shown in Figure 3. Permanent magnetic synchronous motors specifically designed for Formula Student by AMK Automatisierungstechnik are mounted in each wheel hub. Each motor weighs 3.7 kg, has an RPM range of 0 to 20000, and can produce a maximum torque of around 21 $\mathrm{N\,m}$. Gear boxes with a fixed gear ratio amplify the torque output of each motor by 15.5, giving Gnist a total maximum torque output of 1310 $\mathrm{N\,m}$ and a top speed of 110 $\mathrm{km\,h^{-1}}$. High quality sensors provide the control systems in Gnist with information about the state the vehicle is in. These systems aid the driver in controlling the driving and braking torque of each motor in order to enhance cornering and straight line performances.

Figure 3: The Revolve 2016 car; Gnist.

1.4 The objectives of this thesis

An ideal race car would respond accurately and immediately to steering inputs. In reality, however, a typical vehicle's yaw rate response to steering input is both relatively slow and imprecise. The yaw rate is the angular velocity about a vertical axis with origin at the center of gravity of the vehicle. The response is relatively slow since the tires must deform before enough lateral force is generated to steer the vehicle, and it can overshoot and oscillate due to the nonlinear behavior of the tires and the suspension. The balance between agility and stability can be shifted by tuning the suspension of the vehicle, but in this case one must sacrifice one characteristic to improve the other. An emerging technique called torque vectoring aims to improve both of these characteristics simultaneously. This technique is especially suited for a 4WID-EV, as the basis of torque vectoring is to individually control the driving and braking torque of each motor. Torque vectoring can be defined as "Creating a difference in the braking or driving forces at each wheel to generate a yaw moment (torque) with the intention of controlling yaw rate" [30]. By actively controlling the yaw rate of the vehicle one can improve the yaw rate response to steering input drastically. Compared to a non-torque vectored vehicle the yaw rate response is faster, more precise, oscillates less, and is more linear all the way up to tire saturation (loss of grip).

4WID-EVs with torque vectoring systems were first seen in Formula Student competitions in 2011, and they have consistently taken the top positions in the electric class every year since. The 2015 FSE competitions yet again demonstrated very clearly that the four-wheel-independent-drive-electric-vehicle (4WID-EV) has emerged as the superior design concept to tackle the dynamic challenges in Formula Student competitions:

- The acceleration event was dominated by 4WID-EVs through their capacity to utilize the traction forces of all tires compared to just two for a front/rear wheel driven car,
- The endurance event was dominated by 4WID-EVs due to their capacity to regenerate more energy during the race compared to single and dual electric motor configurations, which allows the use of a smaller battery pack that still provides enough energy to drive hard the whole race,
- The autocross event was dominated by 4WID-EVs due to their superiority in dealing with the handling and acceleration demands imposed by the track layout,
- The skid pad event was dominated by 4WID-EVs due to their superior lateral control compared to traditional configurations.

To enable Revolve NTNU to compete successfully against the top teams it was therefore considered mandatory to make the shift from a rear-wheel driven car to a 4WID-EV. The overall objective of this thesis was to contribute to this shift by designing and implementing a control system package consisting of torque vectoring, traction control and power limit control.

To give the car a competitive edge I decided to take full advantage of the opportunity to achieve superior yaw moment control through asynchronous power output to each wheel. I therefore particularly focused on a method called Direct Yaw Moment control (DYC) that uses both driver inputs and vehicle states to generate a desired yaw moment. Traditionally, DYC has been used to keep a vehicle within its stability limits through the use of individually controlled friction brakes, and torque reduction. The added yaw moment generated by such systems comes at the cost of comfort and vehicle speed reduction, and these systems are therefore normally only activated in emergency situations where the yaw rate or the vehicle sideslip angle is beyond a threshold that is deemed safe. However, DYC is now evolving from being mainly a vehicle safety system to becoming a means to achieve smooth continuous operation that increases general comfort, vehicle handling, as well as safety. This

extended use is only possible on vehicles where torque to the wheels can be individually controlled, either through electromechanical torque vectoring differentials or individually actuated electric motors.

As DYC is used mainly to improve cornering performance, I also focused on implementing traction control to enhance the longitudinal performance of Gnist by preventing excessive wheel spin and wheel lock in connection with driving and braking maneuvers.

A comprehensive quantitative real-word test of the control systems developed in this thesis could only be performed if I had access to a fully operational car with a dedicated team of minimum six persons over several days. Even if all other team members in Revolve executed their tasks properly so that we got an operational car, I knew from experience that the time window from when the car appeared on the road till the last competition day would be be cramped with all sorts of practical challenges that would probably not allow the luxury of doing this comprehensive quantitative testing. I therefore considered the primary objectives of this thesis to be:

1. To provide a complete in silico implementation of the control systems verified through HIL simulations and ready to be tested in Gnist.
2. Assuming the availability of an operational car, to document successful tuning of the systems so that they in a normal competition setting performed close to what was to be expected from simulations.
3. To get a positive assessment of the car's performance by the drivers, and document that the control systems worked properly in the competitions Gnist was able to participate in.

2 Preliminaries

This thesis will be used as a starting point for further development by the Revolve 2017 team. Some background information about key topics is therefore included to ease communication.

2.1 Tire forces

Tires are the primary source of the forces and torques which provide control and stability of the vehicle. Applying a torque to the wheel spin axis results in longitudinal slip. The term slip ratio is used to quantify the amount of slip in relation to the vehicle

longitudinal speed. Many different slip ratio definitions exist, and it is common to use a definition suited for the application at hand. The slip between the tire and the surface and the tire normal force are the primary factors controlling the magnitude of the longitudinal tire force. The magnitude of the lateral tire force varies mainly with the slip angle (SA) (α), which is the angle between the lateral velocity and the longitudinal velocity of the contact centre of the tire. Figure 4 shows a plot of tire longitudinal force F_x against slip ratio for varying loads and slip angles, and it illustrates the relationship between the slip ratio and the amount of longitudinal force the tire exerts.

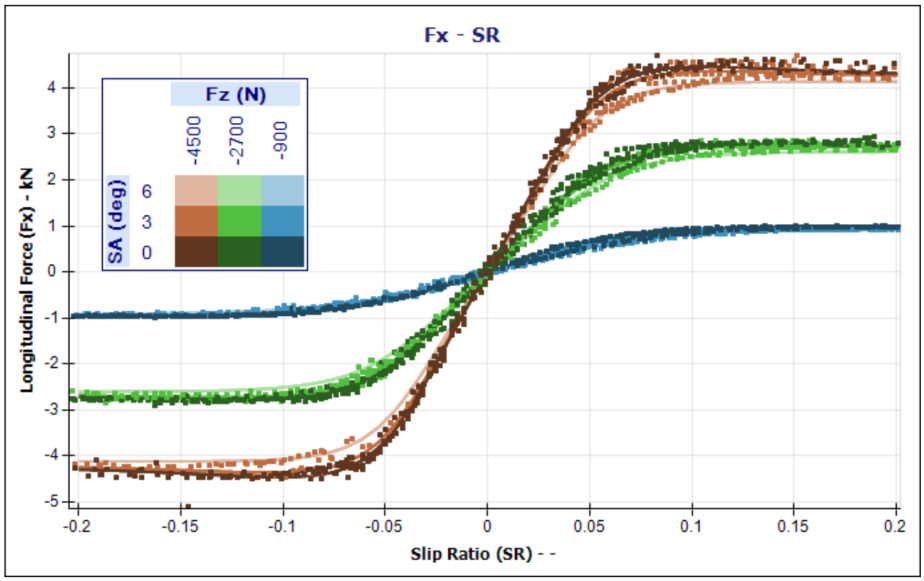


Figure 4: Longitudinal tire force F_x vs slip ratio for different loads (F_z) and slip angles (SA) based on experimental tire data. The figure is taken from the Optimum G documentation.

The combined generation of longitudinal and lateral tire forces is often represented as a friction ellipse, shown in Figure 5.

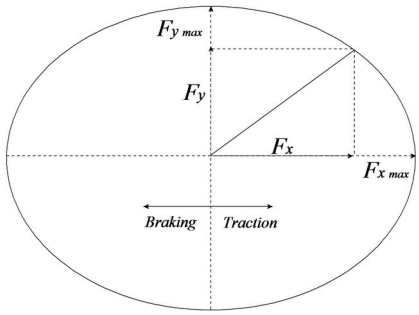


Figure 5: Friction ellipse for a tire [5].

The friction ellipse concept describes the force-limiting behavior for combined steering and driving (or braking) [27]. This means that the combined generation of forces, in longitudinal and lateral tire directions, is limited by a total friction capacity. Exceeding this total friction capacity is supposed to lead to slip. The friction ellipse concept is a simplification, however, as the shape of the capacity limit of a real tire will not be a perfect ellipse. Every tire will also have its own friction capacity shape.

2.2 Direct Yaw Moment Control

The main structure of 4WID-EV DYC systems proposed in the literature is shown in Figure 6.

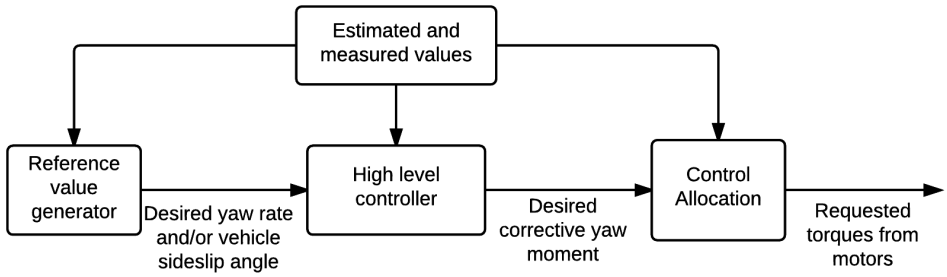


Figure 6: Overall structure of modern 2WID-EV and 4WID-EV DYC systems.

In modern DYC systems the reference value generator outputs both a reference yaw rate and a vehicle sideslip angle. The reference yaw rate is used to change the steering characteristic of the vehicle, while the reference vehicle sideslip angle is used to ensure vehicle stability. The vehicle sideslip angle is used due to its inverse proportional relationship with attainable yaw moment through steering wheel actuation [25]. Large sideslip angles thus cause the driver to not perceive any significant effect of turning the steering-wheel, which is why it must be kept at acceptable levels.

The high level controller is responsible for converting deviations from the reference values into a desired corrective yaw moment. In commercial vehicles parameters such as total mass and the distribution of mass can vary drastically due to actual passenger and baggage loads. Varying surface conditions will also induce large variations. Control algorithms that are robust to such variations are therefore explored in many research articles. The sliding mode control principle is especially popular in DYC research ([4], [28], [18]), but optimal control methods, such as composite nonlinear feedback are also employed [13]. Even though the general trend is a focus on

robust and optimal control methods, the simple PID controller has also been shown to be effective [8]. Feedforward control in addition to a feedback control method is also utilized in reference [8] with good results.

The control allocation method must convert the desired corrective yaw moment into requested motor torques. The individual torque requests must adhere to constraints such as motor saturation and tire adhesion, while the concerted actuation scheme can optimize energy efficiency and performance. Due to the multifaceted challenges caused by these constraints and goals, optimal control methods such as quadratic programming is a very popular method in DYC research ([11], [26]). Optimal control methods are computationally complex, however, and the much less complex weighted pseudoinverse control allocation method is therefore used in research that is more targeted towards realizable implementations on hardware with limited computing power [19].

2.3 Traction Control

Traction control is normally defined as a system that prevents excessive wheel slip during takeoff and acceleration by modulating torque at the wheels. An anti-lock braking system (ABS) on the other hand, is a system that prevents wheel lock during braking to achieve maximum deceleration while simultaneously maintaining vehicle stability. In a combustion vehicle this can only be done by actuating friction brakes at the wheels. For a 4WID-EV the situation is very different as every wheel can apply both driving and braking torque individually. The separation between traction control and ABS is therefore no longer relevant for this vehicle configuration, and instead a system that unifies wheel slip regulation for both driving and braking maneuvers can be implemented.

3 Design constraints

3.1 Processing power

The processor responsible for running the control systems in the car is an ARM Cortex M7 microprocessor that can run at 300MHz. Although the Cortex-M7 core is quite powerful in the microprocessor segment, its performance is dwarfed by desktops running multi-core CPUs at >3 GHz, and state of the art GPU based computing boards. The number of DYC-related algorithms that can actually run in a 100-150 Hz control loop on this hardware is therefore quite limited, implying that available processing power is one of the main factors deciding which algorithms to run.

3.2 Time considerations

Time is the most critical resource for Formula Student teams. The Revolve team has to conceptualize, design, produce, and test a race car in a time span of 8 months. The Revolve 2016 team planned for a test ready vehicle in early May 2016, which would give ample time for testing the vehicle before it was shipped to the first competition July 7th. However, based on experience from previous years I knew we could become exposed to a serious delay.This possibility made it very risky to make use of complex control algorithms that are difficult to implement and time consuming to tune.I therefore considered it mission-critical to design a work-flow that ensured the presence of a working and bug free system at the start of the vehicle testing period in order not to waste precious time .

3.3 Sensors

Several vehicle states can be measured by available sensors, but the costs of some sensor solutions were beyond Revolve's financial capabilities. The following list shows the control variables that were provided by the sensor packages available to the Revolve 2016 team:

- Longitudinal vehicle velocity V_x,
- Lateral vehicle velocity V_y,
- Vehicle sideslip angle,
- Roll angle and rate,
- Pitch angle and rate,
- Yaw angle and rate,
- Steering wheel angle,
- Throttle position,
- Regenerative brake pedal position,
- Damper displacements.

The following quantities had to be acquired either by estimators or direct calculations:

- Damper rates,
- Steering angle of the steered wheels,
- Tire normal contact force $F_{z,i}$,
- Peak tire-road friction coefficient (μ).

A GPS assisted inertial navigation system (INS), VectorNav VN200, and a non-contact optical speed sensor, Correvit SF-II, ensured that I had accurate yaw rate and longitudinal velocity information.

3.4 Formula Student imposed design rules

The Formula Student rule demanding that a control system can only lower the requested torque from the driver requires that additional logic or constraints must be implemented in the control allocation procedure. The rule saying that any 4WID-EV cannot exceed a certain power limit requires that the control system limits the torque output when one is close to the power limit. And the rule saying that you cannot regenerate energy when the car has a forward velocity of less than $5\,\mathrm{km\,h^{-1}}$ requires that the control system disables regenerative braking below this limit.

4 In silico representation of Gnist

4.1 Formula CarMaker

The vehicle dynamics simulation package Formula CarMaker by IPG Automotive was used to construct an in silico representation of Gnist. In addition to providing a high quality vehicle model, CarMaker provides a virtual driver, virtual tracks and rendering of simulations. The vehicle model that is provided by Formula CarMaker is fully parameterizable, enabling a fairly accurate dynamic representation of Gnist and its operation.

4.2 Adding motors and inverters

I have used a base model from CarMaker that does not contain models for the inverters and the motors. I therefore added simple representations of the motors and the inverters that were used in Gnist, and Figure 7 shows the Simulink implementation of these.

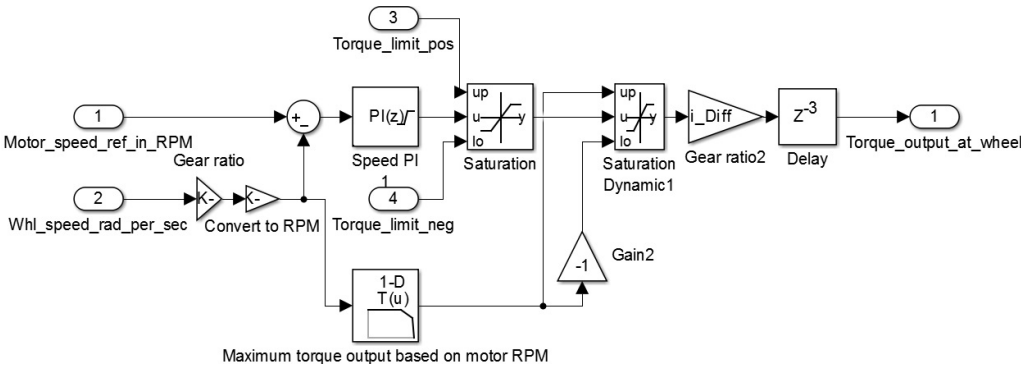


Figure 7: Simulink representation of motor and inverter.

The model assumes perfect tracking of the requested torque magnitude, but with a 3 ms delay from request to torque output from the motor. This delay was chosen based on data from inverter and motor testing in our test bench where I observed that the torque output of the motors tracked the inverter torque setpoints within a 2 to 4 ms delay. The inverters from AMK came with a built-in speed control loop, which is why a PI speed controller is present in the model.

4.3 General vehicle parameterization

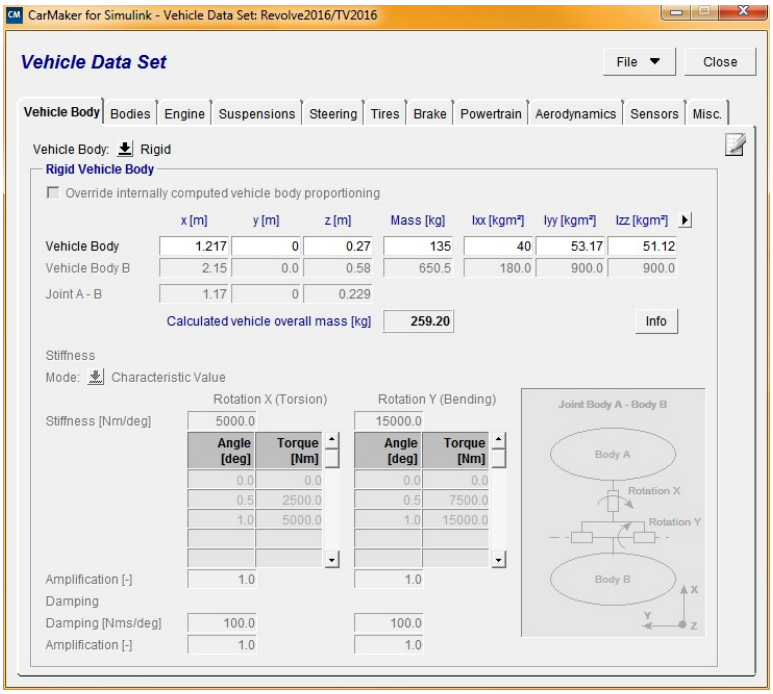


Figure 8: CarMaker GUI for inputting car parameters and models.

Figure 8 shows the GUI for adjusting parameters and choosing different models for the vehicle. Most of the general parameters were found by using the detailed CAD model of Gnist.

4.4 Suspension kinematics

The Revolve 2016 team member Kurt Erik Nesje was responsible for designing the suspension of Gnist. He supplied me with parameters that describe the complete geometry of the suspension, which were used in CarMaker.

4.5 Aerodynamic parameterization

The aerodynamic down force and drag were modeled with two static coefficients, found through simulations by use of the computational fluid dynamics software STAR CCM. This method is accurate to within 10% of the complex full car aerodynamic simulations, which I considered to be good enough for the purposes of this thesis.

