# Design, modeling and implementation of the power train of an electric racing car

Control of a permanent magnet machine and implementation of a torque vectoring process in a FSAE car

Sylvain Blaszykowski

Master Thesis in Vehicle Engineering

Department of Aeronautical and Vehicle Engineering
KTH Royal Institute of Technology

TRITA-AVE 2013:23
ISSN 1651-7660

| ***Postal address*** | ***Visiting Address*** | ***Telephone*** | ***Telefax*** | ***Internet*** |
|---|---|---|---|---|
| KTH | Teknikringen 8 | +46 8 790 6000 | +46 8 790 6500 | www.kth.se |
| Vehicle Dynamics | Stockholm | | | |
| SE-100 44 Stockholm, Sweden | | | | |

## Abstract

This work emphasize the design, implementation and optimization of an electric power train for a Formula Student racing car.
As a first part, theory and control of a PMSM machine, in an automotive context is investigated: a CAN bus communication system has been implemented and a field weakening strategy. Precise modeling of the car has then been performed using CarMaker, developed by IPG and making it possible to perform accurate tests and forecast regarding the performances of the vehicle. This model was then used to develop and test different launch control strategies together with a torque vectoring strategy and study the influence of different parameters on the vehicle performances.

## Preface & Acknowledgments

This study is the first step of a real improvement for electrical machine control at KTH Racing. Even if the R9e prototype were not finished in time in 2012, most of the solutions and tools developed during this thesis have been tested and are working.

All difficulties encountered were surpassed and this work has been a fantastic mean of learning for me. For the first time, I have been given the chance to investigate a vehicle in its whole. I voluntarily did not focus this work on the hardware side of the implementation of an electric power train (manufacturing of parts, electronic design and coding) which has been done not only by me but by the whole KTH Racing team.

Now that I am working in a totally different field, I still use everyday learnings that I got thanks to the time I have spent on this wonderful project. I have not only developed my hard skills such as Permanent Magnet Synchronous Machine control or racing car simulation, but also and especially some very useful soft skills: work in groups or under pressure to cite only two of them and this makes me be a better engineer and manager everyday.

I would like to thank some people who made it possible for me to work on this amazing subject that is Formula SAE, and thus spending time on more than work, but also a passion:

- Daniel Wanner, my supervisor without whom none of that could have been possible, and Lars Drugge, coordinator of the department for his help and knowledge
- The 2011/2012 KTHRacing's team, with whom I have spent so much time, designing, building, and unfortunately not testing the R9e
- The Vehicle Engineering department of KTH, where I have discovered so many fascinating things and which gave me the opportunity to work on a subject I loved

# Contents

# Chapter 1

# Introduction

Automotive industry has changed drastically for the last few years due to the future lack of oil. Electric power trains are now more commonly used, for hybrid technology, and for full-electric cars. This currently represents a huge challenge for the industry. Better batteries have to be developed to increase autonomy and performance, weight savings have to be done to compensate for heavier power train. Racing has always been one step ahead in the car industry and is a way of testing and developing the future technology, and it is now very natural that electric racing cars starts to develop. One of them, The Formula SAE (FSAE) competition is specialy dedicated to students with an appropriate format of competition.

The FSAE competition is a global engineering design competition where students have to build a complete racing car prototype. Judging does not only focus on dynamic performances, but also on the quality of the design and on two other static events focusing on the business realism of the prototype and on its potential cost. This makes Formula Student an unique kind of project for students where all skills have to be developed to their highest level.

This work emphasize on the design and implementation of an electric power train for an electric racing car satisfying the FSAE rules [1] and especially the rules of the German competition, in the electric category [2]. Design specifications and power limitations led to a choice of an electric prototype whose rear axle is powered by two independent permanent magnet synchronous machines (PMSM). This increased the possibility to control each powered wheel independently compared to a combustion car and offers some remarkable advantages but also some challenging aspects such as control and safety.

The three main aspects reported in this study is the control of the two

PMSM each using an Infineon inverter, the modeling of a racing car using the software CarMaker and the calculation of the torque distribution ensuring high performances for cornering, acceleration, and efficiency.

## 1.1 Formula SAE competition

Formula SAE originally appeared in 1978 in the United States and is directed by SAE International (Society of Automotive Engineers). Since then, the competition has become global and nowadays, a dozen of events are organized worldwide each year.

This specific design optimization focuses on the FSG competition taking place in Hockenheim each year during the first week of August.
Performances of the different prototypes are judged during seven events[1]:

- Design event (50 points): innovation, engineering design, technical meaningful solutions and their reasoning, safety of the car are judged by a group of experienced actors of the industry

- Cost event (100 points): A full cost description of the prototype has to be performed in order to know the real price of the car in case of a full size production. Team using too specific methods (requiring too long hand work for instance) or too expensive materials might then have a very good design, but would lose points in the cost event.

- Business plan event (75 points): The team has to imagine a business plan in order to sell a certain number of cars in a year.

- Acceleration event (75 points): Each car has to perform a 75m drag race

- Skidpad event (75 points) : Each car has to perform a cornering test on a wet track

- Autocross event (100 points): Each car has to perform a race with a length of around 1km. Two attempts are allowed

- Endurance event (325 + 100 points for efficiency): Each team has to perform a long race (22km) including a pit stop and a driver change. Not only speed performances are judged, but also energy efficiency.

## 1.2 KTH Racing

KTH is a student association based in Stockholm. Since 2004, its aim is to build prototypes ready to race in the Forumla SAE competitions. It gathers student from different countries mainly students at KTH. Prize list of the team is interesting with a 3rd and 4th place won in European competitions, together with a "best use of electronics" price won in 2008.

After 7 combustion prototype (R1 to R7), the team decided to address the newly opened electric competition with its first electric concept, the R8e, sharing its rolling chassis with the R7. After this first attempt, the team decided to go further in the electric concept and to create its first fully electric prototype, the R9e (presented on figure 1.1 with the 2011/2012 team).

**Figure 1.1.** KTH Racing R9e - 2012 Hockenheim competition

## 1.3 Racing car prototype - R9e

This study is focusing on the R9e (figure 1.2) which is the ninth prototype developed at KTH Racing by students. Choice has been made to use a tubular frame design with rear wheel drive. Drivetrain is composed of two permanent magnet synchronous machines (PMSM) able to produce 52kW each and a two stages fixed gear box. Batteries are mounted behind and around the driver seat and suspension is designed with double A-arms and pushrods.

Table 1.1 gives a summary of technical characteristics and performances of the R9e as designed.

**Table 1.1.** R9e - Design specifications and performances

| Name | KTHRacing R9e |
|---|---|
| Power | 2 x 52kW |
| Inverters | Infineon HybridKit 2 |
| Torque vectoring | self designed independent wheel control |
| Battery nominal voltage | 370V |
| Battery capacity | 14.4 A.h |
| Weight | 220kg |
| Suspensions | double carbon A arms, pushrods |
| Brakes | ISR disk / calipers |
| Steering wheel | self designed with LCD screen |
| Chassis | tubular steel frame |
| Communication system | CAN bus, 500kbits |
| Data logging | 3D system |
| Bodywork | Carbon fiber |
| Sensors | Accelerations, yaw rates, GPS |
| | wheel speed (4), Dampers position (4) |
| | tires (12), brakes (4), cooling system (4) temperatures |
| | throttle/brake/steering wheel position |
| Maximum speed | 160km/h |
| 0-100km/h | ca 3.1s (simulation) |
| Lateral acceleration | ca 1.55g (simulation) |