4.6 Tire representation

As tires are the main source of the forces acting on the vehicle it is important that the model representing the tires was as accurate as possible. As Revolve got access to test data for the tires that were used on Gnist, the Revolve team member Kurt Erik Nesje therefore generated a combined longitudinal and lateral magic formula version 5.2 model using the software Optimum Tire. The tire model was then imported into CarMaker. Unfortunately, the tire forces generated with this model were incredibly unstable in CarMaker. The reference manual for CarMaker mentions that they have trouble with the stability of magic formula type models at slow speeds. Lack of documentation regarding this issue and time pressure forced me to abandon the magic formula tire model, and instead use an existing tire model in CarMaker that was tuned to match the force generation of the tires that were used on Gnist.

4.7 Power usage

Total power usage of the motors must be available in the simulations to test the power limiting controller, and to determine the loads the battery cells will be subjected to. An estimate of the power use can be calculated from the mechanical power output of all the motors and the motor efficiency. The mechanical power output (P_{em}) from a motor is given by:

$$P_{em} = \omega_m T_m, \tag{1}$$

where ω_m is the angular speed in $\mathrm{rad\,s^{-1}}$ and T_m is the torque output. The input power to the motor (P_{el}) can be expressed by:

$$P_{el} = \frac{P_{em}}{\eta}, \tag{2}$$

where η is the efficiency rating of the motor. A table of motor efficiency values was supplied by AMK:

		speed [rpm]									
Current [Arms]	Torque [Nm]	500	1000	2000	3000	4000	6000	10000	12000	15000	19000
5	1,3	64,37	71,33	73,64	74,70	75,43	76,57	77,00	77,08	77,56	78,14
10	2,7	58,42	70,48	77,57	80,40	82,01	83,92	85,16	85,44	85,97	86,50
20	5,4	44,94	60,81	73,35	78,82	81,94	85,43	88,20	88,88	89,71	90,44
30	7,9	35,59	51,90	67,02	74,26	78,54	83,42	87,58	88,65	89,84	90,86
40	10,4	29,14	44,78	61,01	69,41	74,57	80,62	85,93	87,34	88,86	90,16
50	12,5	24,17	38,71	55,22	64,39	70,24	77,30	83,73	85,48	87,37	88,98
60	14,4	20,41	33,76	50,04	59,65	65,99	73,88	81,33	83,42	85,66	87,59
70	16,0	17,31	29,40	45,10	54,87	61,55	70,10	78,56	80,97	83,56	85,81
80	17,4	14,82	25,75	40,67	50,41	57,28	66,34	75,70	78,40	81,34	83,91
90	18,5	12,81	22,67	36,72	46,30	53,25	62,67	72,77	75,75	79,02	81,91
100	19,6	11,17	20,05	33,21	42,51	49,44	59,09	69,82	73,06	76,63	79,83

Table 1: Efficiency of the motor at different RPMs and current. Taken from the motor datasheet.

By creating a lookup table of Table 1 in Simulink, the combined electrical power usage was found.

5 Estimated control variables

A few critical control system variables had to be estimated. These were normal tire forces, longitudinal tire forces, and peak tire-road friction coefficient. I therefore developed methods to estimate these variables.

5.1 Normal tire force

The normal tire force on each wheel was used to find the tire adhesion limits and to distribute power between the front and rear motors. Two approaches were assessed in silico. One was based on combining aerodynamic simulation data with load transfer equations, while the other was based on information obtained from damper displacement sensors.

5.1.1 Use of load transfer and aerodynamic lookup tables

If one ignores suspension dynamics and assumes pitch and roll dynamics coupling, a nonlinear formulation of the normal force for each wheel due to load transfer is given by the following equations [24]:

$$
\begin{aligned}
F_{z,FL} &= \frac{1}{2} m_v \left(\frac{l_r}{L} g - \frac{h_c}{L} a_x\right) - m_v \left(\frac{l_r}{L} g - \frac{h_c}{L} a_x\right) \frac{h_c}{t_f g} a_y, \\
F_{z,FR} &= \frac{1}{2} m_v \left(\frac{l_r}{L} g - \frac{h_c}{L} a_x\right) + m_v \left(\frac{l_r}{L} g - \frac{h_c}{L} a_x\right) \frac{h_c}{t_f g} a_y, \\
F_{z,RL} &= \frac{1}{2} m_v \left(\frac{l_f}{L} g + \frac{h_c}{L} a_x\right) - m_v \left(\frac{l_f}{L} g + \frac{h_c}{L} a_x\right) \frac{h_c}{t_r g} a_y, \\
F_{z,RR} &= \frac{1}{2} m_v \left(\frac{l_f}{L} g + \frac{h_c}{L} a_x\right) + m_v \left(\frac{l_f}{L} g + \frac{h_c}{L} a_x\right) \frac{h_c}{t_r g} a_y.
\end{aligned}
\tag{3}
$$

These equations calculate the vertical forces statically, and do not include the additional force from aerodynamic contributions. They also assume a constant value for the vehicle's center of gravity (CoG) height, which for a normal vehicle can change considerably. For the car in question the COG height is not an issue. Not including suspension dynamics was justified by the fact that the vehicle in question would be driving on relatively flat tracks with no considerable bumps.

The Revolve 2016 car was also equipped with anti-roll bars. The stiffness will however be quite even at the front and rear, which means that the load distribution will not be affected significantly. As this method only considers loads due to dynamic and static load transfers, aerodynamic forces were included to get an accurate representation of the normal forces.

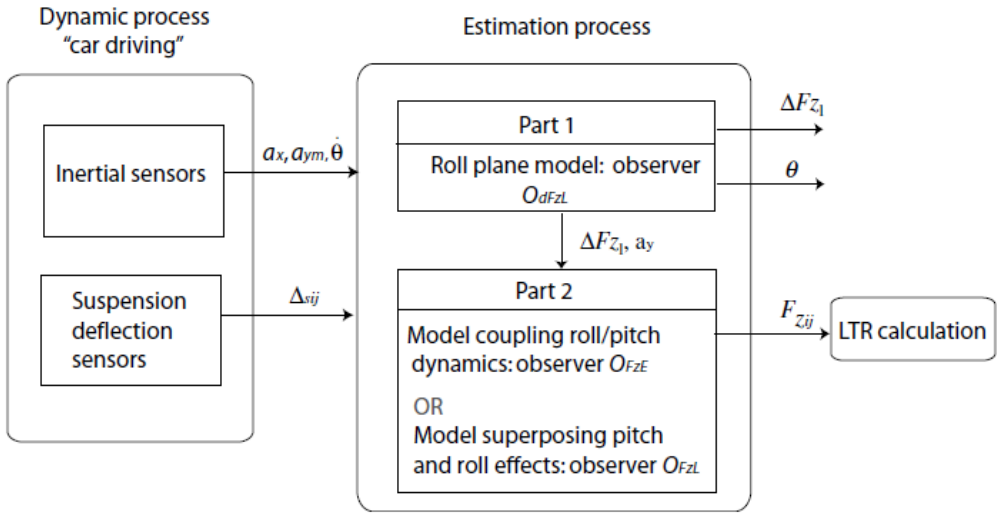


Figure 9: Overview of observer structure [24].

Figure 9 shows the overall observer structure of a popular roll angle, lateral load transfer and normal force estimator where a quasi-flat road without irregularities is assumed. Part 1 estimates the lateral load transfer and the roll angle, while Part 2 estimates the vertical force at each wheel. Part 1 is normally implemented using a linear Kalman filter, while Part 2 can be implemented with either a linear or extended Kalman filter, depending on whether a linear or nonlinear load transfer model is used. This estimator was implemented using linear Kalman filters only, and was tested thoroughly in the full car simulations. However, it was not used in the vehicle as the method outlined in the next section worked better.

5.1.2 Use of damper position and rate

Damper displacement sensors on the vehicle can be used to find the normal force on each wheel. I chose to use a method outlined in reference [23] that uses the damper displacements, damper rate and the vertical acceleration of the wheel carrier to calculate the normal force. Figure 10 shows how the sprung mass of the vehicle interacts with the unsprung mass through a spring constant k and a damping constant b, and where the unsprung mass has damping and spring constants given by the tire characteristics.

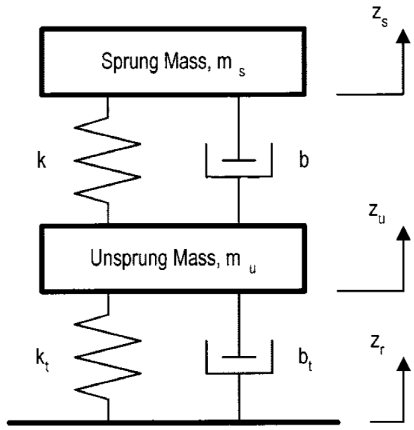


Figure 10: Schematic of simplified damping system on a car [23].

From reference [23] the following formula is used to find the normal force:

$$F_z = b(\dot{z}_s - \dot{z}_u) + k(z_s + z_u) - m_u\ddot{z}_u \tag{4}$$

In the Revolve 2016 vehicle the suspension made use of the pushrod principle shown in Figure 11.

Figure 11: Pushrod suspension on a Lamborghini.

Vertical wheel displacement causes a damper displacement that depends on the geometry of the push-rod and the bell crank. This relation is called the motion ratio (MR). I excluded the contribution due to vertical wheel carrier acceleration as the vehicle only drives on flat and even tracks where bumps that cause significant acceleration of the wheel carrier are very rare. The formula to find the normal tire force (F_z) for our suspension design may then be expressed as:

$$F_z = K_s \triangle z MR + C_s(\dot{z}) \triangle \dot{z} MR, \tag{5}$$

where K_s is the spring coefficient and $C_s(\dot{z})$ is the compression/rebound coefficient of the damper. The damper based estimation method proved to be accurate and robust, and ended up as my choice of method for acquiring normal forces on each wheel.

However, some changes to eqn. (5) had to be made due to some issues that were discovered during vehicle testing. I first attempted to calculate the normal forces without considering rebound and compression forces. However, this caused excessive wheel spin when wheel lifts occurred. There is a delay when extending the suspension which is caused by rebound damping. If rebound is not considered the calculated normal force will be higher than what the wheel is actually experiencing. After including rebound in the calculation, wheel spin during wheel lifts were greatly reduced. I also had to adjust the equation to accommodate for preload settings. A

preloaded damper must be exposed to a force beyond the preload threshold before it starts compressing. The normal force for a wheel not in contact with the ground will therefore be equal to the preload force. A wheel not in contact with the ground should not be allocated any torque. I therefore added a scaling effect so that the final normal force equations became:

$$\begin{aligned}
F_{z,FL} &= K_{s,front}\triangle z_{FL}MR - RB_{FL} + preload_{front}K_{s,front}min(\triangle z_{FL}, 1),\\
F_{z,FR} &= K_{s,front}\triangle z_{FR}MR - RB_{FR} + preload_{front}K_{s,front}min(\triangle z_{FR}, 1),\\
F_{z,RL} &= K_{s,rear}\triangle z_{RL}MR - RB_{RL} + preload_{rear}K_{s,rear}min(\triangle z_{RL}, 1),\\
F_{z,RR} &= K_{s,rear}\triangle z_{RR}MR - RB_{RR} + preload_{rear}K_{s,rear}min(\triangle z_{RR}, 1),
\end{aligned} \tag{6}$$

where RB is the rebound effect. Figure 12 shows the calculated normal forces on each wheel for a skidpad run where the driver is constantly pushing the car to the traction limit.

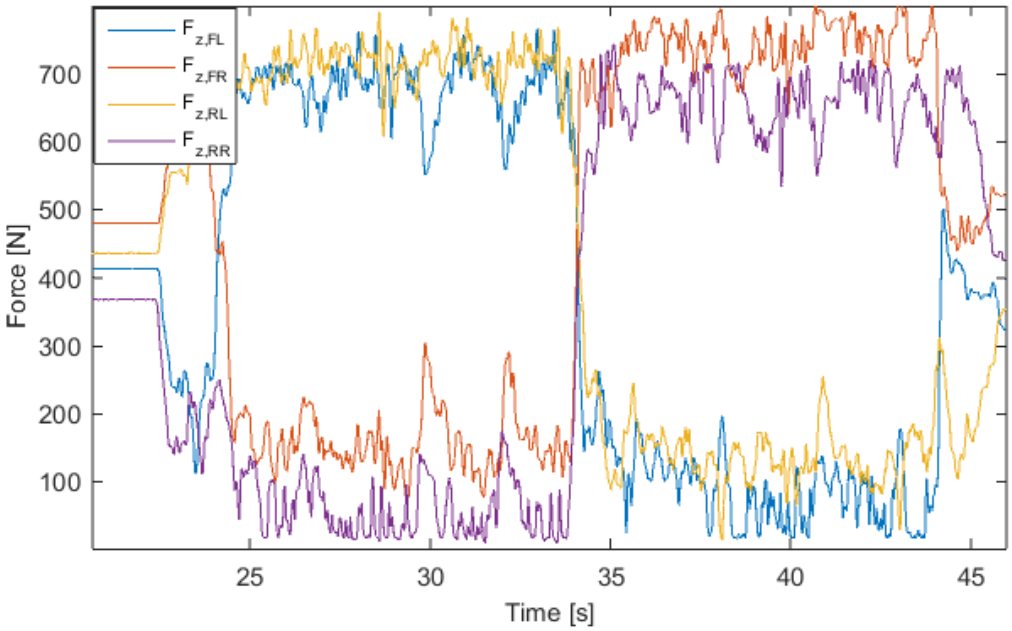


Figure 12: Normal forces on each wheel during a skidpad run.

5.2 Longitudinal tire force estimation

To find the longitudinal tire force I made use of the following equation for each wheel:

$$J\dot{\omega} = T - RF_x, \tag{7}$$

where J is the total wheel inertia, $\dot{\omega}$ is the angular acceleration of the wheel, T is the applied torque, R is radius of the wheel and F_x is longitudinal force. The applied motor torque was found using the following equation:

$$T = I_q K_t, \tag{8}$$

where I_q is the torque current, and K_t is the torque constant of the motor. The angular acceleration of the motor can be found by taking the derivative of the RPM values, which the inverters calculate from high resolution encoder measurements. The longitudinal tire force F_x can therefore be calculated directly from eqn. 7.

5.3 Peak tire-road friction coefficient

The peak tire-road friction coefficient was used by both the traction and the yaw rate controller. The traction control used the peak tire-road friction coefficient to limit the allowable torque, which makes the PI traction controllers more effective, while the yaw rate controller could use it to calculate realizable yaw rate reference set points.

The normalized traction force (τ) for only longitudinal maneuvers is given by:

$$\tau = \frac{F_x}{F_z}. \tag{9}$$

The maximum value of this number is the peak tire-road friction coefficient. The slip ratio at this point will be the ideal slip ratio one tries to maintain with traction control. The allowed maximum torque for the wheel can be found from:

$$T_{max} = \frac{F_z \cdot \mu_{peak}}{r_{eff}}, \tag{10}$$

where μ_{peak} is the peak tire-road friction coefficient and r_{eff} is the effective wheel radius. More torque than T_{max} will not cause any further performance increase. Reducing the available torque domain also improves the PI traction controllers as there is a smaller area to regulate on.

The peak tire-road friction coefficient can then be found by plotting the normalized traction force against the slip ratios. Using this method together with data acquired from the test track under dry conditions suggests that the ideal slip ratio lies somewhere between 0.12 and 0.17, and the peak tire-road friction coefficient somewhere between 2.2 and 2.8. The peak tire-road friction coefficient is so high because the calculated normal forces do not consider unsprung mass. I did not prioritize estimat-

ing the peak tire-road friction coefficient and the ideal slip ratio more accurately, as it turned out to be sufficient to make educated guesses and determine values based on vehicle performance.

6 Description of the embedded system implementation

The vehicle control unit (VCU) running the control systems consisted of a printed circuit board (PCB), embedded real-time operating system, drivers, generated control system code, and all the self-written code that handled safety and other functions.

6.1 VCU hardware

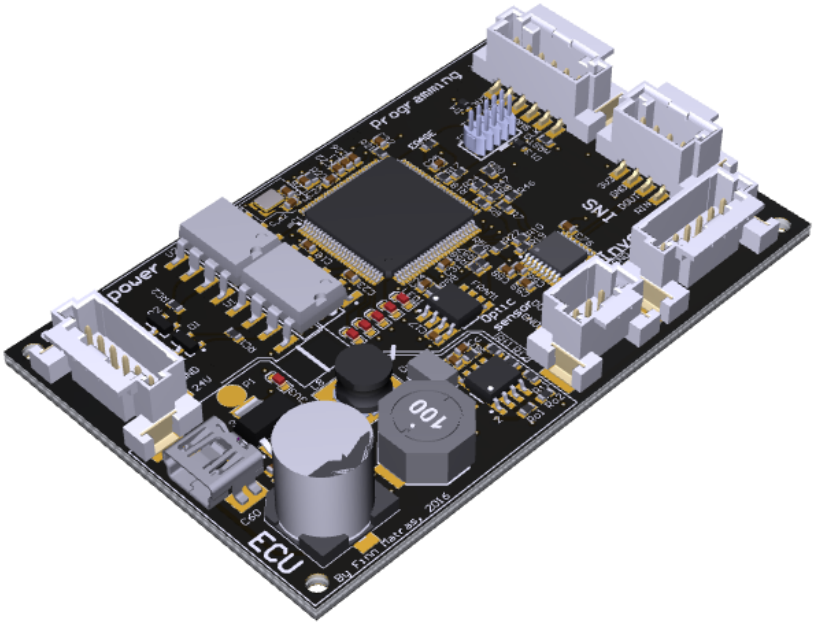


Figure 13: The top layer of the in-house developed vehicle control unit PCB. In Revolve the VCU is called engine control unit (ECU), which is why the PCB is marked ECU.

The VCU hardware was a printed circuit board (PCB) designed by Finn Matras from the Revolve 2016 team. The top layer of the PCB is shown in Figure 13. The key hardware components of the VCU PCB are an Atmel SAME70 microcontroller, two CAN 2.0 transceivers, two UART transceivers and one ferroelectric-RAM (FRAM) IC. The Atmel SAME70 microcontroller is built around an ARM Cortex M7 microprocessor. This microprocessor can run at up to 300 MHz, and it has a floating point unit (FPU)

supporting both single and double precision floating point numbers. Hardware floating point support is incredibly useful as one can perform all calculations with floating point numbers, instead of performing fixed-point arithmetic to optimize execution efficiency.

The two Controller Area Network (CAN) transceivers were needed to connect the microcontroller to the two CAN bus networks that were used to pass information between the systems located throughout Gnist. One UART bus was used for communication between the GPS-aided inertial navigation system (Vectornav VN200 Rugged) and the microcontroller, while the other one was used for communication between the non-contact optical speed sensor (Correvit SFII) and the microcontroller. Non-volatile memory was needed to store important information such as tuning parameters. This was handled with non-volatile flash memory on the SAME70 microcontroller, but a FRAM IC was included on the PCB for redundancy in case the flash memory on the SAME70 became corrupted.

6.2 FreeRTOS - embedded real-time operating system

FreeRTOS is a light weight embedded real-time operating system made specifically for microcontrollers. FreeRTOS provides methods for tasks (threads), mutexes, semaphores, queues and software timers. This functionality is extremely helpful when writing code for relatively complex systems on microcontrollers. Excellent documentation of FreeRTOS' functionality makes it easy to learn the operating system quickly.

Many FreeRTOS ports for specific microcontrollers, including SAME70, can be downloaded from the FreeRTOS webpage. Setting up FreeRTOS for the SAME70 microcontroller was therefore unexpectedly painless.