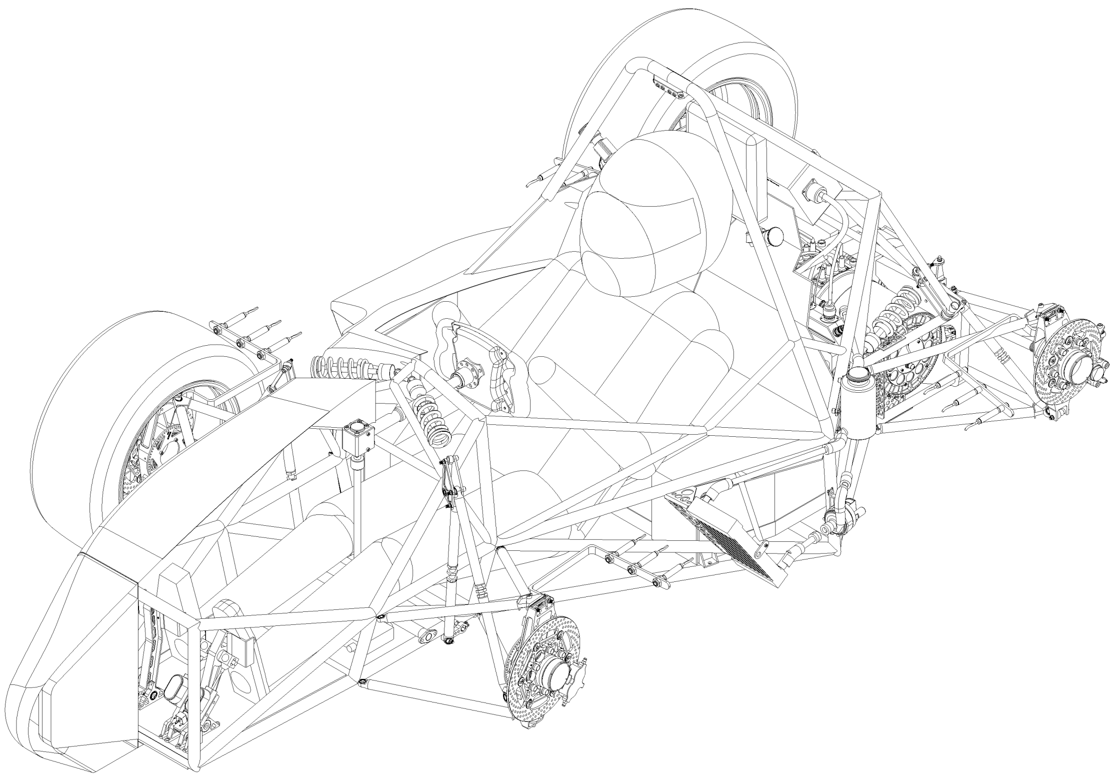


**Figure 1.2.** KTH Racing R9e - Isometric view

# Chapter 2

# Permanent Magnet Synchronous Machine & Inverters

The two electric machines used in the R9e are two Siemens 1FE1-082-6WP (see figure 2.1). These high performance water cooled machines can produce 52kW continuously and offer a very good power density. Their main characteristic are given in table 1.1. Their control is assured by two Infineon Hybrid Kit (see figure 2.2) which includes a logic board, a driver board, and a water cooling system capable to control systems up to 85kW each. Main characteristics of the inverters can be found in table 1.2.

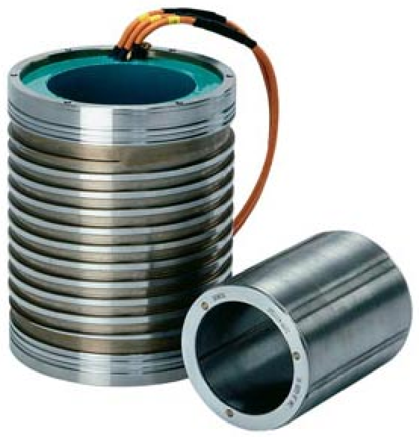


**Figure 2.1.** Siemens 1FE1-082 : active parts as received used in the R9e

**Table 2.1.** Siemens 1FE1-082-6WP - Technical data and characteristics

| Designation | Siemens 1FE1-082-6WP |
|---|---|
| Number of poles | 6 |
| Rated speed | 5000rpm |
| Maximum speed | 8500rpm |
| Rated Current | 112A |
| Maximum current | 130A |
| Maximum power | 52kW |
| Efficiency at $n_{rated}$ | 94% |
| Voltage constant | 68V/1000rpm |

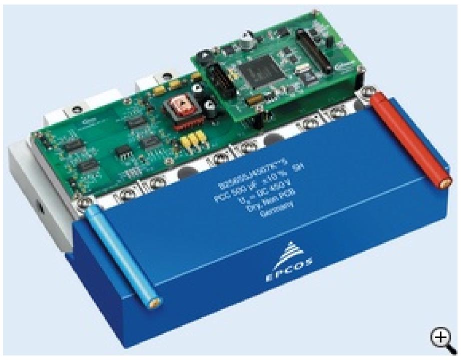


**Figure 2.2.** Infineon HybridKit 2

**Table 2.2.** Infineon HybridKit 2. Technical data and characteristics

| Designation | Infineon Hybrid Kit for HybridPACK 2 |
|---|---|
| PWM frequency | 20kHz |
| Voltage rated | 450V |
| Current rated | 200A |
| Rated power | 85kW |
| DC capacitor | 500uF |
| Cooling system | water cooled |
| Communication interface | serial (debug) / CAN (control) |
| Position sensor interface | resolver / encoder |

## 2.1 Mathematical model of a PMSM

It is important to build a realistic model of the PMSM in order to understand the best way to control it, and the principle of the field weakening and its advantages. In the next paragraph, $x$ stands for $a, b$, or $c$, designating the three phases of the motor. Usually, these values are physically accessible with a voltmeter for instance, but it will be shown that it s not a really useful reference.

In an electric motor with permanent magnet, equation for the voltage in the stator are given by [5] :

$$u_x = R.i_x + \frac{d\phi_x}{dt} \tag{2.1}$$

where $R$ is the resistance of one winding of the stator, and $\phi_x$ is the phase flux linkage.

Moreover, because of the structure of the stator windings (as shown on figure 2.3), it is evident that :

$$i_a + i_b + i_c = 0 \tag{2.2}$$

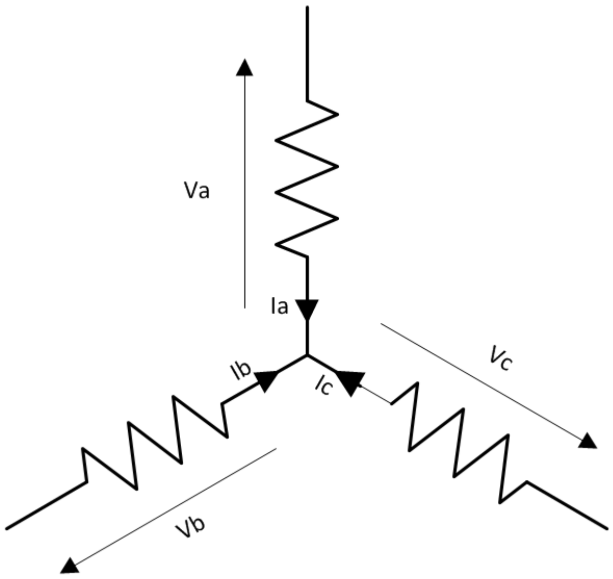


**Figure 2.3.** PMSM : Structure of the internal windings of the stator

The natural frame of reference $(a, b, c)$ is very easily physically accessible, but can be simplified by using the Clarke transform. The purpose is to use only two orthogonal directions $\alpha$ and $\beta$ instead of three. For more convenience, $\alpha$ is chosen so that $\alpha = a$. The linear relation existing between the different vectors makes it possible to write the relation as :

$$\begin{bmatrix} x_\alpha \\ x_\beta \end{bmatrix} = \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix} \tag{2.3}$$

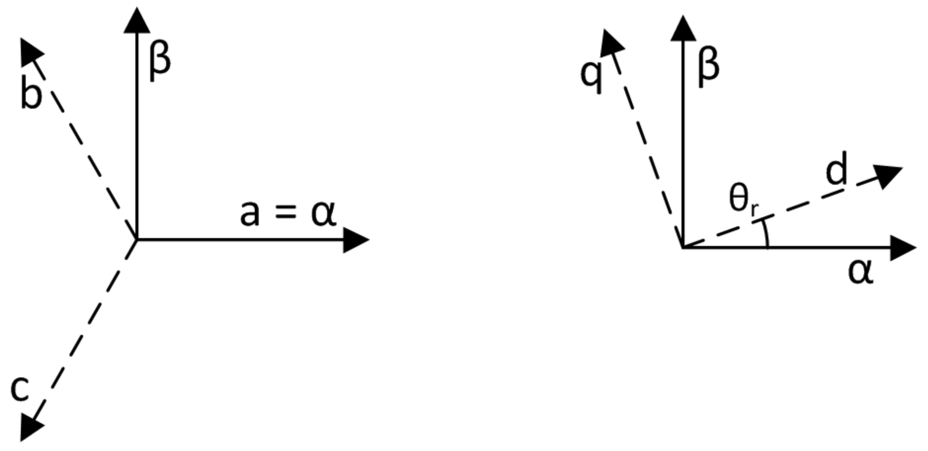


**Figure 2.4.** PMSM : Clark Transform and Park transform

The $(\alpha, \beta)$ referential is fixed to the stator. It is actually more interesting to introduce a referential where the rotor is fixed, making the calculations much easier as shown later. The transform is easily done by using a rotation matrix :

$$\begin{bmatrix} x_q \\ x_d \end{bmatrix} = \begin{bmatrix} -\sin\theta_r & \cos\theta_r \\ \cos\theta_r & \sin\theta_r \end{bmatrix} \begin{bmatrix} x_\alpha \\ x_\beta \end{bmatrix} \tag{2.4}$$

where $\theta_r$ is chosen so that the $d$ axis is aligned with the permanent magnet vector $\phi_m$ of the rotor. Using this notation, one has to admit that the expression of the torque generated for a surface mounted PMSM is [6] :

$$T = \frac{3}{2} p \phi_m i_q \tag{2.5}$$

where $p$ is the number of pole (six on the Siemens 1FE1-082-6WP) .

Since the axis has been chosen so that the $d$ axis is aligned with $\phi_m$, the flux can be expressed in the $(q, d)$ axis as :

$$\phi_d = \phi_m + L_d i_d \qquad \phi_q = L_q i_q \tag{2.6}$$

When applying the two referential transforms to the voltage equation, it can be shown that [5]:

$$U_d = R.i_d + \frac{d\phi_d}{dt} - \omega_r \phi_q \tag{2.7}$$

$$U_q = R.i_q + \frac{d\phi_q}{dt} + \omega_r \phi_d \tag{2.8}$$

where $\omega_r$ is the speed of the rotor.

The goal of the electrical machine is to create a torque that can be transmitted to the wheels. As shown on equation 2.5, this torque is created by the current along the $q$-axis. This axis does not correspond to any fixed axis but to rotating axis (fixed from the rotor point of view). Induction of a current along this axis is the aim of the inverter.

## 2.2 Control of the PMSM

Control of a PMSM is based on the control of the current vector (along $d$ and $q$ axis). In the further study, it is assumed that it is done using an inverter developed by Infineon (HybridKit Pack 2) especially for the automotive industry. To control this inverter, a logic board is needed, and an AMR EVK1100 board was used. The AMR EVK 1100 board is a development board incorporating a micro controller able to calculate in real time the different values to send. An add-on had to be developed in order to make it compatible with the CAN-Bus.

The internal structure of the inverter (shown in figure 2.5) makes it possible to have a current control of the PMSM (and not a voltage control) so that the orders sent by the torque vectoring module are the current commands $i_d^*$ and $i_q^*$ for each inverter. As a response, the inverters send the actual currents $i_d$ and $i_q$ but also some control variables like the DC voltage, the temperature of the inverters, or the speed or the motors.