6.3 Microcontroller drivers

Microcontroller drivers are needed for utilizing microcontroller functionality such as CAN, SPI and UART communication. Drivers tie hardware and software together by providing methods for specific actions that abstract away low level actions, such as register handling. The Atmel Software Framework (ASF) is an attempt by Atmel to provide ready-to-go drivers, which developers can use to reduce development time. However, the ASF package is quite messy as it supports a multitude of different microcontroller models in one package. Normally one implements methods to utilize only a small subset of the complete functionality within one hardware module. ASF drivers on the other hand, offers methods to utilize all of the hardware functional-

ity in each module. This versatility makes the ASF drivers somewhat bloated and confusing. As a driver package should ideally be as clear and specific as possible to facilitate correct use, ASF is in my opinion therefore only a good starting point for driver development if one desires clean and specific interfaces. I therefore invested a substantial amount of time to adapt ASF drivers for the SAME70 microcontroller, which were tailored for both myself and other users in Revolve. The hardware module drivers I did the major development on, were; CAN, UART, SPI, and the power management controller (PMC).

6.4 Simulink Embedded Coder

The Embedded Coder package for Simulink played an integral part in the embedded implementation by translating the Simulink implementation of the control system to C code. It is important to design the system in Simulink with code generation in mind, by using buses and appropriate data types for variables and parameters in the system. Buses allow you to pack multiple signals into one object. The Embedded Coder converts Simulink buses to C structs, which are very handy for modularizing system information. The default data type for all signals in Matlab is a double, a 64 bit floating point number. This accuracy was not needed in my case, and all floating point variables and parameters were therefore represented as singles, which are 32 bit floating point numbers. The FPU has a higher throughput for single precision calculations than double precision, which is also a factor to consider. I also set up the Embedded Coder to utilize the ARM math libraries, which are optimized for the microcontroller hardware. This means that, for example, a call to sin(x) in Simulink will be translated to armsin32(x) in C.

6.5 Microcontroller software architecture

The software development for the VCU was done by myself and Ørjan Gjengedal from the Revolve 2016 team. An overview of the system is shown in Figure 14.

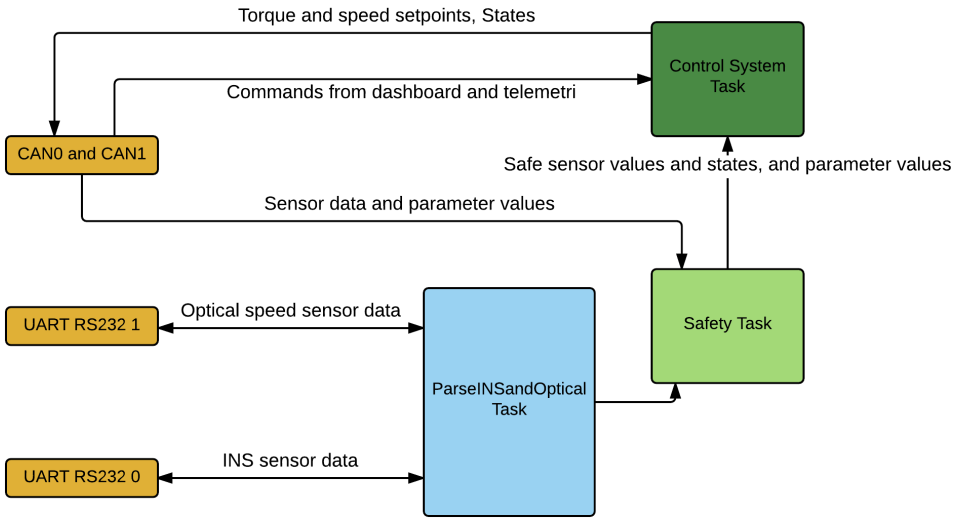


Figure 14: High level overview of the VCU software and communication.

Ørjan was responsible for the safety task, but I also contributed significantly to this task. The safety task fetched sensor data from the communication buses, validated sensor data, computed the derivative of a few sensor signals, filtered some sensor signals, and passed validated sensor data to the control system task. In the case that a sensor signal dropped out or was outside its value bounds, the safety task notified the control system task.

The control system task was mainly responsible for executing the control system at a fixed frequency, passing the calculated setpoints to the inverters via the CAN buses, and controlling the state of the inverters. Code to start and stop the vehicle, activate and deactivate launch control, ensure redundancy in case of sensor or system failures, was also added to this task.

The INS parsing task accessed the UART bus which was connected to the INS to read out data. The INS sent data over UART as ascii characters. The INS was programmed to output blocks of information at a specified frequency. Sensor values such as velocities, yaw rate and accelerations were sent at 150 Hz, while sensor values such as GPS latitude, longitude, altitude coordinates, and GPS status were sent at 10 Hz. The fast sensor data message was 56 bytes, while the slow sensor data message was 36 bytes. Making the processor responsible for handling every byte received over UART is a suboptimal strategy, as the microcontroller had a hardware unit called direct memory access controller. This controller was programmed to read

out bytes from the UART unit, without any processor load, and placed them in the microcontroller memory. Once the direct memory access controller had read out a complete message, the INS parsing task read and copied the memory block into C structures which were made available to the safety task for further processing.

6.6 Control systems tuning interface

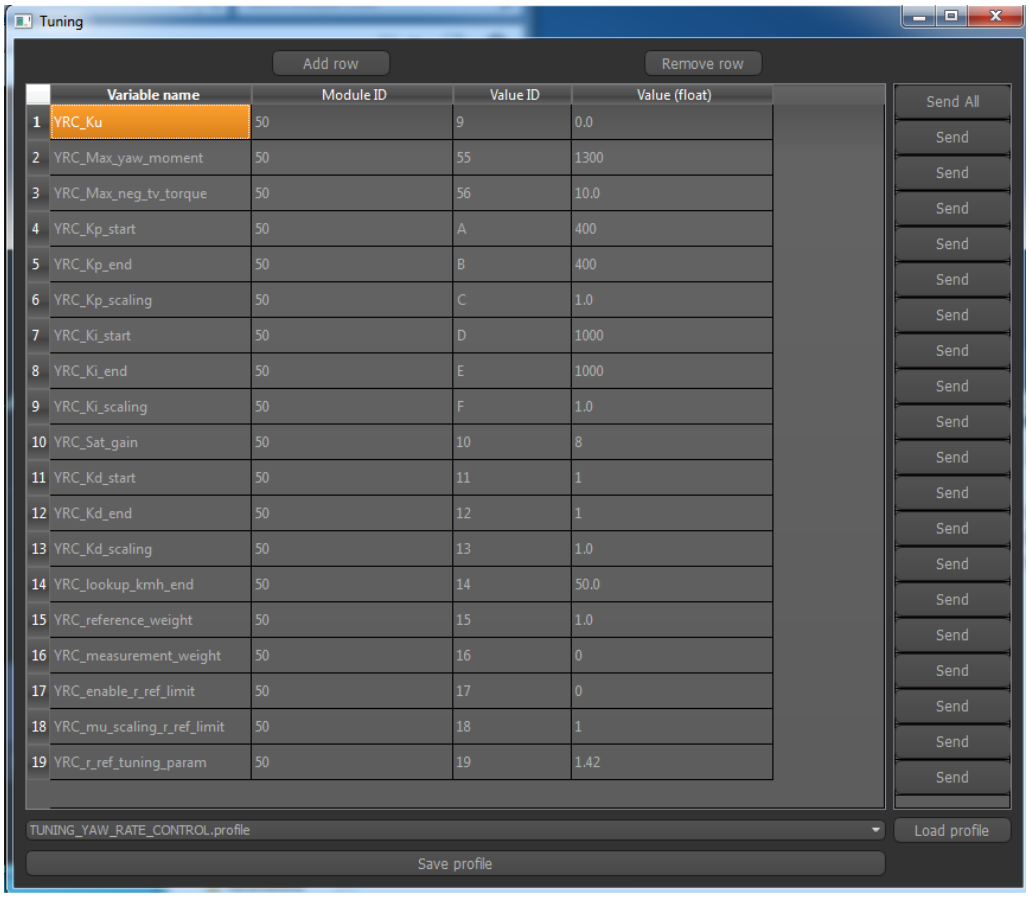


Figure 15: Overview of the tuning interface.

A wireless interface was used to send control system and vehicle parameters from a laptop to the vehicle control unit. The interface consisted of Revolve Analyze, a multi-purpose in-house developed software package, a telemetry base station, and a telemetry car module which forwarded data on the two CAN buses in Gnist. I programmed support to accept parameters via this telemetry link on the VCU. Checksum verification was implemented to ensure that the data was valid. The rationale

for doing this was that being able to quickly change parameters when testing is critical for saving time. The alternative is to reprogram the microcontroller every time a parameter must be changed, which is extremely time consuming. Figure 15 shows the tuning interface in Revolve Analyze.

6.7 Tracealyzer - code execution visualization

When programming on an embedded platform it is often difficult and cumbersome to gain insight into how the code is being executed. Percepio is the company behind the software *Tracealyzer*, which aims to provide the tools necessary to visualize information about code execution on embedded platforms. *FreeRTOS+Trace* is a version of *Tracealyzer* specifically made for microcontrollers running FreeRTOS.

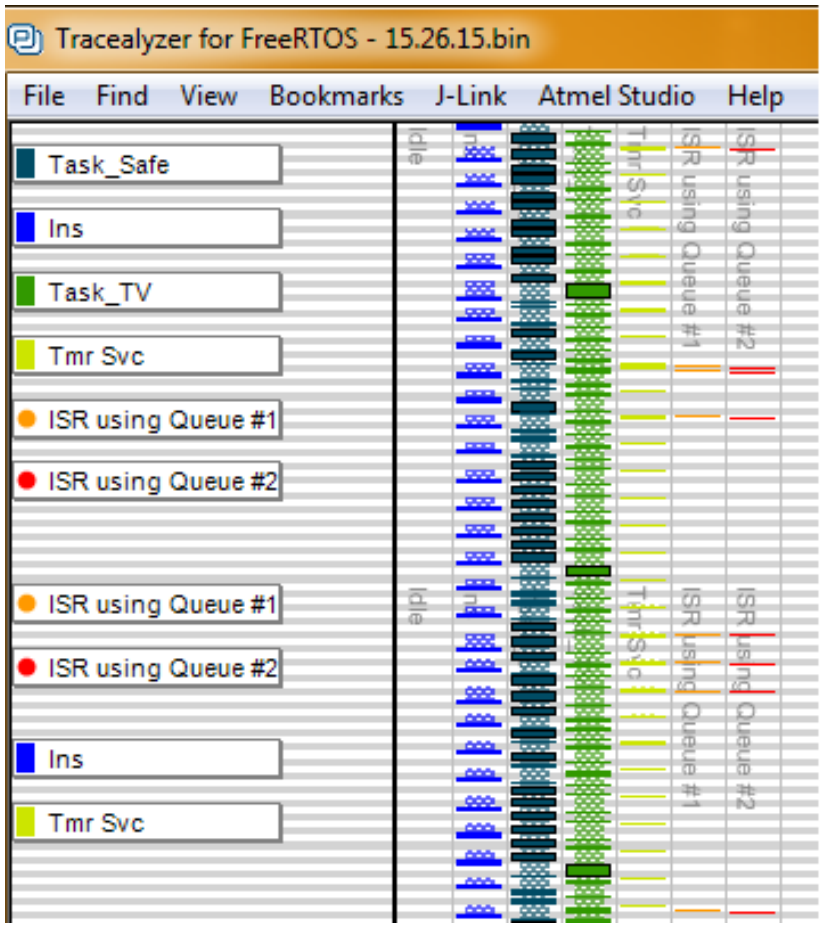


Figure 16: Tracing three tasks on the microcontroller.

Figure 16 shows a screenshot of the main overview screen of the FreeRTOS+Trace software. The size of the solid colored blocks represent the execution time of the code, while the perforated blocks represent the amount of time the task is in ready mode before it actually gets executed. In the depicted case all the tasks had equal priority, and the scheduling worked on a first in first out principle.

The basic workflow for using *FreeRTOS+Trace* is as follows. Add a C code library for your specific microcontroller and FreeRTOS version into your code project. Call two functions from this library in your code. Start a debug session using a debugger which supports RAM dumping. While the program is running in debug mode, events which describe the execution of the individual parts of your code are written to RAM. To see the events that have been stored, you break the debugging and dump the information stored in RAM to *FreeRTOS+Trace* via the debug probe. Now the software can display information such CPU load, task execution time and much more from the data stored in the ram dump.

I used *FreeRTOS+Trace* to gain insight into task execution frequencies, execution durations, queue accesses, semaphore unlocking and locking, and mutex unlocking and locking. An analysis of these elements are vital to ensure the correctness, stability, and efficiency of the VCU software. By using *FreeRTOS+Trace* in addition to manual debugging, instead of only using manual time debugging and CAN debug messages, I saved a lot of time.

Results from tracing showed that it took maximum 400 µs to run both the estimation part and the control system part one time. As the execution frequency of the control system was 100 Hz, this was short enough. The tracing also showed that the tasks, mutexes and queues were working as intended.

6.8 Microcontroller-in-the-loop

The embedded implementation of the control systems had to be verified before they could be used in the vehicle. I chose to do this by simulating the vehicle and generating sensor values in real-time in Simulink on a desktop PC, which were sent to the microcontroller running the control system. The embedded version of the control system on the microcontroller then calculated the torque setpoints to the inverters, which were sent back to Simulink and used in the vehicle simulation. A CAN bus was used to communicate between the microcontroller and Simulink and a CAN to USB adapter was used to make the desktop PC able to receive the CAN data. Simulink requires an addon package called Vehicle Network Toolbox to be able to fetch CAN

data received over USB. The Simulink Desktop Real-Time addon was used to simulate the vehicle model and generate sensor values in real-time. The problem with this approach was that neither Windows 7 nor USB are made for real-time applications. To address this, Simulink I gave real-time priority in the Windows task manager. This provided Simulink more run time and thus opportunities to fetch CAN messages from the USB stack. With this setup, Simulink running in normal mode was able to access around 1000 CAN messages per second. As long as the amount of data and the cycle time of the data were below this limit, the timing errors could be kept quite small. Readers experienced with Simulink might wonder why Simulink external real-time mode is not used. This is because CarMaker for Simulink has implemented their vehicle model as S-functions, and I was unable to generate an executable using the target language compiler files that were available to me.

My approach did, however, seem to work reasonably well. To compare the performance of the Simulink version of the control system and the embedded version, two identical simulations were performed. One was done solely in Simulink, while the other used the method detailed above. Figure 17 shows the result of a maneuver consisting of accelerating, turning left and then back to neutral. This confirmed that the control system worked almost identically on the microcontroller as in Simulink.

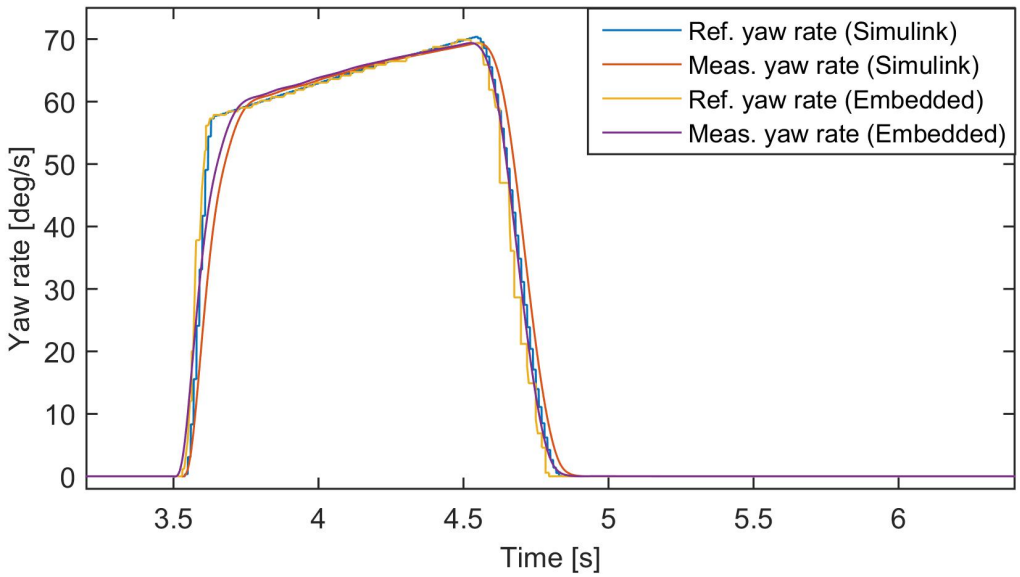


Figure 17: Comparing two simulations, where one is performed solely in Simulink, while the other one runs the control system on the microcontroller and the vehicle simulation in Simulink.

7 Systems overview

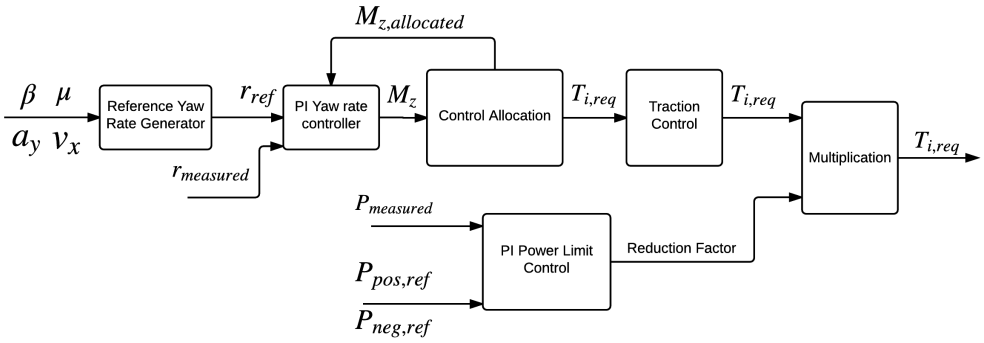


Figure 18: Systems overview.

Figure 18 shows an overview of the control systems that were designed and implemented. The reference yaw generator calculates a desired yaw rate using steering wheel angle input (β), tire-road friction coefficient (μ), lateral acceleration (a_y), and longitudinal velocity (v_x). The yaw rate controller outputs the yaw moment necessary for the vehicle to track the desired yaw rate (r_{ref}). $M_{z,allocated}$ is the actual yaw moment that was possible to allocate within the physical limitations of the system, and is used as an anti-wind up measure in the yaw rate controller. The control allocation converts the desired yaw moment into torque requests for each motor. These torque requests are modified by the traction control block if the wheels exceed a certain slip ratio. By reducing the torque requests equally if a power limit is exceeded, the power limit controller ensures that one does not exceed the power limit defined by the competition rules.

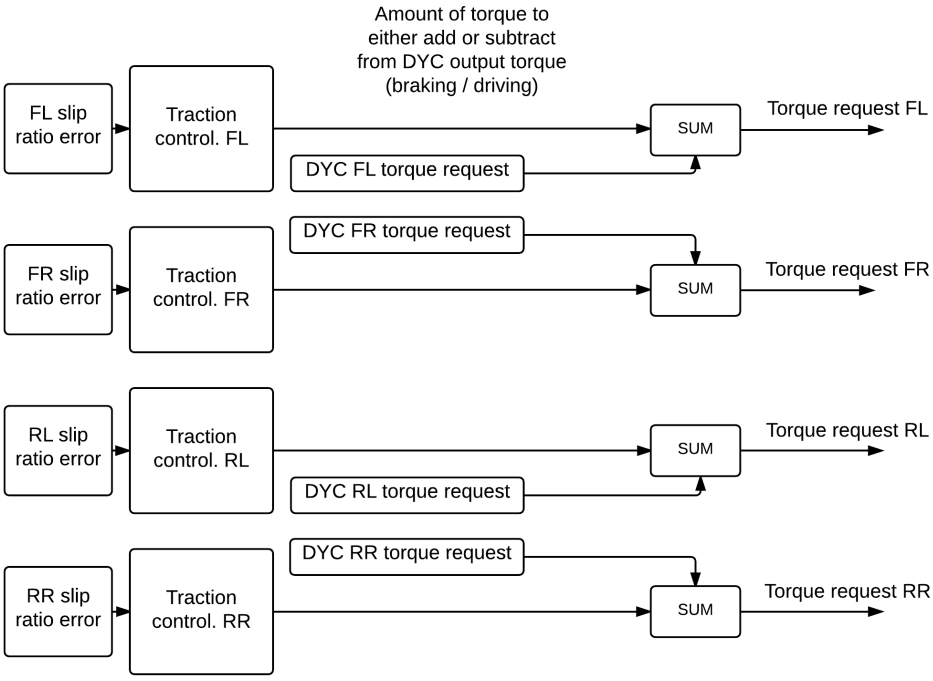


Figure 19: Traction control system overview.

An overview of the traction control (TC) system is shown in Figure 19. Torque requests from the DYC system are passed into the TC system, and modified if necessary. Because of the uniform friction conditions of the tracks Gnist was supposed to drive on I decided that there was no need to merge traction controller outputs. This also reduced the complexity of the system.

8 Direct Yaw Moment control - design concept options

A DYC system for 4WID-EVs consists of three main subsystems; reference value generation, high level controller and control allocation. The actual methods used in these subsystems vary substantially in the literature. The DYC literature is also very much focused on commercial vehicles that are different from the Revolve vehicle, which impacts several design aspects. The most important differences are summarized in Table 2.

Table 2: Table of relevant differences between commercial vehicle applications and Gnist.

Type	Commercial vehicle	Gnist
Customization	The system cannot be customized by the owner.	Full customization of the system is possible for Revolve.
Performance	The performance of the vehicle is not to come at the expense of safety.	Maximum performance is required, at the expense of safety features.
Driver Skills	No assumptions can be made regarding the skill of the driver.	The drivers will be trained specifically to master the the vehicle as a part of the control system.
Parameter Variations	Large variations in load due to passengers and baggage.	Almost no variations in load. Variations can be manually adjusted for in the control system.
Environmental Variations	The vehicle must be able to tackle; bumpy roads, gravel, ice and more.	Gnist is driven on flat and even race tracks. Friction variations will occur due to water and asphalt variations.

Being aware of these differences made it possible to choose, adapt and design methods that were streamlined to optimize the performance of the race car.

8.1 Reference variables

A yaw rate reference and feedback makes it possible to track a desired steering characteristic, which is given by the method used to generate the reference yaw rate. Yaw rate control can also be sufficient in keeping the vehicle within is stability limits, which is nicely summarized in reference [8]: "In general, the value of the sideslip angle can be kept within the stability limits of the vehicle through a yaw rate controller if the friction coefficient at the tire–road contact is accurately estimated and a correct reference yaw rate is generated. Under these conditions, yaw rate feedback control is sufficient to ensure safe handling and driving behavior." However, it cannot be assumed that these conditions are met at all times, which makes it sensible to also introduce a sideslip angle reference if vehicle stability is paramount. I would argue that ensuring stability is not the most important factor in a race car. Lap times will be slower if a control system cuts motor power and slows the car in a situation where the driver would otherwise be able to keep a higher forward speed. Due to this I chose to only use a yaw rate reference, and therefore only a yaw rate controller, in this system.

8.2 Yaw rate controller

The yaw rate controller can generate the corrective yaw moment either by feedback control only, or by a combination of feedback and feedforward control. The general requirements to be fulfilled by such a controller are:

- Predictable controller behavior,
- Easy to tune,
- Acceptable performance compared to alternatives,
- Low computational complexity.

8.2.1 Feedback control

The main goal of the feedback controller is to produce a corrective yaw moment that should make the vehicle track the desired yaw rate. For a commercial vehicle the total mass and distribution of mass and the tire-road friction coefficient can vary drastically. Control methods that are robust to system uncertainties and external disturbances are therefore very popular in DYC systems. One method to generate the corrective yaw moment is either to make use of a first order sliding mode controller [28] or a second order sliding mode controller [18]. However, sliding mode controllers are quite complex and therefore conflict with the requirement that the chosen methods should be intuitive and easy to tune. Moreover, our drivers are fairly similar, being around 170-180 cm and 70-75 kg, and no significant extra load will be removed or added to the vehicle. The Revolve car is thus not subject to large uncertainties in total mass and distribution of mass. The tire-road friction coefficient will vary though, but it is quite uniform as we are driving on race tracks, and we can also tune the chosen controller specifically for certain events and operating conditions. Reference [8] compared a PID controller with feedforward against two different sliding mode control formulations and concluded: "The PID-based controllers achieve very good vehicle performance in steady-state and transient conditions, whereas the controllers based on the sliding-mode approach demonstrate a high level of robustness against variations in the vehicle parameters".

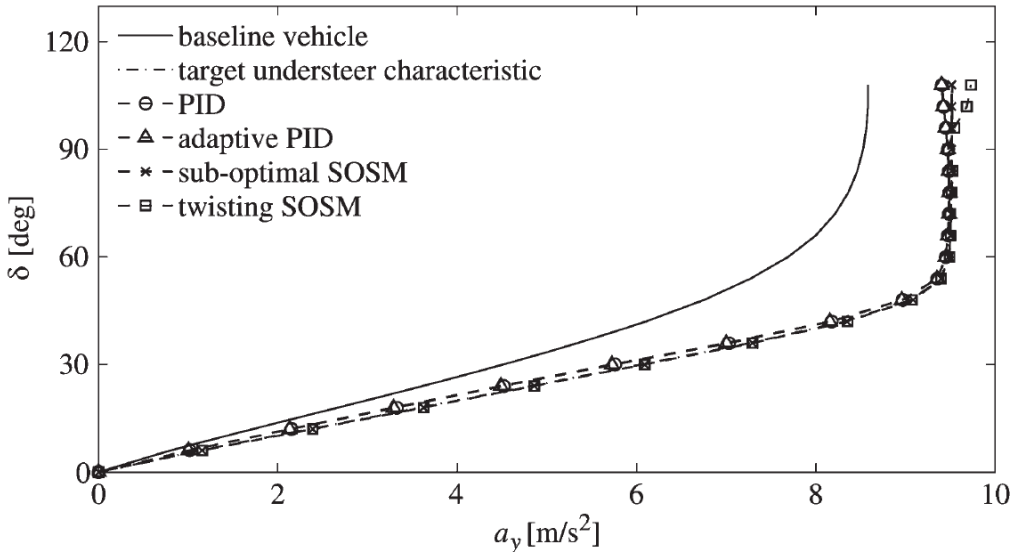


Figure 20: Understeering characteristic evaluated at V = 90 km h^{-1} for (solid line) the baseline vehicle and for the controlled vehicle equipped with different yaw rate controllers in high-friction conditions [8]. β is the steering wheel angle.

Figure 20 (taken from reference [8]) shows that the much simpler PID controller is able to achieve almost the same performance as the second order sliding mode controller. Based on the information above and the nature of the controller requirements, I found it reasonable to focus on PID-based controllers only. These exhibit predictable behaviour, are easy to tune and have a low computational complexity. Whether or not a derivative term can be used successfully for this application depends on sensor data quality and cannot be decided before the vehicle is driving. I therefore chose to focus on a PID-based approach including the possibility for introducing a derivative term that could be switched on or off, and the possibility for using only the reference or the measurement, or a weighted combination of both.

8.2.2 Feedforward control or not?

A fundamental premise for making use of feedforward terms for DYC control is that the baseline yaw dynamics of the vehicle in question is known. This enables the design of algorithms than can calculate the necessary added yaw moment to realize the desired understeer characteristic. Feedforward is included in addition to feedback controllers in many published DYC solutions ([14], [8], [15]). The main argument for feedforward contributions is that they can act faster than feedback systems, and therefore better utilize the responsiveness of the electric motors.

The feedforward control system designed in reference [15] is based on two independent terms, one static and one dynamic. The static feedforward term is designed to achieve the reference understeer characteristic under steady-state conditions, while the dynamic feedforward term is designed to reduce the yaw rate and sideslip oscillations induced by fast steering inputs. The static feedforward term is found through an offline optimization procedure based on a quasi-static vehicle model [16]. To find the necessary added yaw moment to realize the desired understeer characteristic, this procedure relies on knowing what yaw rate the baseline vehicle with no control is able to achieve under a large set of conditions. Based on this information and the vehicle model, the added yaw moment needed for shifting the baseline yaw rate to the desired yaw rate can be calculated. The results are put into a look-up table which can be used in the real vehicle. The added yaw moment term relies on an experimentally validated vehicle model that represents the lateral dynamics of the actual vehicle. This model is not trivial to construct due to nonlinearities in the tires at lateral accelerations over $5\,\mathrm{m\,s^{-2}}$. To realize this feedforward design, considerable effort must therefore be invested to construct an experimentally validated vehicle model.

Another proposed feedforward scheme [14] is based on a linear single-track vehicle model, which the authors assumed could accurately describe the lateral dynamics up to a lateral acceleration of $4\,\mathrm{m\,s^{-2}}$. In this case a transfer function has to be constructed that maps the front steering wheel angle to the necessary yaw moment needed to achieve a desired understeer characteristic.

In theory, feedforward control with feedback to correct for inaccuracies in the feedforward models would make the system faster than using pure feedback. I therefore seriously considered to implement this control regime. However, in practice, the design, implementation, and tuning of such a system is substantially more complex than a PI/PID controller. And both the above feedforward schemes rely on experimentally validated models, which were only possible for the Revolve team to acquire after the car was supposed to be running in May/June. This experimental validation was anticipated to be so time consuming that is probably could not be properly pursued in the already hectic and short test period. In order not to risk project failure by attempting to implement an overambitious control regime, I therefore decided that the inclusion of feedforward control had to be postponed till later.

8.3 Control allocation

The control allocation scheme is responsible for determining the correct motor torques to achieve the reference yaw moment it receives from the yaw rate controller, while satisfying tire adhesion and motor saturation limits as well as the driver torque request. In reference [11] the authors express the control allocation as a quadratic programming (QP) problem where the objective is to find an optimum combination of motor torques that simultaneously satisfies the yaw moment demand, minimizes the control efforts and ensures that each tire is within its capacity limit. This QP problem takes 11.5 ms to solve with an active set solver on a dual-core 2.9 GHz CPU. The facts that active set solvers are among the most computationally efficient ones, and that the control loop would run at 100-150 Hz, implied that no constrained optimal programming methods could be used in my control allocation procedure for the Revolve 2016 car.

An alternative to optimal programming is non-optimal linear unconstrained control allocation. The weighted pseudoinverse [21] was used in reference [19] to allocate wheel torques based on how much tire adhesion each wheel has left. As constraint satisfaction cannot be guaranteed by the weighted pseudoinverse method, the constraints have to be handled by an additional method. A promising candidate is the redistributed pseudoinverse [22] method where new pseudoinverse problems are solved until a satisfactory solution is acquired or no further improvement can be made.

Reference [19] used the following weighted pseudoinverse allocation strategy:

$$\tau = Bu \tag{11}$$

where $\tau = \begin{bmatrix} M_{z,ref} \\ F_{x,driver} \end{bmatrix}$, and $B = \begin{bmatrix} -l_1 & l_2 & -l_3 & l_4 \\ cos(\delta) & cos(\delta) & 1 & 1 \end{bmatrix}$,
and $u = \begin{bmatrix} F_{x,FL} & F_{x,FR} & F_{x,RL} & F_{x,RR} \end{bmatrix}^T$
where
$l_1 = \frac{t_f}{2} cos(\delta) - l_f sin(\delta)$,
$l_1 = \frac{t_f}{2} cos(\delta) + l_f sin(\delta)$,
$l_3 = l_4 = \frac{t_r}{2}$,
t_f and t_r is the front and rear track width, l_f is the distance from CoG to the centerline of front wheels and δ is the mean front steering angle.

To allocate forces to each wheel (i) the following expression was used:

$$u = W^{-1}B^T(BW^{-1}B^T)^{-1}\begin{bmatrix} F_{x,req} \\ M_{z,ref} \end{bmatrix}, \tag{12}$$

where W is a diagonal weighting matrix with elements based on longitudinal, lateral and normal tire forces

$$\frac{\sqrt{\hat{F}_{x,i}^2 + \hat{F}_{y,i}^2}}{\hat{F}_{z,i}}. \tag{13}$$

The weighting matrix therefore ensures that wheels are allocated torques based on their remaining adhesion capacity. However, there are some issues attached to using this strategy in a vehicle. One issue is that this method allows positive force to be allocated to a wheel when the driver is requesting zero force. This happens since the error between the sum of allocated forces and the requested force is minimized. If one wheel is allocated 500 N while another wheel is allocated -500 N the sum is zero. This is not allowed according to the rules in Formula Student. Another issue is that the strategy is based on accurate tire force estimation. If the control system is to rely on lateral and longitudinal tire force estimators, they must be able to produce completely stable, robust, and accurate values that are in sync at high g-maneuvers. I have not found any article where these properties have been demonstrated experimentally. Modifying the strategy to not use negative torque, and to rely on less knowledge about tire forces, solves the above mentioned issues, but constraint satisfaction is still not guaranteed. As I regarded the merging of logic with a pseudoinverse formulation to guarantee constraint satisfaction to be unnecessarily complicated for achieving a proper control allocation, I tried to come up with a simpler solution.

The 4WID-EV is an over-actuated system if you consider all four motors at the same time. In this case we have four actuators and because there are only two control variables it is an under-determined system. However, if the allocation problem is split into two independent problems, front wheels and rear wheels, we have two actuators and two controlled variables. A solution can therefore be found directly. As the amount of force a tire can exert is mainly governed by the normal tire force and the slip, it is natural to split the total requested torque and yaw moment based on the proportions of normal force that are experienced by the front and rear tires. This solution is more intuitive and can handle constraints in a more straightforward manner than pseudoinverse formulations. This allocation strategy is thus a solution that ensures constraint satisfaction and use of tire information based on a simplified measure of available friction force. This allocation strategy may therefore be viewed as a simple

and pragmatic engineering solution compared to more sophisticated strategies, but which is still capable of delivering a satisfactory control allocation.

9 Direct Yaw Moment control - systems design

9.1 Reference yaw rate

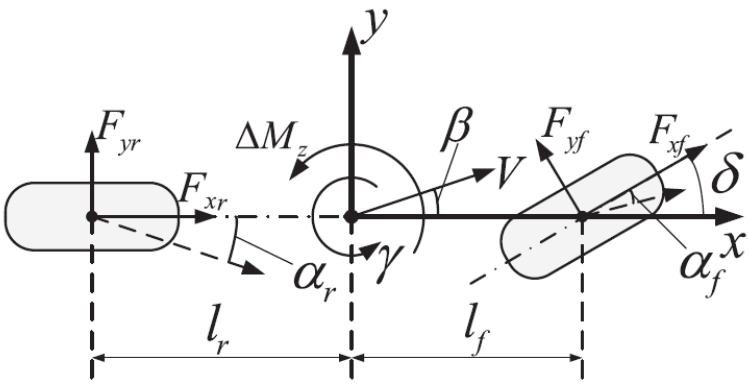


Figure 21: Schematic of the single track model for vehicle lateral dynamics [11].

A method to generate a desired yaw rate that, if tracked, leads to increased lateral performance compared to the baseline vehicle is a key feature of the control system. Among several options I chose to use a linearized two degrees of freedom planar vehicle model, often referred to as the bicycle model (Figure 21). By using this model the steady-state yaw rate response when navigating a circular road was found [1]:

$$\dot{\psi}_{des} = \frac{V_x}{(l_f + l_r) + K_v V_x^2}\delta. \tag{14}$$

Here δ is the front steering angle, and K_v is the understeer gradient defined according to reference [1] as:

$$K_v = \frac{ml_r C_r - ml_f C_f}{2C_f C_r L}, \tag{15}$$

where C_f and C_r are the front and rear tire cornering stiffnesses, l_r, l_f are distances from the center of gravity to the rear and front wheel centers, and L is the wheelbase.

By using the following formula for the steady-state steering angle required to navigate a turn of radius R with a longitudinal velocity V_x [1]:

$$\delta = \frac{L}{R} + K_v \frac{V_x^2}{R}, \tag{16}$$

the three main steering characteristics of the planar vehicle can be determined. If K_v is positive the vehicle is understeered, which means that as the vehicle velocity increases the steering angle must be increased to keep the same turning radius. If K_v is negative, the vehicle is oversteered, and the driver must reduce the steering angle as the velocity increases to keep the same turning radius. If $K_v = 0$ the vehicle is neutral-steered, which means that the turning radius is independent of velocity.

If the desired yaw rate, while cornering, can be achieved with an equal torque distribution between all wheels, then one is likely not utilizing the full potential of the control system. It is therefore very important to ensure that the desired yaw rate is higher than the baseline yaw rate of the uncontrolled vehicle. This implies that equation (14) must be experimentally tuned according to the baseline yaw rate dynamics of the vehicle by adjusting the value of K_v, as well as the wheelbase parameters, l_f and l_r.

The desired yaw rate calculated with eqn. (14) will not be achievable in many cases, however. To determine the actual achievable yaw rate based on the given vehicle state, I chose to make use of the maximum lateral acceleration $a_{y,max}$ the car can safely have at a specific tire-road friction coefficient [1]:

$$a_{y,max} \leq \mu g. \tag{17}$$

This leads to the following tunable expression for an upper bound on the desired yaw rate:

$$\dot{\psi}_{bound} = c\frac{\mu g}{v_x}, \tag{18}$$

where c must be experimentally tuned to achieve the desired behavior.

9.2 Yaw Rate Controller

The chosen PID based controller design will be outlined in this section. Gain scheduling based on longitudinal velocity was implemented as the yaw dynamics change based on speed. Anti-windup was needed to suppress the integrator term when the control allocation procedure is not able to achieve the reference yaw moment due to actuator saturation or tire adhesion limits.

No differentiation was done in the Simulink implementation of the controller due to how the system is designed. This is because the control system is executed at a set frequency regardless of when new sensor values arrive. Taking the derivative inside the yaw rate controller may thus lead to differentiating identical sensor samples,

depending on control system and sensor sampling frequencies, and communication delays. To ensure robustness, all derivatives in the system are therefore performed in sync with new sensor value arrivals to the system, which happens outside the control system code that is generated from the Simulink diagrams.

9.2.1 Gain scheduling

The lateral dynamics of the vehicle vary substantially with forward speed. Making a sharp turn with a low starting speed will use less lateral tire force compared to a sharp turn made at a higher initial speed. It therefore made sense to schedule the gains in the yaw rate controller based on longitudinal velocity, with aggressive yaw rate control at slow speeds and less aggressive at higher speeds.

9.2.2 Anti-Windup

The anti-windup of the integrator plays a significant role in the control system due to the choice of a simple PID controller. Many situations will arise where the control allocation is not able to satisfy the reference yaw moment. The integrator in the PI controller must then be winded down to avoid the integral term escaping into oblivion. I therefore chose an anti-wind up strategy using the difference between reference yaw moment and allocated yaw moment:

$$I = \int (K_i r_{error} - K_{sat} M_{z,error}) dt,$$

where K_i is integrator gain, $r_{error} = r_{ref} - r$, K_{sat} is the anti-wind up gain, and $M_{z,error} = M_{z,ref} - M_{z,allocated}$.