The inverter is based on a closed loop current control and uses the different transforms (Park and Clarke) defined previously. The two input signals are the two currents $i_q^*$ and $i_d^*$ calculated by the torque vectoring module. From these two signals is subtracted the actual $i_q$ and $i_d$ measured by the current sensors in order to create the required current command. Using Clarke's and Park's transform, it is then possible to calculate the required $i_a$, $i_b$, and $i_c$. A Pulse-Width Modulation (PWM) module is then used to convert the DC voltage of the batteries into an adjustable source for each stator voltage $v_a$, $v_b$, and $v_c$ (see fig. 2.5).

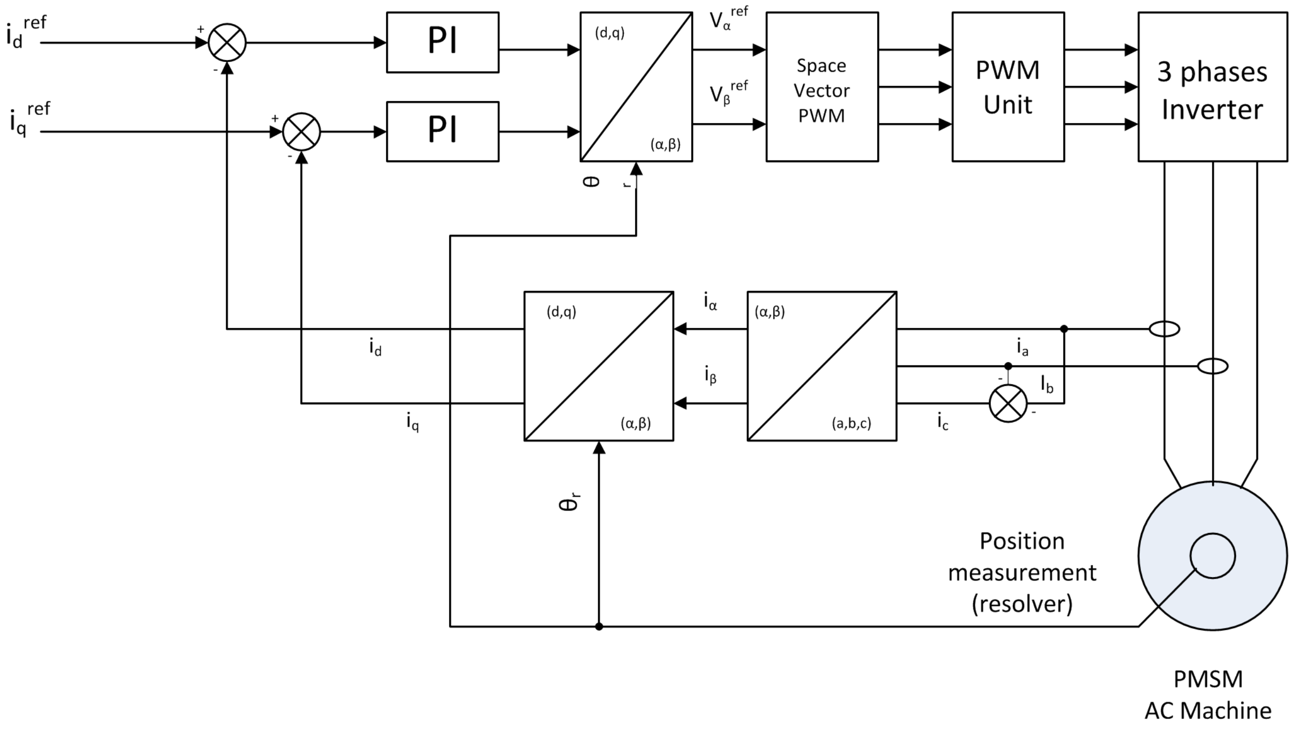


**Figure 2.5.** PMSM : Internal control of the inverter

Although it can look simple, all these calculations require precise setup. A closed loop control such as the one use in the inverter needs to be trimmed, and results are highly dependent on the reliability of the position control.

## 2.3 CAN bus communication

The CAN bus has been introduced by the automotive industry in the 80's in order to dramatically reduce the growing amount of wires used in new cars. It is now widely used and easy to implement as a lot of micro-controllers offer it as a functionality.

The most common physical layer used is based on the transmission over four wires (GND, V+, CAN Hi, CAN Lo), the two first being the supply which can be insulated from the other supplies in the vehicle, and the two last being used to transport the signal as a mirrored digital signal to reduce the influence of electromagnetic disturbances.

Messages are transported on frames identified by an *id*. Since all nodes receive all messages, filters have to be set in order to allow each node to receive only the messages it might be interested in. Since all devices are talking on the same line, conflicts can sometimes happen and priority rules and error frames are used to prevent it.