9.3 Control Allocation - a pragmatic engineering solution

Figure 22 shows the torque and power curves for the motors that were used in the Revolve 2016 car.

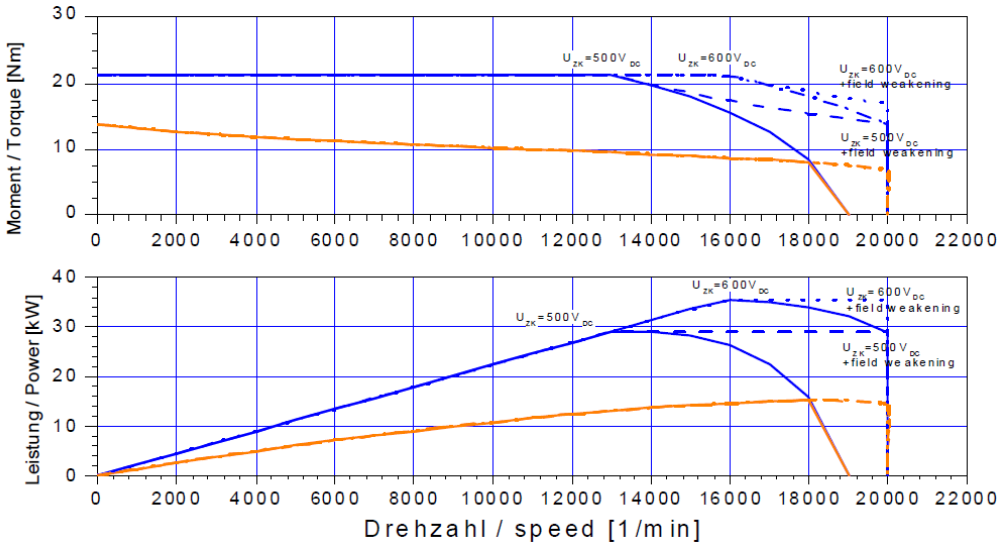


Figure 22: Torque and power curves for the motors that were used in the car. This figure is taken from the AMK motor datasheet.

The motors can have a constant torque output up until field weakening is reached. Field weakening is when the induced voltage in the motor is higher than the DC voltage. The phase currents can be manipulated in this region to increase speed and reduce torque. In the field weakening area, the output torque depends mainly on the available power and limitations caused by the physical components in the motor. It is critical that the control allocation procedure does not request torques that are higher than the achievable torque from the inverters in this region. If too much torque is requested, the control system in the inverter will become unstable. To calculate the allowed torque setpoint I developed the following formulas:

$$T_{m,front,max} = min(T_{m,max}, (\frac{V_{DC} \cdot Discharge_{max} \cdot (1 - \alpha_r)}{2 \cdot max(1, \omega_{FL}, \omega_{FR})}), \tag{19}$$

$$T_{m,rear,max} = min(T_{m,max}, (\frac{V_{DC} \cdot Discharge_{max} \cdot \alpha_r}{2 \cdot max(1, \omega_{RL}, \omega_{RR})}), \tag{20}$$

where $T_{m,max}$ is the maximum motor torque, V_{DC} is the measured battery voltage, $Discharge_{max}$ is the maximum current discharge from the battery, α_r is the normal force rear to front ratio, and ω is the measured angular velocity of the specified motor. The maximum torque for each of the front wheels are then given by $T_{m,front,max}$, and each of the rear wheels by $T_{m,rear,max}$.

The control allocation procedure must also be aware of the maximum torque the motors can output at all times. This information is needed to ensure that the cal-

culated added yaw moment to be applied by the control allocation procedure can actually be fulfilled by the motors.

The main goal of the control allocation procedure was to achieve as much of the reference yaw moment and torque request as possible under the given motor and tire constraints. Four control allocation algorithms were designed and implemented:

Name	Description
Static	Allocates 25 % of the torque request to each wheel.
Only positive torque allocation	Will only allocate negative torques if the driver actuates the regenerative brake pedal.
Full allocation	Can allocate negative torques to wheels to achieve the desired yaw moment.
Full allocation without Fz	Can allocate negative torques to wheels to achieve desired yaw moment, but does not use estimated tire normal forces.

The static, positive, and full allocation without Fz torque allocation algorithms were implemented in case I experienced problems with the sensor data required to run the full allocation algorithm. This section will only detail the full allocation algorithm.

Available tire adhesion is represented with the simple approximation $F_z \cdot \mu$. This approximation will overestimate the capacity of the tires during cornering, which was, as mentioned earlier, a desired feature. It was therefore up to the driver to stay at the limit of tire adhesion.

Table 3: Table of the four critical constraints that must be accounted for.

Name	Unit	Description
$T_{m,max}$	Nm	Maximum motor torque based on motor operating region and available electric power.
$T_{tire,max}$	Nm	Maximum motor torque based on tire adhesion limit.$T_{tire,max} = \frac{F_{z,i}\mu R_{eff,i}}{GearRatio}$.
T_{driver}	Nm	Requested total torque by the driver.
$T_{TV,des}$	Nm	Desired torque difference between left and right wheels.

Table 3 lists the most important constraints that must be directly accounted for by the allocation procedure. The first procedural step of this approach is to split the requested driver torque and reference yaw moment into two contributions, front and rear. Due to the relationship between traction and normal force, the proportion of normal force can be used to decide how much torque should be applied by the front and rear. Separating the problem into two two-wheel allocation problems also sim-

plifies the allocation procedure considerably. The rear to front ratio α_{rear} is calculated based on the amounts of normal force the front and rear are experiencing:

$$\alpha_{rear} = \frac{F_{z,RL} + F_{z,RR}}{F_{z,FL} + F_{z,FR} + F_{z,RL} + F_{z,RR}}. \tag{21}$$

The yaw moments to be distributed by the front and rear are:

$$M_{z,front} = (1 - \alpha_{rear}) \cdot M_{z,ref} \tag{22}$$

and

$$M_{z,rear} = \alpha_{rear} \cdot M_{z,ref}. \tag{23}$$

The requested torque from the driver is distributed by the same procedure:

$$T_{driver,front} = (1 - \alpha_{rear}) \cdot T_{driver}, \tag{24}$$

and

$$T_{driver,rear} = \alpha_{rear} \cdot T_{driver}. \tag{25}$$

The following steps shows how the torque setpoints for the front wheels are found with the full allocation algorithm. The procedure is identical for the rear wheels.

Step 1

Find motor torque limits and tire adhesion limits:

$$T_{FL,m,max} = f(P, RPM_{FL}), \tag{26}$$

$$T_{FL,tire,max} = f(F_{z,FL}, \mu), \tag{27}$$

$$T_{FR,m,max} = f(P, RPM_{FR}), \tag{28}$$

$$T_{FR,tire,max} = f(F_{z,FR}, \mu). \tag{29}$$

The torque limits for each wheel are given by:

$$T_{FL,max} = min(T_{FL,m,max}, T_{FL,tire,max}), \tag{30}$$

$$T_{FR,max} = min(T_{FR,m,max}, T_{FR,tire,max}). \tag{31}$$

The total difference in torque between the left and right wheels ($T_{TV,des}$) that is necessary for achieving the reference yaw moment is given by:

$$T_{TV,des} = abs(\frac{(1 - \alpha_r) M_{z,front} r_{eff}}{GR \cdot t_f}) + abs(\frac{(1 - \alpha_r) M_{z,front} r_{eff}}{GR \cdot t_f}), \tag{32}$$

where t_f is the front trackwidth, and r_{eff} is the effective wheel radius.

Step 2

The desired direction of turning is found by checking the sign of the yaw moment reference Mz_{ref}. If $Mz_{ref} \leq 0$, the yaw rate controller wants the car to turn right, and the left side should be allocated torque within the given limits first. The case when this is true is explored in the following steps.

Step 3

The front left and front right torque setpoints are found using the following code:

$T_{FL} = min(T_{TV,des}, T_{driver,front}, T_{FL,max})$;
if $T_{FL} < T_{TV,des}$ **then**
  $T_{FR} = -min((T_{TV,des} - T_{FL}), T_{FR,max}, negativeTorqueLimit)$;
else
  $T_{FL} = T_{FL} + min((T_{FL,max} - T_{TV,des}), T_{FR,max}, (T_{driver} - T_{FL})/2)$;
  $T_{FR} = min((T_{FL,max} - T_{TV,des}), T_{FR,max}, (T_{driver} - T_{FL})/2)$;
end

The above steps describe the basic torque distribution logic. To optimize the distribution further a logic layer on top of this was needed to ensure that as much as possible of the requested torque becomes distributed. Imagine that the vehicle is accelerating, giving the rear wheels 60% of the total normal forces. The rear wheels are allocated 60% of the requested torque, while the front wheels are allocated 40% of the requested torque. If the total requested torque is 84 Nm, the rear wheels are given $0.6 \cdot 84 = 50.4$ Nm. Assuming that the motors are only limited by the maximum torque they can achieve (21 Nm), $50.4 - 42 = 8.4$ Nm of torque is not distributed. The front wheels are in this case only allocated 33.6 Nm. Assuming that the front wheels can also use 42 Nm, then 8.4 Nm of torque is wasted if no further changes are made to the distribution of torque. This can be solved by implementing the following logic:

if $FrontAllocatedTorque < FrontRequestedTorque$ *AND*
$RearAllocatedTorque == RearRequestedTorque$ **then**
  $TorqueRemaining = RequestedTorque - FrontAllocatedTorque$;
  $MzRemaining = MzReference - FrontAllocatedMz$;
  $RedistributeRearTorque(TorqueRemaining, MzRemaining)$;
else if $FrontAllocatedTorque == FrontRequestedTorque$ *AND*
$RearAllocatedTorque < RearRequestedTorque$ **then**
  $TorqueRemaining = RequestedTorque - RearAllocatedTorque$;
  $MzRemaining = MzReference - RearAllocatedMz$;
  $RedistributeFrontTorque(TorqueRemaining, MzRemaining)$;

10 Direct Yaw Moment control - results

10.1 Simulation results - DYC

Figures 23 and 24 show steer step tests without and with PID yaw rate control active. Without yaw rate control activated the vehicle yaw rate does not track the ideal yaw rate reference at all.

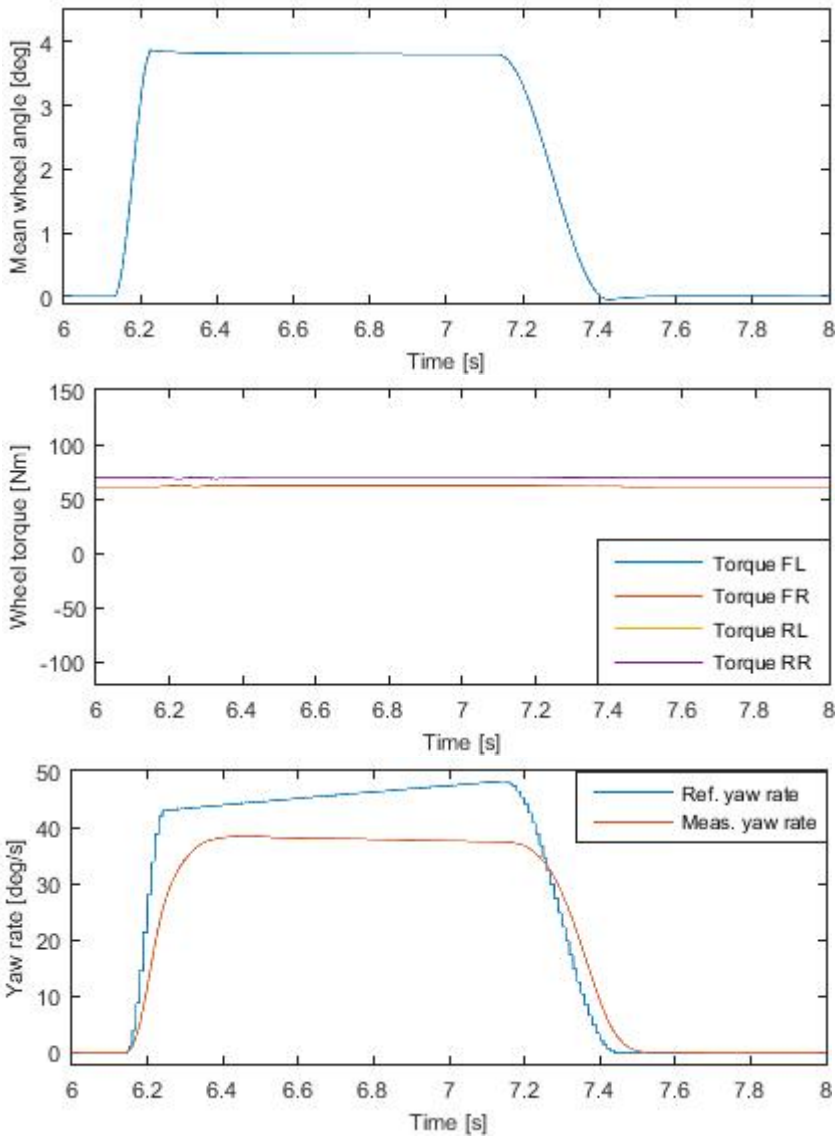


Figure 23: Steer step test without yaw rate control.

In contrast, with yaw rate control one can see that the yaw rate tracks the reference yaw rate very well.

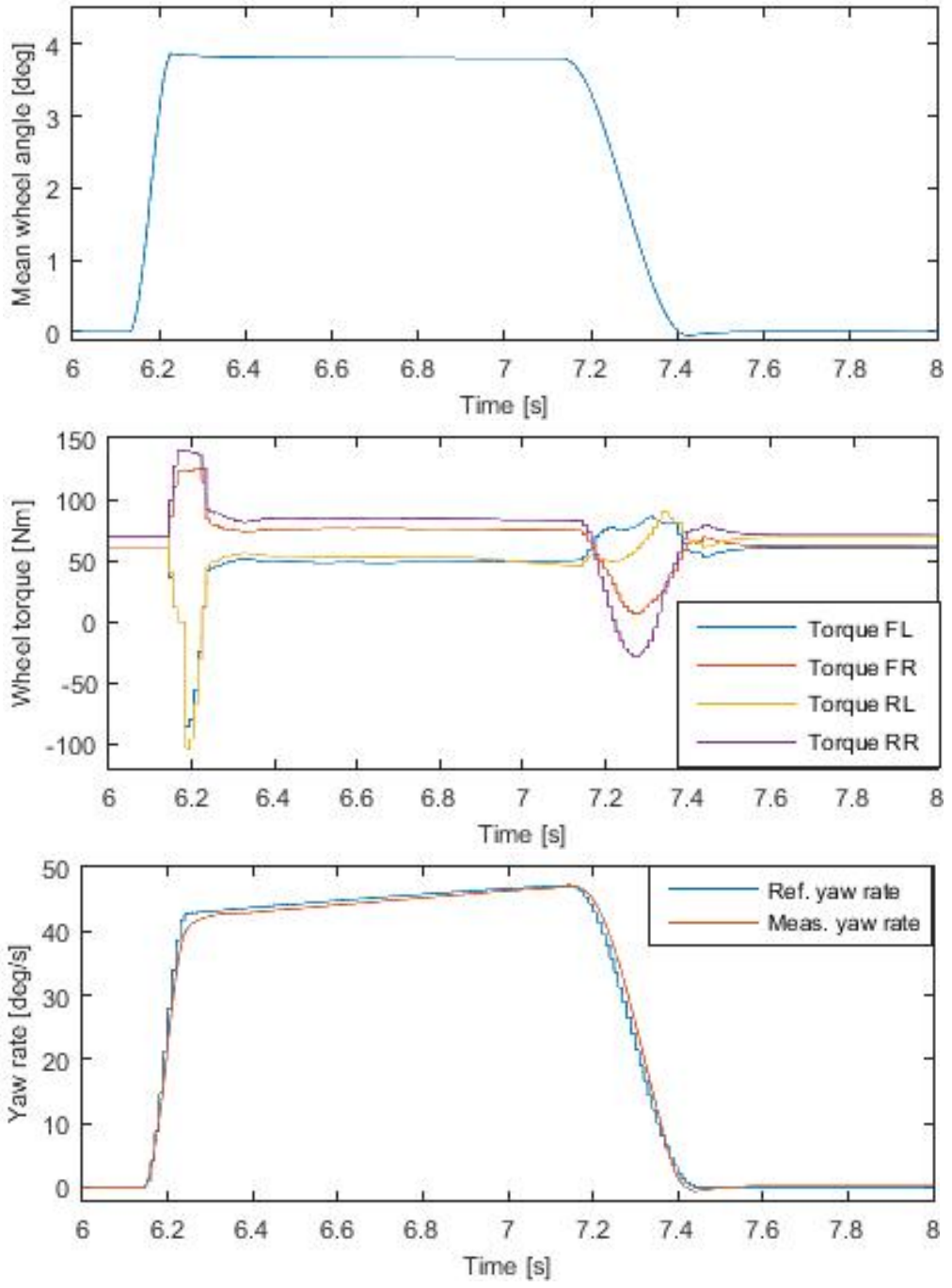


Figure 24: Steer step test with yaw rate control.

10.2 Experimental results - DYC

Figure 25: Quiet evening testing of Gnist at Vaernes Airport.

The feedback on the direct yaw moment control from the drivers has been very positive. They said that the controllers in Gnist helped them to both corner and accelerate/decelerate better than cars they have driven before. Here are a couple of quotes; "It really feels like the car pulls me around corners", "It is impossible to spin around in this car".

Figures 26 and 27 below show yaw rate and motor torques for a skid pad run (figure of eight with two rounds in each circle). PI yaw rate control was active for this run. Skid pad is all about pushing the vehicle to the traction limit to achieve the highest yaw rate and lateral acceleration possible. The right cornering circle part of the data (negative yaw rate) looks different to the left cornering circle due to the driver being more comfortable with turning right, and is therefore better able to push the vehicle to the limit in right corners. Looking at the part of the graph with negative yaw rate, it can be seen that the reference yaw rate is reduced in magnitude. This is caused by the driver adjusting the steering wheel angle and/or the torque pedal. Looking at the motor torque graph one can clearly see that more torque is allocated to the outer wheels than the inner wheels. To help the driver achieve steady state stability,

the controller gains for skidpad were intentionally set quite low, with most focus on the integral part. The yaw rate tracking is therefore much slower than for example with autocross settings. This parameter set was reached based on driver feedback and timing of skidpad runs.

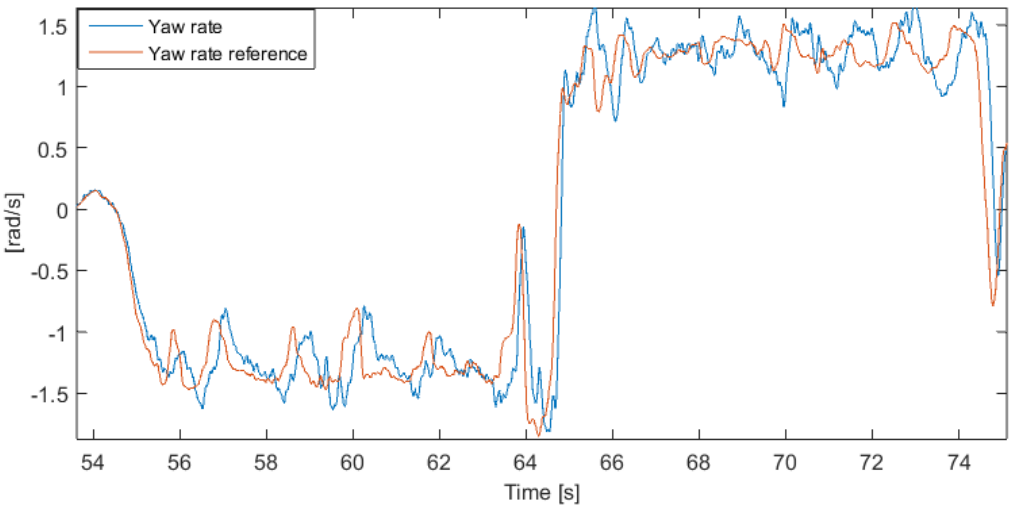


Figure 26: Yaw rate

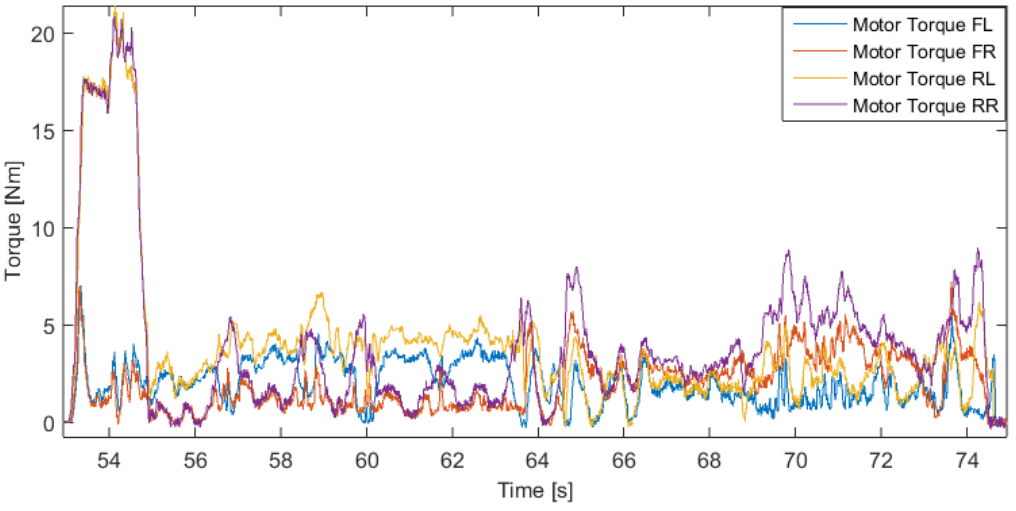


Figure 27: Motor torques for a skidpad run.

I successfully managed to implement a derivative term based on the reference value in the yaw rate controller, which made the car more responsive to fast steering inputs. The drivers said that the derivative term helped them corner faster, and it was

therefore active for the autocross and endurance events. More specifically, Figures 28 and 29 show yaw rate and motor torque data from 35 seconds of autocross driving, and one can see how motor torque is constantly being allocated in a way that makes the vehicle yaw rate track the yaw rate reference, which it does fairly well. The yaw rate reference is often larger than the yaw rate in magnitude, as no limitations are imposed on the yaw rate reference calculation, and the reference value can be impossible to achieve through torque allocation. I chose to not impose any limits on the yaw rate reference mainly because I had no way of accurately calculating attainable yaw rate reference values, and because I did not want to limit the lateral performance of the car by imposing inaccurate yaw rate reference limits.

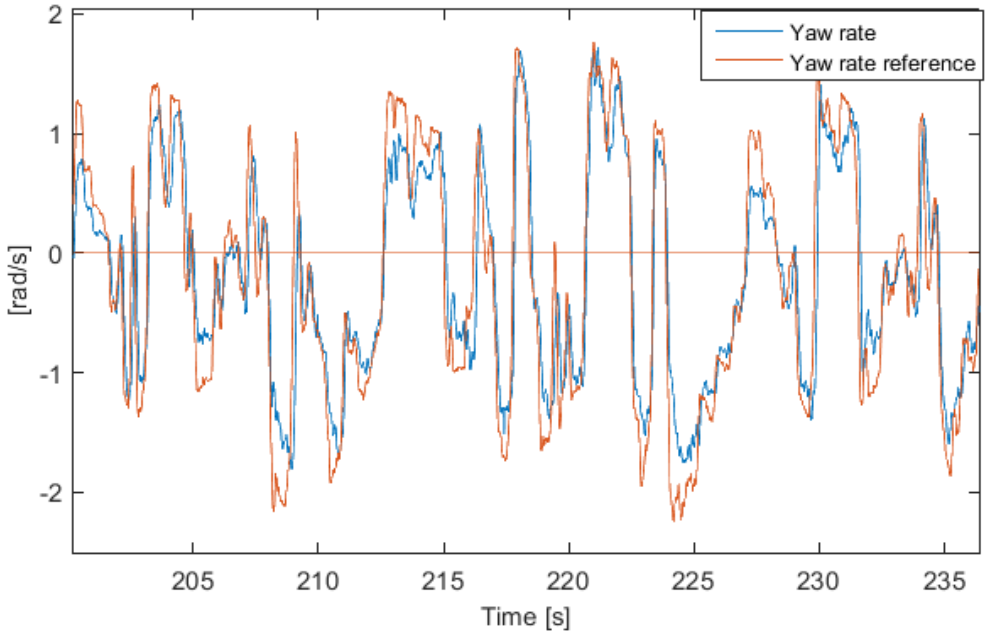


Figure 28: Excerpt of yaw rate and yaw rate reference from autocross driving.

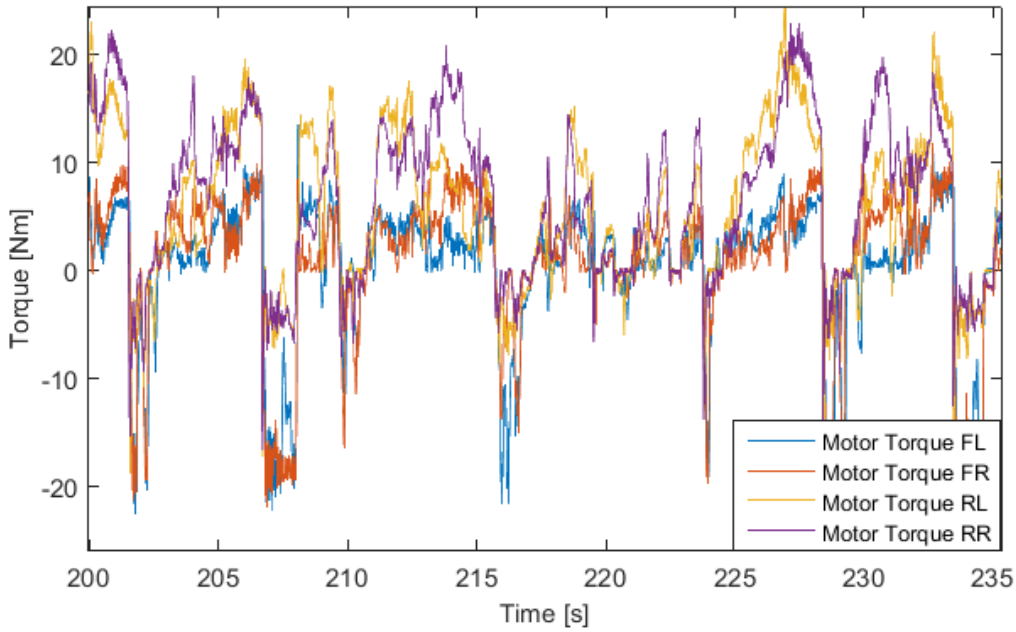


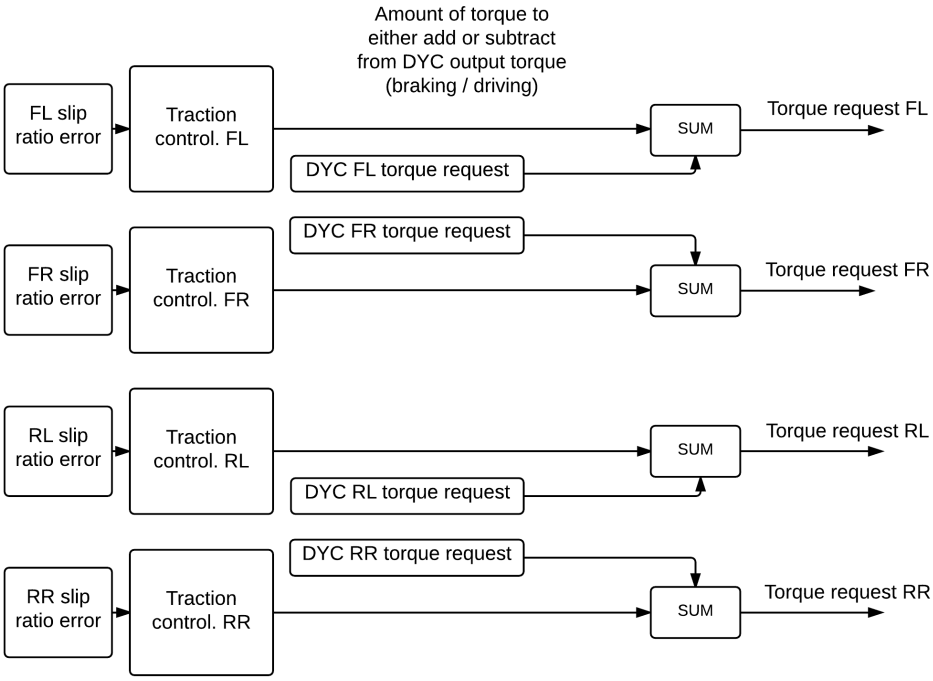
Figure 29: Excerpt of motor torques from autocross driving. Motor torque is calculated directly from the torque current, which is calculated by the inverters.

10.3 Discussion - DYC

The yaw rate controller was not thoroughly tuned and performance tested as I was not allocated enough time to do this due to a number of practical issues with other parts of the car that had to be solved first. So I do not know what the "optimal gains" are for the different events. Gain scheduling based on longitudinal vehicle velocity was also not activated due to lack of time. More tuning and activation of gain scheduling are most likely to improve the yaw tracking performance. Tuning a system such as this is extremely complex, as the human element is so strongly present. The controller is perfectly tuned for a driver when he is able to achieve his fastest lap times. Different drivers also have different car behavior preferences, which implies the need for driver specific parameter tuning. Tuning the system must therefore be conducted based on tracking performance, driver feedback and lap times. Due to the time constraints I therefore had to mainly make use of driver feedback when tuning the system this summer. Considering this restriction it is encouraging that a driver who also drove our previous RWD car, Vilje, clearly concluded that Gnist felt a lot faster and easier to handle.

11 Traction control - design and results

The system I have designed combines traditional traction control and anti-lock braking system (ABS) into one system, as I can control both driving and braking forces from the motors. The traction control overview (Figure 19) is depicted once more to ease the reading of this section.



Traction control system overview.

Torque requests from the DYC system are passed into the TC system, and modified if necessary in separate traction control blocks for each wheel. The surfaces Gnist drives on are so friction uniform that no stability measures in the TC system are needed. In addition to being able to use the motors as brakes, Gnist is equipped with friction brakes. The driver can use both of these separate braking mechanisms with the two-stage brake pedal in Gnist. Actuation within the first stage determines the magnitude of the negative torque request, which is also referred to as regenerative braking. Actuation within the second stage controls the magnitude of friction braking. The only tunable parameter for the friction brakes is the brake balance between the front and the rear. A friction brake based anti-lock braking system (ABS) weighs so much that it is not viable for Formula Student cars. As the friction brake pressure can-

not be modulated electronically, the prevention of exceeding the ideal negative slip ratio had to be achieved by deliberately unifying the total motor and friction braking efforts through control of the amount of motor braking only. This could not be achieved at all times with the given configuration, as the friction brakes on Gnist are powerful enough to completely lock the wheels at 50 $\mathrm{km\,h^{-1}}$.

11.1 Reference slip ratio

A typical definition of longitudinal slip, also called slip ratio is:

$$SR = \frac{\omega r_{eff} - v_x}{max(\omega r_{eff}, v_x)}, \tag{33}$$

where ω is the angular speed, r_{eff} is the effective wheel radius, v_x is the longitudinal vehicle velocity at the wheel hub. At the front wheels the wheel angles are included to find the equivalent longitudinal velocity. The optical speed sensor which measures longitudinal velocity is mounted at the rear left of the car. To find the equivalent vehicle longitudinal velocity at each wheel, the yaw rate contribution must be included. I chose to first transform the v_x measurement to the same point where the INS measures the yaw rate. This value was then transformed out to each wheel to correct for the yaw rate contribution, which gave the following equations:

$$\begin{aligned}
v_{x,FL} &= v_{x,CoG} - \sqrt{x^2 + y^2} \cdot r \cdot sin(atan(\frac{y}{x})),\\
v_{x,FR} &= v_{x,CoG} + \sqrt{x^2 + y^2} \cdot r \cdot sin(atan(\frac{y}{x})),\\
v_{x,RL} &= v_{x,CoG} - \sqrt{x^2 + y^2} \cdot r \cdot sin(atan(\frac{y}{x})),\\
v_{x,RR} &= v_{x,CoG} + \sqrt{x^2 + y^2} \cdot r \cdot sin(atan(\frac{y}{x})).
\end{aligned} \tag{34}$$

Here x and y are the distances to each wheel hub and r is the measured yaw rate. The values for x and y are identical for each wheel as the INS is positioned in the center of the vehicle. The final slip ratio equations are thus given by:

$$\begin{aligned}
SR_{FL} &= \frac{\omega_{FL} r_{eff} \cdot cos(\beta) - v_{x,FL}}{max(\epsilon, \omega r_{eff}, v_{x,FL})},\\
SR_{FR} &= \frac{\omega_{FR} r_{eff} \cdot cos(\beta) - v_{x,FR}}{max(\epsilon, \omega r_{eff}, v_{x,FR})},\\
SR_{RL} &= \frac{\omega_{RL} r_{eff} - v_{x,RL}}{max(\epsilon, \omega r_{eff}, v_{x,RL})},\\
SR_{RR} &= \frac{\omega_{RR} r_{eff} - v_{x,RR}}{max(\epsilon, \omega r_{eff}, v_{x,RR})},
\end{aligned} \tag{35}$$

where epsilon is a small constant.

11.2 Effective wheel radius

The effective wheel radius is not constant while driving, and will vary mainly based on tire pressure and vertical force. It is important to have information of the effective wheel radius as it is used to calculate the slip ratio, which is the controlled variable in the TC system. The vertical damping of the tires at our chosen tire pressure is 0.0071 mm N^{-1}. The difference in effective wheel radius for a wheel loaded with 200 N and 800 N is 2%. I therefore considered setting the effective wheel radius as a constant to be accurate enough for my purposes.

11.3 Individual controllers

Figure 30 shows a traction control block for one wheel:

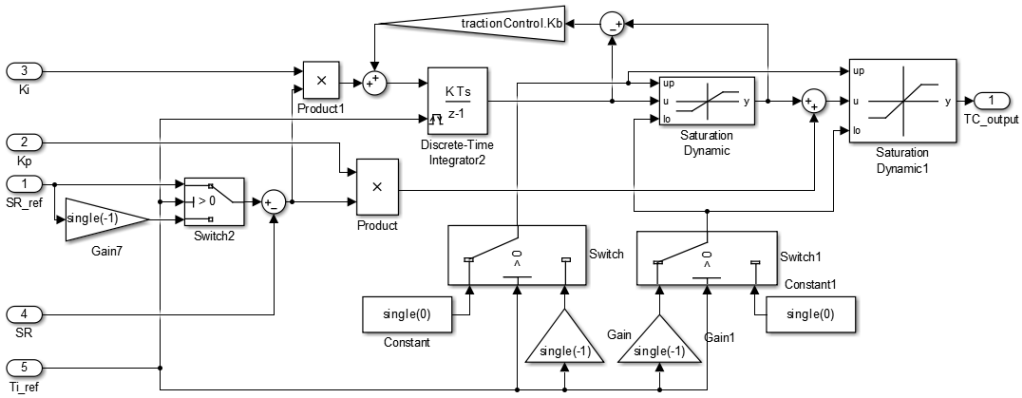


Figure 30: Traction control system overview. $T_{i,ref}$ is the requested torque from the DYC system.

The reference slip ratio is positive if the DYC requested torque is positive, and negative if the requested torque is negative. It is important to note that the controller will only subtract torque in case of exceeding the positive reference slip ratio, and only add torque (reduce amount of regenerative braking) if exceeding the negative reference slip ratio. To make the controller work for both braking and driving scenarios the upper and lower saturation limits of the anti-windup and output saturation blocks were set based on the sign of the DYC torque request. The integrator part was winded down by adding the difference between the output of the first saturation block and

the integrated input to the integrator input. The integrator was reset to zero each time the torque reference crossed zero, which makes the integrator response faster when crossing from acceleration to braking. The output from each TC block is thus how much should be added or subtracted from the DYC requested torques.

The proportional and integral gains were implemented as linear functions of vehicle speed in driving mode, and as constants in braking mode. In driving mode, the slip ratio changes rapidly to its maximum value, 1, when the car is accelerating from standstill. It is therefore beneficial to have lower gains at very low speeds and higher gains at higher speed. An additional gain scheduling based on requested torque was used to handle a normal torque request curve (ramp signal). Almost any torque request from standstill will cause the slip ratio to reach its maximum value immediately. This gain scheduling ensures that if the torque pedal is being ramped, traction control will not regulate the small torque request to zero in an oscillatory manner.

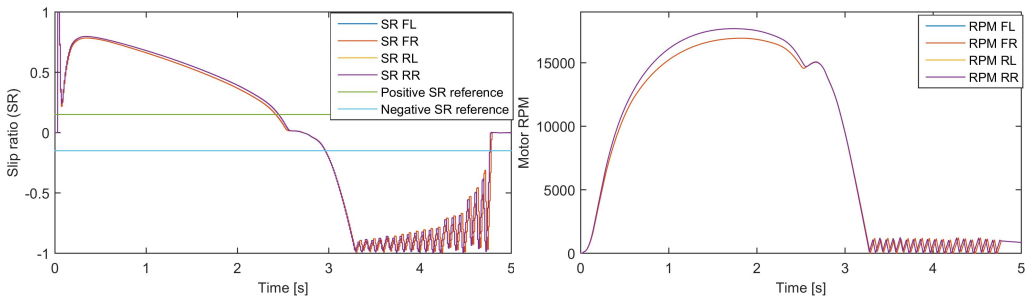
The outputs from all the TC blocks were just added to the DYC requested torques. During testing I discovered that it was not necessary to merge the TC outputs in any way to ensure vehicle stability. To achieve stability turned out to be no problem with decoupled traction controllers for each wheel, which simplified the system by eliminating the need for introducing extra logic.

11.4 Simulation results - traction control

The premises underlying these traction control simulation results were:

- Driver requests maximum driving torque until the car has reached a speed of 80 $\mathrm{km\,h^{-1}}$,
- When the car reaches 80 $\mathrm{km\,h^{-1}}$, the driver requests maximum regenerative torque until the regenerative brake speed limit is reached,
- Friction coefficient corresponding to wet asphalt

Figure 31 shows a run with no traction control enabled where one can see that all wheels are spinning for the duration of the forward acceleration.



(a) Slip ratio

(b) Motor RPM

Figure 31: Slip ratio and motor RPM without traction control.