In order to keep the control easy, only five CAN messages (see table 2.3) are used between the controller and each inverter. For more convenience, the CAN *id* for the inverter left all start with a 1, while the CAN id of the inverter right start with a 2.

**Table 2.3.** CAN messages between the control board and the inverter left

| CAN id | Name | Direction | Content |
|---|---|---|---|
| 110 | Status 1 Left | Inverter left → TV. module | Safety status (Cooling required, AC fault...) |
| 111 | Status 2 Left | Inverter left → TV. module | Monitoring (IGBT temperatures, DC voltagem rpm) |
| 112 | Status 3 Left | Inverter left → TV. module | Current monitoring (in $(a, b, c)$ base) |
| 113 | Status 4 Left | Inverter left → TV. module | Current monitoring (in $(q, d, 0)$ base) |
| 114 | Set Left | TV. module → Inverter left | Current value setting (in $(q, d, 0)$ base) |

## 2.4 Control strategy and field weakening

The control strategy of a PMSM is the table used to calculate the two inputs given to the inverter : $i_d^*$ and $i_q^*$ according to the rotation speed and the throttle position for instance. This strategy is calculated by the logic board and sends the CAN messages to the inverter controlling the electrical machines.

An electrical machine is usually limited by two different factors: its maximum voltage, and the maximum current it can handle $I_{max}$. One must be careful to keep $I = \sqrt{i_d^2 + i_q^2} < I_{max}$. Maximum voltage is not considered as a potential issue here since the battery voltage is below all components limitation, and regeneration is not considered.

Since the torque is only influenced by $i_q$ (c.f. equation 2.5), it is useful to keep $i_d = 0$ in order to maximize the range of $i_q$ and thus the torque available. However, according to equation 2.8, in the stationary state, there is a maximum rated speed that can be reached if $i_d = 0$ due to the back electromagnetic voltage $\omega_r \phi_d$.
Once this speed has been reached, the available torque vanishes, unless a negative current on the $d$ axis is applied. However, one must be careful that the maximum torque achievable for a speed above the maximum rated speed will be lowered due to the total current limitation. This process is called field weakening.

Equation 2.8 makes it possible to calculate the rated speed for the specific motor used. If $i_d = 0$ the voltage constant (68V/1000rpm) and the DC voltage used (370V) give a speed where the torque available vanishes (proportional to $i_q$) of :

$$\omega_r^{rated} = \frac{V_{DC}}{\phi_d} = \frac{370}{68} \simeq 5400rpm \tag{2.9}$$

Nevertheless, this speed corresponds to the very maximum speed achievable without any load and would thus never be attained, due to the friction for instance. In order to be able to produce torque constantly, the rated speed considered now is the maximum speed achievable in steady state condition with maximum torque ($i_q = 110A$). The rated speed is then given by :

$$\omega_r^{rated} = \frac{V_{DC} - r \times i_q}{\phi_d} = \frac{370 - 0.4 \times 110}{68} \simeq 4800rpm \tag{2.10}$$

where $r$ stands for the resistance of the windings. Above this speed, the back electromagnetic voltage makes it impossible to reach the current limit (it is actually a voltage limit).

Several ways can be used to perform a field weakening (constant voltage, constant power, constant current constant power, optimal current control [5]...) but not all of them requires the same knowledge of the electrical machine.

Choice was made to use the Constant Current Constant Power (CCCP) [5] algorithm for its simplicity and good efficiency. It consists in decreasing the q-axis current and keeping it inversely proportional to the rotor speed :

$$i_q = i_q^0 \frac{\omega_r^{rated}}{\omega_r} \tag{2.11}$$

where $i_q^0$ is the current that would have been sent without field weakening. The field weakening current $i_d$ is simply calculated by :

$$i_d = -\sqrt{\left(i_q^0\right)^2 - i_q^2} \tag{2.12}$$

This simple algorithm has several advantages :

- It does not require any precise information about the electrical machine, and is very adaptive. It is for instance very easy to adjust it in case of battery voltage drop.

- It ensures a good control of the total current $\sqrt{i_q^2 + i_d^2}$ which is equal to $i_q^0$. This current is important since it is the one that has to be measured to calculate the electrical power used by the motor. Since this power is drastically controlled and limited during the competition, it does matter to monitor it with a good accuracy. Moreover, the DC voltage has a rated value of 370V but can vary from 420V to around 280V during an endurance race. Being able to adapt the maximum current allowed during the competition is then a real advantage. One can imagine that during the beginning of the endurance race, with battery fully charged, the maximum current should be lowered to ensure that the power limitation will not be exceeded, but at the end of the race, and when the battery voltage drop is significant, it can be useful to lower this current limit further.

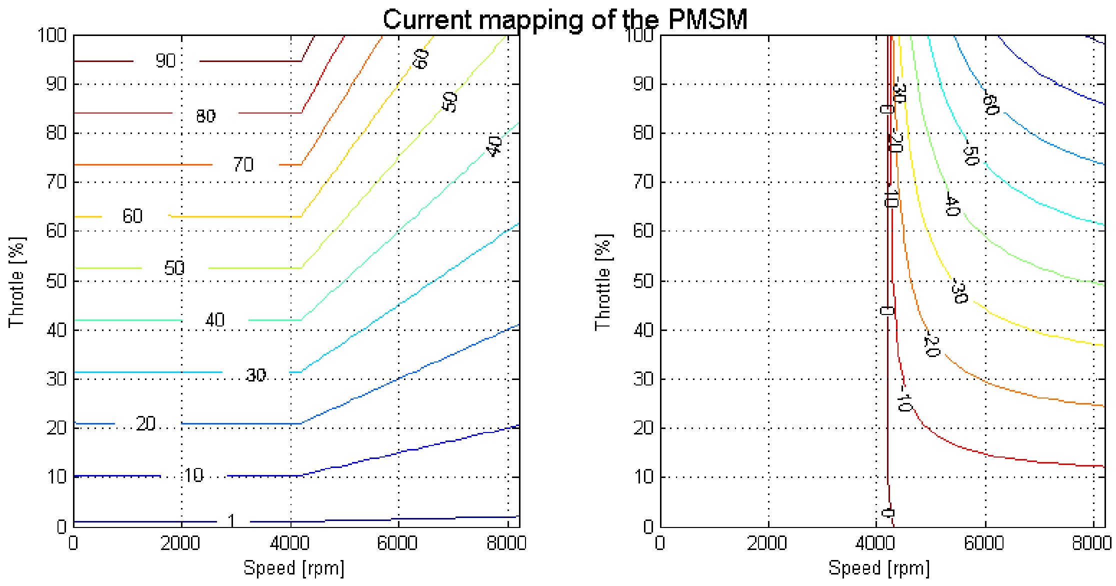


**Figure 2.6.** Matlab - Current mapping of the PMSM for a linear response of the torque according to the throttle (Left figure : $i_q$, Right figure : $i_d$)

Since the command of the inverter is made of $i_q^*$ and $i_d^*$, the figure 2.6 shows the mapping that has to be implemented on the AMR board to calculate the required current values. At any time, the AMR board gets the rotational speed of the electric machines (sent by the inverters on the CAN bus) and the throttle command calculated by its own torque vectoring module. Thanks to the mapping shown above, it can then calculate the appropriate current command to send to the inverters.

For instance, for a rotational speed of the machine of 2000rpm and a throttle command of 50%, one can read on left plot that the $i_q$ command should be 47A while the second plots shows that $i_d = 0A$. Had the speed been 6000rpm. then we would have $i_q = 35A$ and $i_d = -35A$ because of the field weakening.

Since the torque is proportional to $i_q$, the power is kept constant during the field weakening zone while it was increasing linearly until then as shown on figure 2.7 :

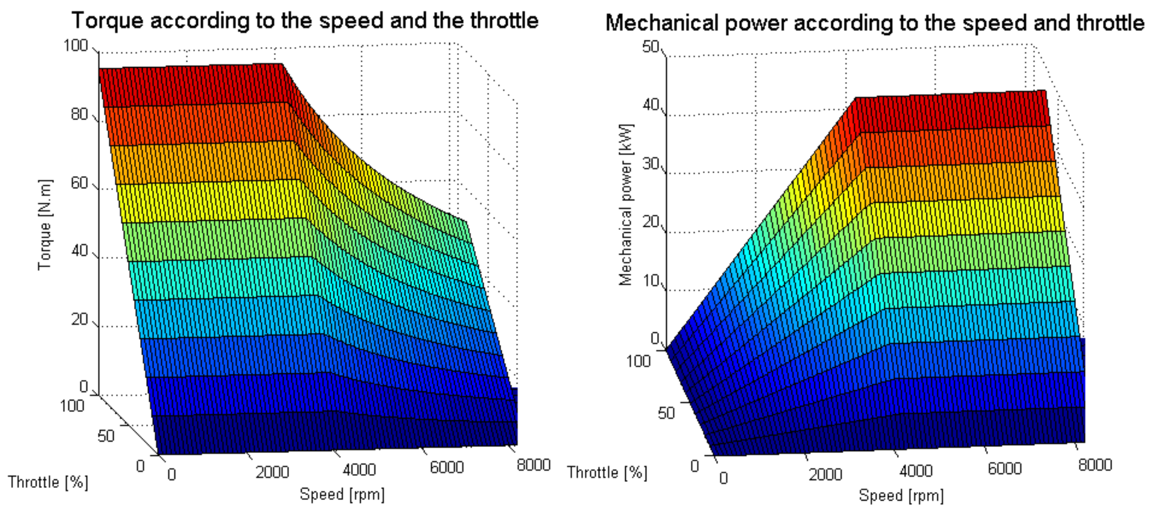


**Figure 2.7.** Matlab - Torque and power mapping of the PMSM according to the motor speed and throttle command

Because the throttle is completely electrically commanded, it is possible to implement some other relation between the throttle position and the torque command. A linear relation has been used for the simulations, but some non-linear relations might lead to a better control for the driver, but have to be investigated experimentally.