In addition to the PI controllers that control the slip ratio, the slip ratio can be controlled by limiting the maximum allowed torque based on available friction. Figure 32 and 33 show the slip ratio and torque at each wheel when the maximum allowable torque exceeds the available friction. Figure 34 and 35 show the slip ratio and torque at each wheel when the maximum allowable torque is more in line with the available friction. It is easier to control the slip ratio when the allowable torque is more in line with available friction, which is reflected in the more accurate response seen in Figure 34.

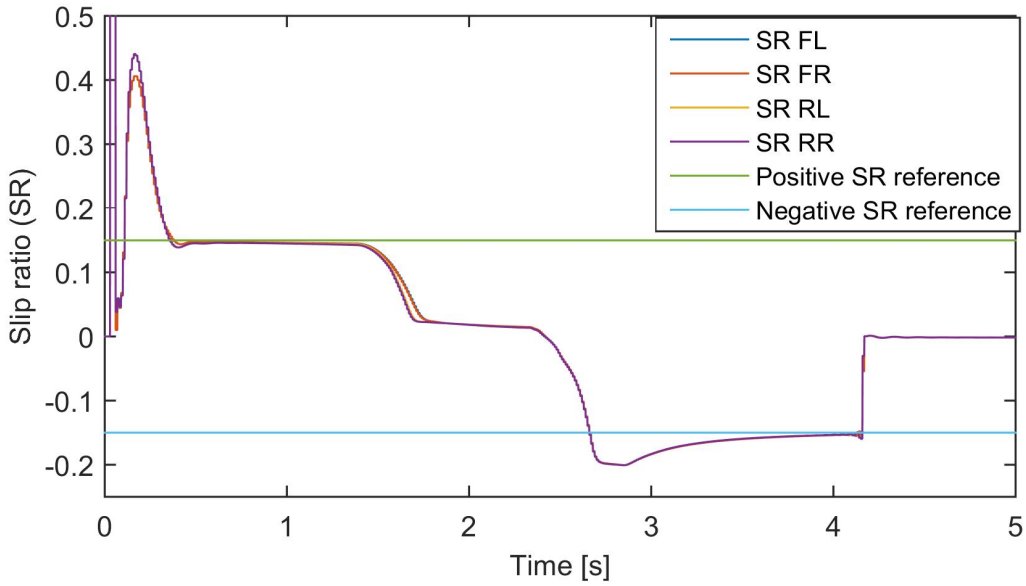


Figure 32: Torque response with traction control and incorrect available friction estimate.

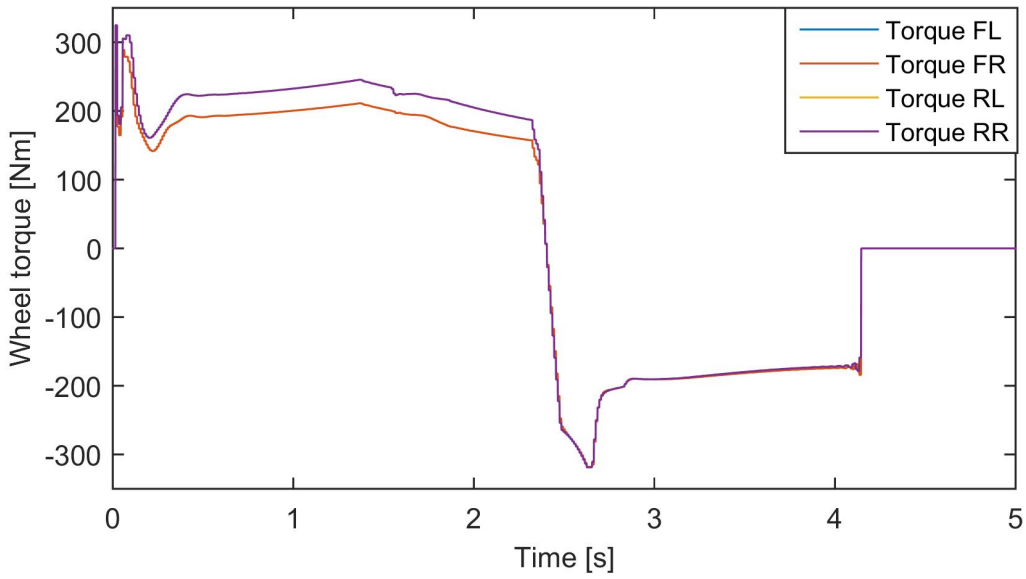


Figure 33: Slip ratio response with traction control and incorrect available friction estimate.

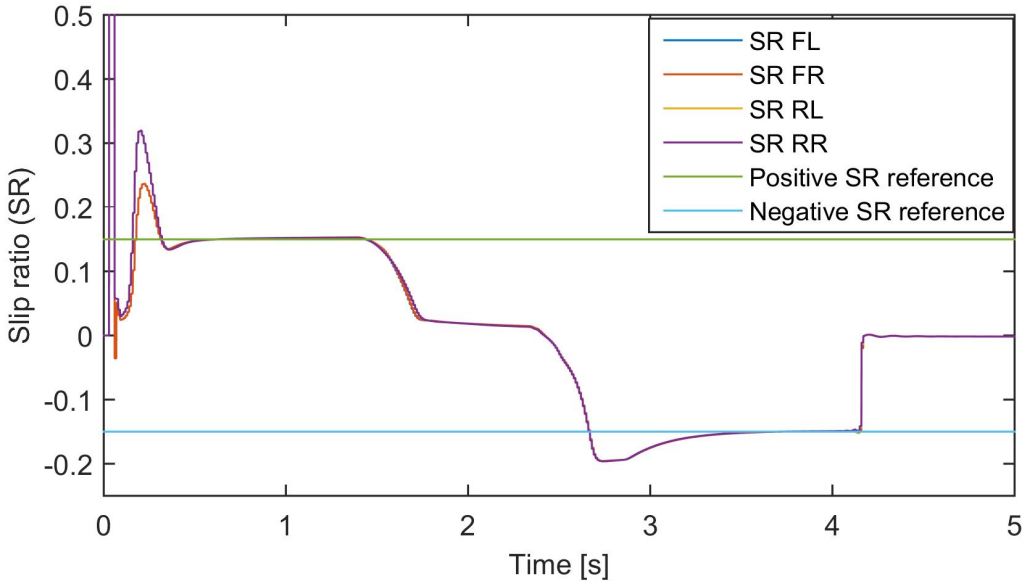


Figure 34: Slip ratio response with traction control and a reasonable friction estimate.

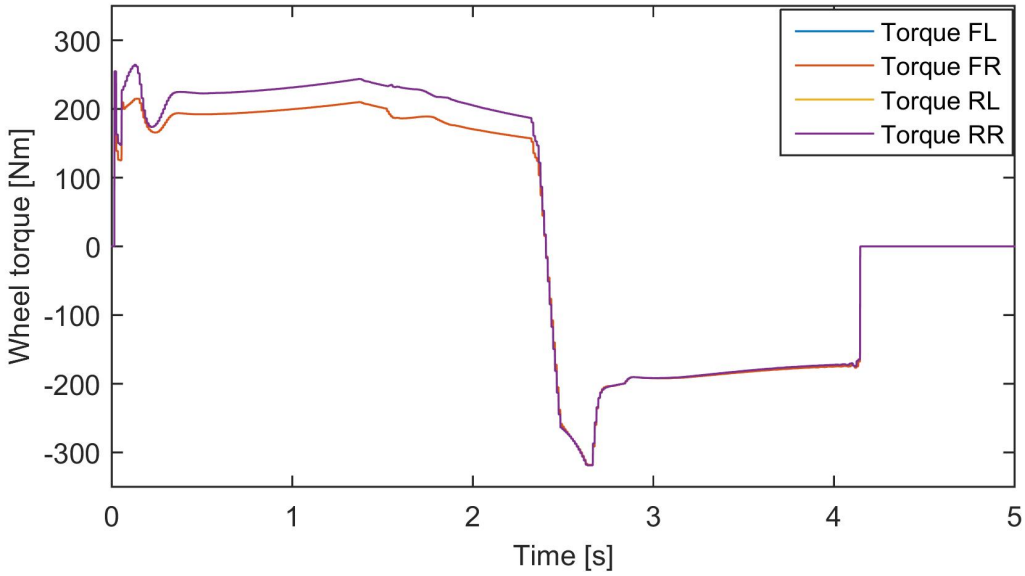


Figure 35: Torque response with traction control and a reasonable friction estimate.

Figure 35 shows that the torques at the front and rear wheels are predicted to be almost the same. In reality this is not possible due to the extreme weight transfer that happens when a car accelerates hard. The parameters governing the impact of weight transfer must therefore be looked at in future work.

11.5 Experimental results - traction control

Two types of traction control were combined and employed in the experimental tests. The first limited torque according to available traction for each wheel, based on $F_z \cdot \mu$. The other used a PI controller for each wheel. The driver requested maximum torque the whole time, and it was thus up to the traction controllers to limit the torque in a way that utilized available traction. While testing traction control I discovered that trying to control the torques during the first 300-500 ms of an acceleration from 0 $\mathrm{km\,h^{-1}}$ with the PI controllers did not work very well. Gain scheduling in the PI controllers was therefore made use of to make the PI controllers very weak in the beginning and increase control strength with vehicle speed. The first part of the acceleration was thus handled by the $F_z \cdot \mu$ term only. Figure 36 shows how extreme the weight transfer is at the start of an acceleration. The drop in normal force at the front wheels leads to a drop in motor torques, seen in Figure 38. After a second, the gains in the PI traction controller are strengthened due to the increase in vehicle speed, and the slip ratios of the traction limited front wheels are regulated to the slip ratio reference.

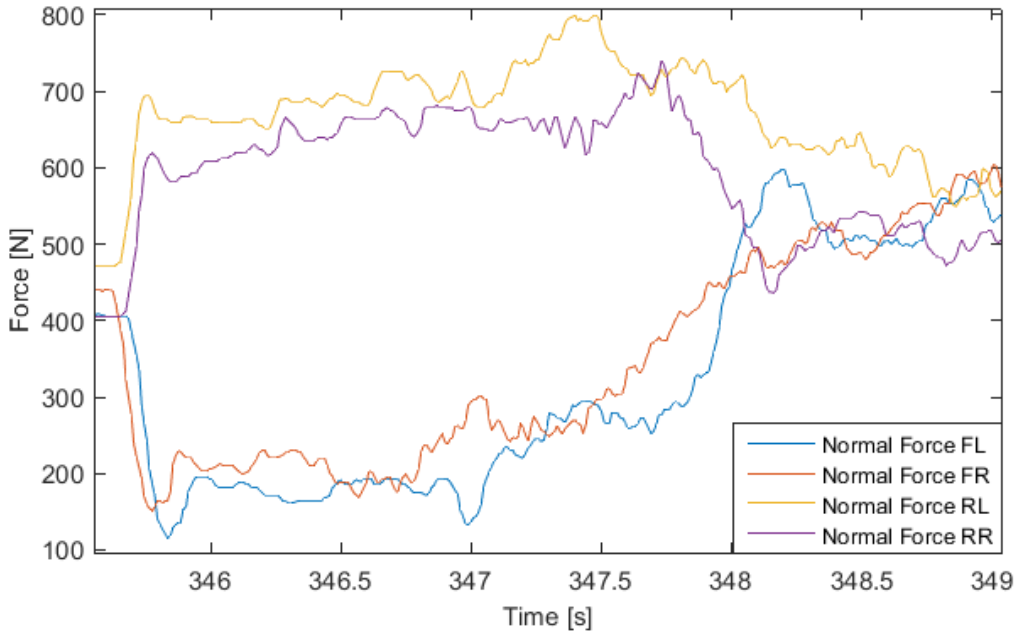


Figure 36: Normal forces on each wheel during an acceleration from 0 to 65 $\mathrm{km\,h^{-1}}$.

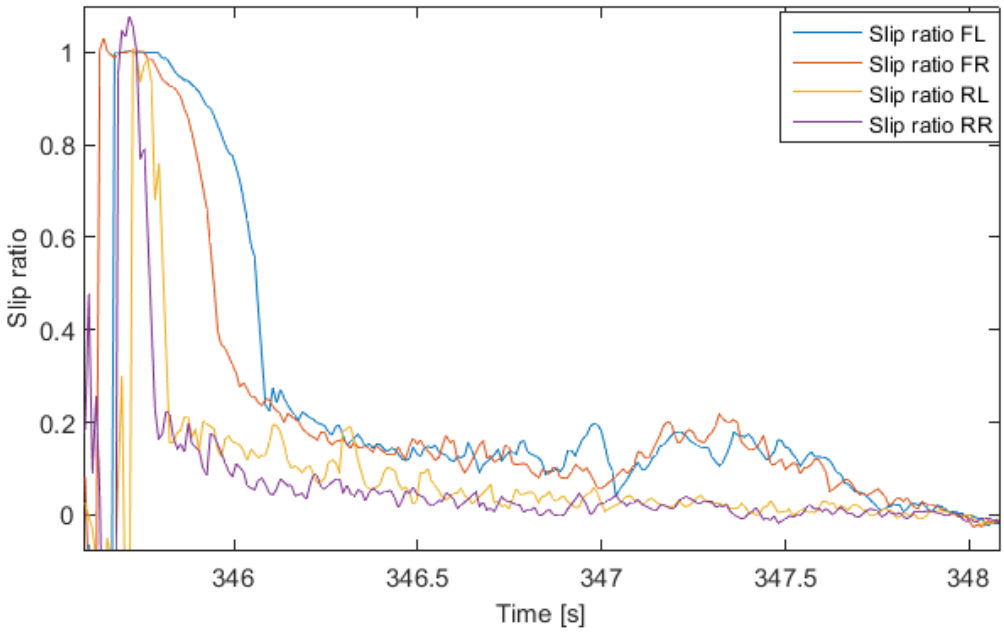


Figure 37: Slip ratios during an acceleration from 0 to 65 $\mathrm{km\,h^{-1}}$.

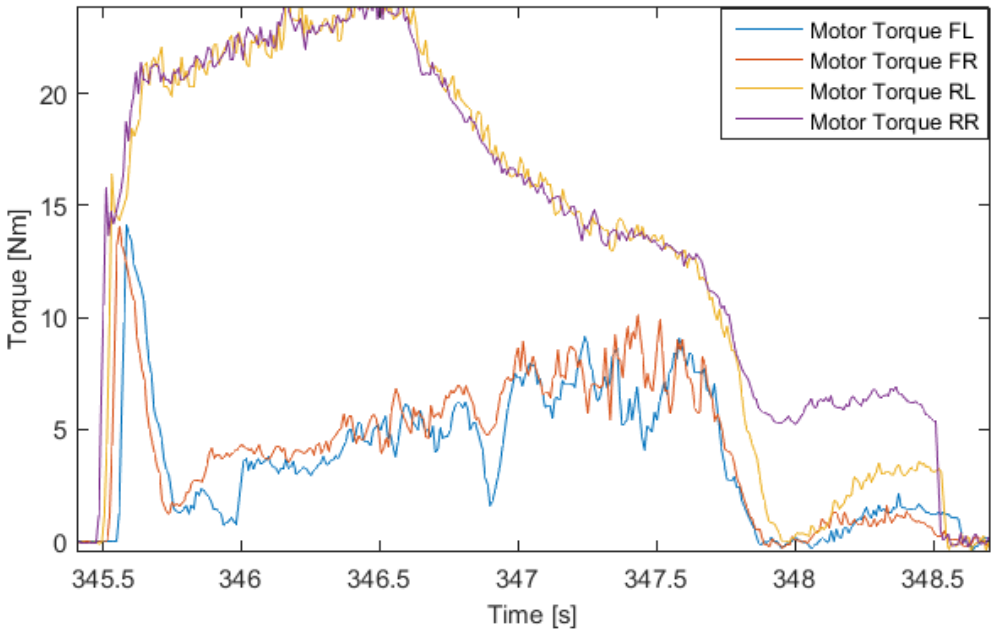


Figure 38: Motor torques during an acceleration from 0 to 65 $\mathrm{km\,h^{-1}}$.

11.6 Discussion - traction control

The traction control system has worked well. The combination of PI controllers and torque limiting based on available traction ensured that the system could handle a wide range of challenges such as from standstill acceleration, wheel lifts in high speed corners and regenerative braking. However, there is of course room for improvement. It is likely that there is still more performance to gain from tuning the current system, but the time slots I were given for this tuning did not allow me to pursue this in a thorough way.

In a future version of the traction control system, the controller gains for the front and rear should be separated to make it easier to tune acceleration from a standstill position. Moreover, the effects of introducing a derivative term as well as using the speed controller in the inverters to control the slip ratio should also be investigated by the 2017 Revolve team.

12 Power limit control - design and results

The Formula Student Silverstone competition imposes a power limit of 60 kW for 4WID-EVs, while Formula Student Austria and Germany impose a power limit of 80 kW. The power is measured by an energy meter connected directly to the battery output. At Silverstone "A violation is defined as exceeding the power limit or exceeding the specified voltage (600 V) for more than 100 ms continuously or exceeding the power limit or exceeding the specified voltage, after a moving average over 500 ms is applied.". In Germany and Austria a violation is defined the same way except for the 100 ms part.

To limit the power, the torque requests sent to the motor controllers can be reduced. This reduction can happen as part of the control allocation strategy, or it can be done externally as an equal reduction to each torque request. As torque is distributed based on available friction force, an equal reduction of the torque requests ensures that the tires with the most available friction force are still allocated the most torque. This method simplifies the system as the power limiting does not need to be a part of the control allocation. To utilize this method, information about the actual combined power usage of all motors must be available, which the battery management system built by the Revolve 2016 team was able to satisfactorily measure.

12.1 PI controller

I chose to use a PI controller as overshoots are allowed in the competitions, and Figure 39 shows the implementation of this controller.

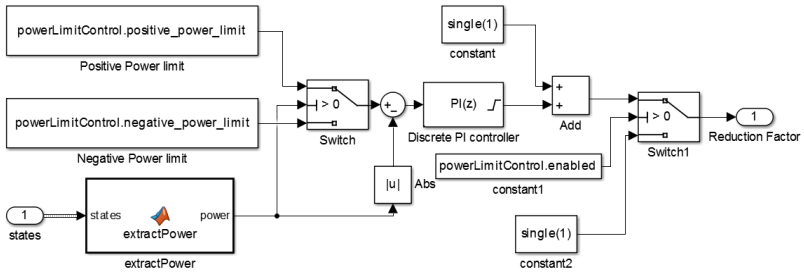


Figure 39: The PI power limit controller.

However, a PID controller may be better suited for this purpose, but fast dynamics of the power and the noise in the power measurements can potentially exclude the use of a derivative term. This should be analyzed more thoroughly by the Revolve 2017 team.

Integrator clamping was used along with an upper saturation limit of 0 to avoid the controller to increase the torque requests. The lower saturation limit will decide the lower limit of the torque request reduction factor. Simulations showed that the maximum reduction factor needed to stay within the power limit is approximately 0.7. The lower saturation limit was therefore set to -0.3. This controller worked in parallel with the control allocation procedure and reduced torque equally when active.

12.2 Simulation results - power limit control

Figure 40 shows the power usage during maximum acceleration over 75 meters with traction control enabled and power limit control disabled. This clearly demonstrates the need for power limit control in order to comply with the rules.

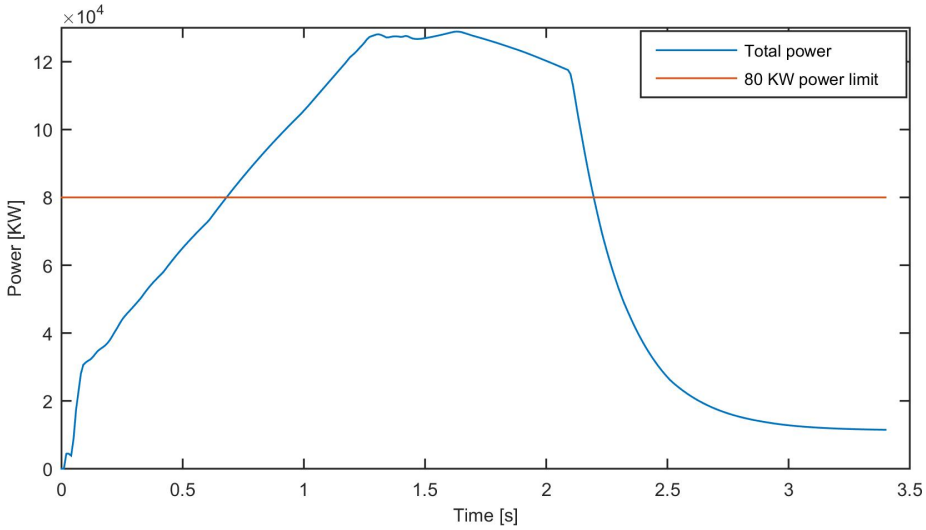


Figure 40: Power usage during maximum acceleration over 75 meters with traction control enabled and power limit control disabled.

Figure 41 shows the same maneuver, but with power limit control enabled. The reference positive power limit is set to 79 kW to ensure a small safety margin. The figure shows that the power usage stays within the allowed domain.

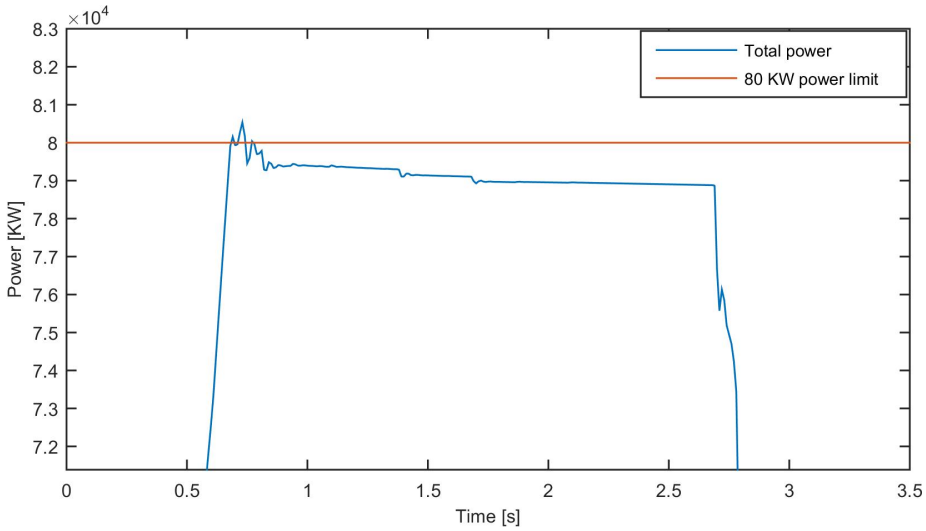


Figure 41: Power usage during maximum acceleration over 75 meters with traction control and power limit control enabled.

12.3 Experimental results - power limit control

Figure 42 shows the power usage for a 75 meter acceleration run, with PI power limit control enabled. It turned out that the power limit had to be set to 57.4 kW to ensure that the power usage did not exceed the 60 kW limit for more than 100 ms at Silverstone. Figure 43 shows the power usage during a straight section of an autocross track at Silverstone, UK.

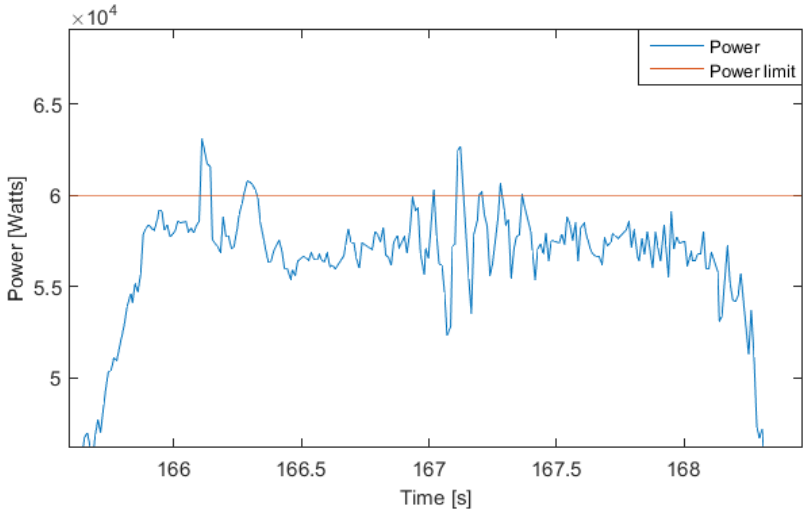


Figure 42: Power usage for an acceleration run over 75 meters with a 60 KW power limit with a PI controller.

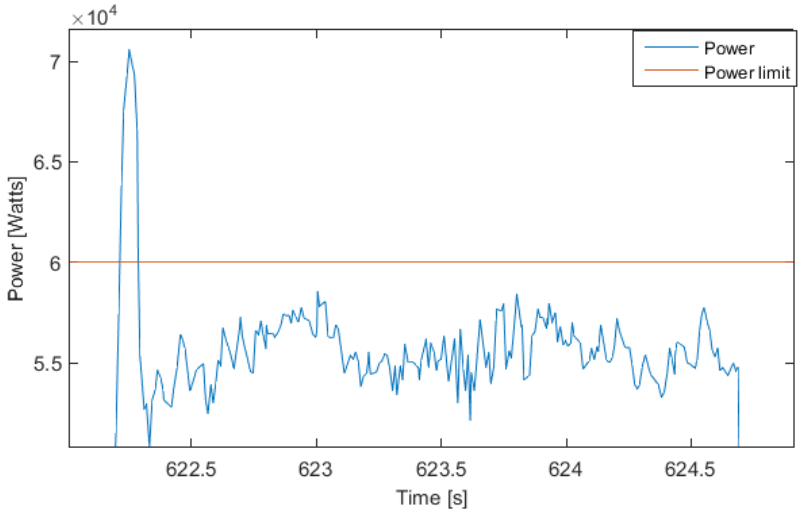


Figure 43: Power usage during a straight section of an autocross track with a 60 KW power limit.

An overshoot can be observed, which is to be expected for a PI controller. The PI power limit control at a 60 KW limit left some power unused when rule compliance had to be guaranteed, which can be observed from both graphs. However, the PI controller worked well with a 80 KW limit and no 100 ms requirement. As mentioned, two out of the three competitions Revolve competed in during the summer of 2016 imposed a 80 KW limit. At these competitions, a power limit violation was defined as exceeding 80 kW after a 500 ms moving average filter had been applied to the data. This made it much easier to stay within the allowed domain. Figure 44 shows power data which has been filtered with a 500 ms moving average filter, where a 79.5 kW limit is used by the power limit controller.

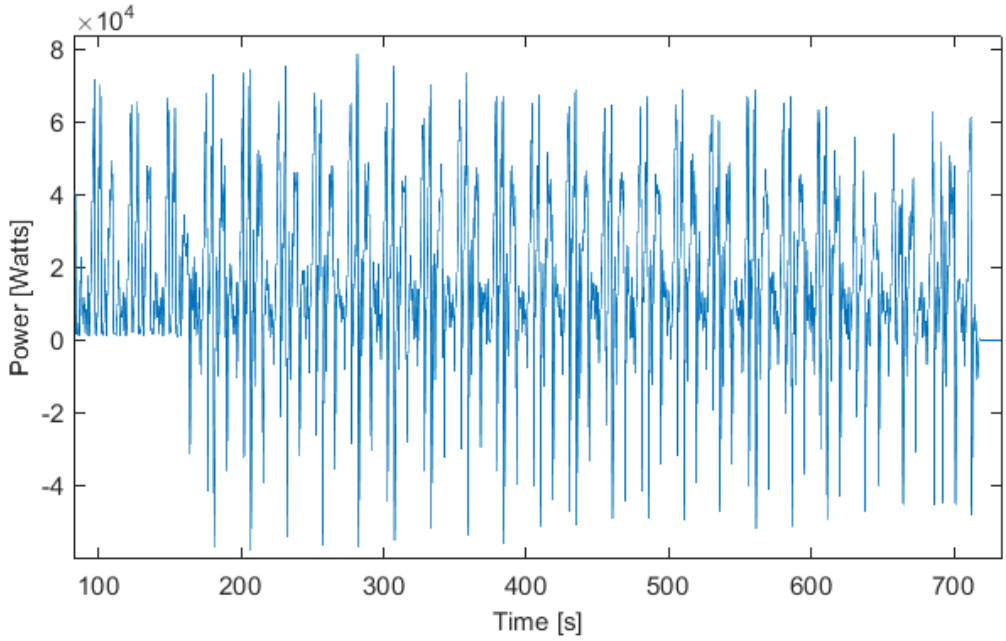


Figure 44: 500 ms moving average filtered power data from autocross testing.

12.4 Discussion - power limit control

The PI power limit controller performed satisfactorily at 60 kW with Silverstone rules, but it should be improved before the 2017 competition at Silverstone to get the power usage even closer to the 60 kW power limit and to reduce the magnitude and the duration of overshoots. When I designed the power limit controller I chose not to introduce a damping derivative term, as I was sceptical about using the derivative of the current measurement due to noise. This might be unfounded, and should be investigated to potentially reduce the overshoot problems. Instead of using torque reduction to reduce power usage, it might be better to proactively control the power

usage. This can be done by estimating the efficiency factor for the motors, and setting a limit for how much mechanical power that can be allocated to the motors. I have done some simulations on a controller of this type, and the results are promising (not shown).

13 Overall discussion and conclusion

All the control systems worked well from the first day they were tested in Gnist at Vaernes Airport. The deliberate effort to reduce complexity as much as possible during the design phase, thorough sensor data verification, HIL simulations, code generation with Matlab Simulink, and a well planned C-code implementation are the main reasons for this. Considering that I know many teams have not managed to make their control systems work the first year, and that a driver from a team with four years of 4WID-EV experience who tried Gnist said the control systems were on par with their second year implementation, I was naturally more than pleased by achieving this. But the most encouraging verdict came from the Revolve team drivers. Their praising of how easy it was to control the vehicle with these systems activated was extremely encouraging. The drivers especially enjoyed having yaw rate control with regenerative braking, which means that they could steer the car while they were braking.

However, it should be noted that the drivers pointed out that the torque vectoring system limited torque too much when they exited tight corners, which is a side effect of prioritizing yaw rate control over longitudinal acceleration. The acceleration from the apex of a turn to outside of the turn was therefore not as high as it could have been. To solve this the driver must be able to override the amount of torque allocated to achieve the reference yaw moment. This can be achieved by first letting the control allocation procedure attempt to apply torques which satisfy the yaw moment request, and thereafter allow as much torque as the driver is requesting within the individual tire and motor constraints. This improvement should be quite straightforward to implement in the 2017 car. Despite this shortcoming I think it is fair to claim that the control systems for Gnist described in this thesis have proved to work according to plan.

The main challenge for me has been lack of testing time after we got the car on the road. I anticipated that the time for doing proper tuning, testing and improvements would become narrow due to unforeseen problems. But it turned out to be much worse than expected. The project as a whole encountered so many problems,

that I often questioned if we would be able to get the car operational at all. The first big challenge for the team was massively delayed and defect gear box parts. They were supposed to arrive the 2nd of May, but we did not get working parts before the 24th of June. The second major challenge was caused by a monocoque failure where the connection points between the monocoque and the front wheel assembly broke during braking because wrong inserts and core material had been used. Just this took five days of intensive work day and night to fix. After that we experienced a lot of trouble with various electronic systems, which had to be fixed on a daily basis. Further, after we came back from the Silverstone competition the car was unstable due to compliance in the monocoque. It took five intensive days to work out where the problem was and to adequately fix it. The end result was that we got in total 10 test days with the car, which is arguably quite a restricted time window for making a prototype 4WID-EV competition ready.

We competed with Gnist in three Formula Student competitions this year; UK, Austria and Germany. At the UK competition we placed 5th in skidpad, 6th in acceleration, and 7th in autocross. During the endurance event a CAN transceiver on PCB responsible for sending power measurements to the VCU failed, which resulted in Gnist exceeding the power limit and therefore being disqualified from this event. Considering that the FSE teams that did better than us all had at least three years of 4WID-EV experience I feel confident in saying that my control systems functioned properly at almost the top level. At the Austria and Germany competitions we experienced a host of problems with two of our inverters. One symptom was that they delivered less power to the rear motors than requested, which resulted in poor performance, and the results from these competitions are therefore not included in this discussion as they do not reflect the actual performance of the control systems. Given the narrow time frame Gnist was fully operational, the successful operation of the control systems and the acclaim from those driving the car, I find it justified to assert that I have met the objectives for this thesis. I am of course unhappy about not getting the opportunity to conduct more thorough control system tuning and performance quantification due to problems outside my control. But I have at least provided a solid scaffold for the 2017 Revolve NTNU team to continue this work.

Despite that the development of a control system that depends on a multitude of components and subsystems prepared by others can cause some frustration from time to time, it has been extremely rewarding to experience how a dedicated team of students representing several different disciplines, and being completely dependent on each other, are capable of creating a functioning new racecar design from scratch. And being part of the Revolve NTNU 2016 team has taught me a lot about

skills other than the purely technical ones that a control engineer needs to possess in order to function well in a multidisciplinary engineering team.

14 Acknowledgements

I am most grateful to Professor Tor Arne Johansen for his willingness to be my Master of Science thesis advisor. And I would also like to thank all members of the Revolve NTNU 2016 team for their highly valuable support. The production of this thesis would not have been possible without the stimulating work environment that has been provided by this team and the resources that have been available through the Revolve NTNU organization. Last, but not least, I want to express my gratitude to the Revolve board members for giving me the opportunity to play an important role in developing Scandinavia's first four wheel-independent-drive electric race car.

References

[1] R. Rajamani, *Vehicle Dynamics and Control,* Mechanical Engineering Series,Springer Verlag, 2006.

[2] Reza N. Jazar, *Vehicle Dynamics: Theory and Application,* Springer, 2nd Edition.

[3] William F. Milliken, Douglas L. Milliken, *Race Car Vehicle Dynamics,* Society of Automotive Engineers, 1995.

[4] Fu, Chunyun, *Direct Yaw Moment Control for Electric Vehicles with Independent Motors, 2014.*

[5] Kaoru Sawase, Yuichi Ushiroda, *Improvement of Vehicle Dynamics by Right-and-Left Torque Vectoring System in Various Drivetrains*

[6] A. T. van Zanten, *Bosch ESP systems: 5 years of experience,* in Proc. Automot. Dyn. Stabil. Conf., Troy, MI, 2000, paper no. 2000-01-1633.

[7] Hans B. Pacejka, E., Bakker *The Magic Formula tyre model,* Vehicle System Dynamics, 21(Suppl. 1), 1-18.

[8] Leonardo De Novellis, Aldo Sorniotti et.al, *Comparison of Feedback Control Techniques for Torque-Vectoring Control of Fully Electric Vehicles,* IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. 63, No. 8, Oct. 2014

[9] R. Tchamna, I. Youn, *Yaw rate and Side-slip Control considering Vehicle Longitudinal Dynamics,* International Journal of Automotive Technology, Vol. 14, No. 1, pp. 5360 2013.

[10] Cheng Lin, Zhifeng Xu, Ru Zhang, *A Yaw Stability Control Algorithm for Four-Wheel Independently Actuated Electric Ground Vehicles considering Control Boundaries,* Mathematical Problems in Engineering Volume 2015

[11] Zhibin Shuai, et.al, *Lateral motion control for four-wheel-independent- drive electric vehicles using optimal torque allocation and dynamic message priority scheduling,* Control Engineering Practice, March 2014.

[12] Kanchwala, H. and Bordons, C., *Improving Handling Performance of an Electric Vehicle Using Model Predictive Control,* SAE Technical Paper 2015-01-0082, 2015.

[13] Mohd H. M. Ariff, et.al *Direct Yaw Moment Control of Independent-Wheel-Drive Electric Vehicle(IWD-EV) Via Composite Nonlinear Feedback Controller* First International Conference on Systems Informatics, Modelling and Simulation 2014.

[14] T. Bunte et. al., *Inverse Model Based Torque Vectoring Control For a Rear Wheel Driven Battery Electric Vehicle* The International Federation of Automatic Control, 2014.

[15] Leonardo D. Novellis et.al., *Direct yaw moment control actuated through electric drivetrains and friction brakes: Theoretical design and experimental assessment* Elsevier 2015.

[16] Leonardo De Novellis et.al., *Wheel torque distribution criteria for electric vehicles with torque-vectoring differentials* EEE Trans. Veh. Technol. 2013;63(4): 1593–602.

[17] Hans B. Pacejka *Tyre and Vehicle Dynamics* 2nd. Edition

[18] Masimo Canale, Lorenzo Fagiano, et.al, *Vehicle Yaw Control via Second-Order Sliding-Mode Technique* IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 55, NO. 11, NOVEMBER 2008

[19] B. Chen, C. Kuo, *Electronic stability control for electric vehicle with four in-wheel motors,* International Journal of Automotive Technology, Vol. 15, No. 4, pp. 573580 (2014)

[20] Y. Shibahata, K. Shimada, T. Tomari, *Improvement of vehicle maneuverability by direct yaw moment control,* Vehicle System Dynamics, vol. 22, no. 5-6, pp. 465-481, 1993.

[21] Ola Harkegård, S. Torkel Glad, *Resolvingactuator redundancy—optimal control vs. control allocation* Automatica 41 (2005) 137-144

[22] Tor A. Johansen, Thor I. Fossen, *Control Allocation - A survery*

[23] Jin-Oh Hahn, Rajesh Rajamani, *GPS-Based Real-Time Identification of Tire–Road Friction Coefficient,* IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY, VOL. 10, NO. 3, MAY 2002

[24] Moustapha Doumiati, Ali Charara et.al., *Vehicle Dynamics Estimation using Kalman Filtering*

[25] Y. Shibahata, K. Shimada, and T. Tomari, *Improvement of vehicle maneuverability by direct yaw moment control,* Veh. Syst. Dyn., vol. 22,no. 5/6, pp. 465–481, Jan. 1993.

[26] Pennycott, A., De Novellis, L., Sorniotti, A., and Gruber, P., *The Application of Control and Wheel Torque Allocation Techniques to Driving Modes for Fully Electric Vehicles* SAE Int. J. Passeng. Cars - Mech. Syst. 7(2):2014, doi:10.4271/2014- 01-0085.

[27] Raymond Brach, Matthew Brach, *The Tire-Force Ellipse (Friction Ellipse) and Tire Characteristics* SAE International 10.4271/2011-01-0094

[28] Sarah Spurgeon, *Sliding mode control : a tutorial* 2014 European Control Conference (ECC) June 24-27, 2014. Strasbourg, France

[29] Jan Erik Stellet, Martin Giessler, Frank Gauterin, *Model-Based Traction Control For Electric Vehicles*

[30] http://www.vehicledynamicsinternational.com/downloads/VDI_Lotus_Vector.pdf, *Torque vectoring*