# Chapter 3

# Modeling and simulation of a racing car

Simulation is, and has always been a big concern for engineers. It can save time, money, and help to improve design before manufacturing. Thanks to the development of computer science, it is now possible to run accurate simulations on its own using professional software. This part of the study is focused on the implementation of the model of the R9e in CarMaker, a software developed by IPG and providing help to FSAE teams.

## 3.1 IPG CarMaker & Simulink

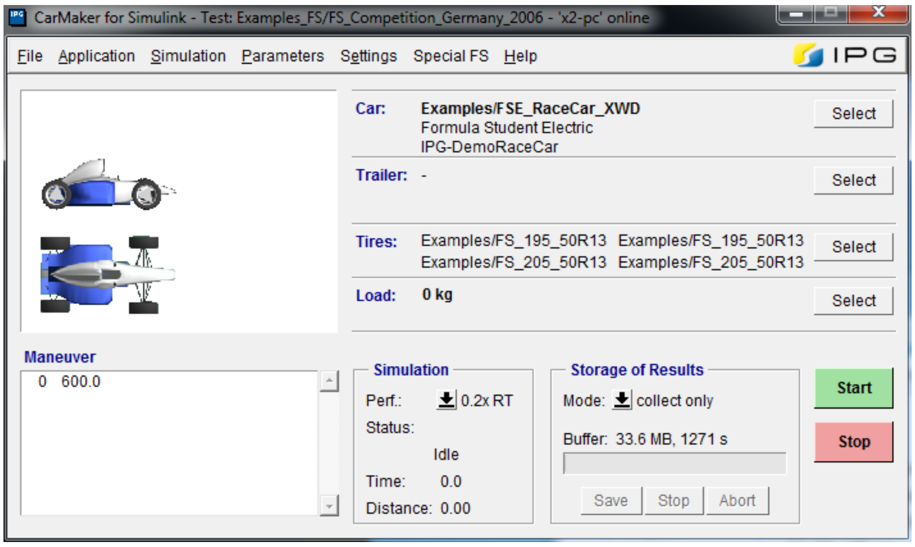
The complete car was simulated in order to be able to predict the effect of parameters changes in the torque vectoring. The simulation solutions offered by IPG is articulated around three main tools :

- CarMaker is the core of the simulation tool. It provides several models, including a basic formula student model. Existence of such a model makes it possible to save a lot of time by focusing on very specific adjustment of the car, such as the suspensions or the power train implementation.

- IPGDriver is the driver module of the simulation tool. It has the capability to drive as a normal driver, or as a racing driver. In the latter mode, a preliminary knowledge is required. The virtual driver can learn from the car by running different basic maneuvers such as acceleration, constant speed cornering, braking, etc.. This learning makes it possible for the racing driver to operate the car at its maximum physical limits. Moreover, learning of the tracks is also possible by running it in a loop. This racing driver is a very good asset because it makes it possible to analyze with a good precision the effect of the change of any parameter.

Nevertheless, the normal driver has also some advantages. Formula student cars are driven by students who do not have the knowledge of real racing drivers and often do not have enough time to perform enough tests. The normal driver makes it possible to simulate a driver who would not drive the car at its best performances, trying for instance to accelerate or brake too hard.

- IPGKinematics is a special module dedicated to suspension, steering and chassis systems. Precise modeling of the suspension system incorporating compliance effect for instance can be done.


CarMaker for Simulink - Test: Examples_FS/FS_Competition_Germany_2006 - 'x2-pc' online

Car: Examples/FSE_RaceCar_XWD
Formula Student Electric
IPG-DemoRaceCar

Trailer: -

Tires: Examples/FS_195_50R13 Examples/FS_195_50R13
Examples/FS_205_50R13 Examples/FS_205_50R13

Load: 0 kg

Maneuver
0 600.0

Simulation
Perf.: 0.2x RT
Status: Idle
Time: 0.0
Distance: 0.00

Storage of Results
Mode: collect only
Buffer: 33.6 MB, 1271 s

**Figure 3.1.** CarMaker : main interface window

CarMaker is a self-sufficient software that does not require any specific software environment to work, nevertheless, it can be coupled with Simulink to increase the control possibility.

Indeed, CarMaker makes it possible to read its control values: a few hundreds of control values are directly accessible to Matlab/Simulink. It makes it possible to store them and perform some complex data analysis. Furthermore, it is actually possible to modify these values through the Simulink interface. There is a risk with modifying these values since they can lead to non-physical situations: One can theoretically force the speed of the vehicle or its acceleration which could have disastrous consequences on simulations. During my whole study, only one original control variable has been modified through Simulink and correspond to the torque distributed to each of the rear wheels.

To maker it easier to understand the further simulink schematic, it can be interesting to notice the three blocks implemented by CarMaker to read / write / initialize a control variable shared with Simulink and CarMaker:

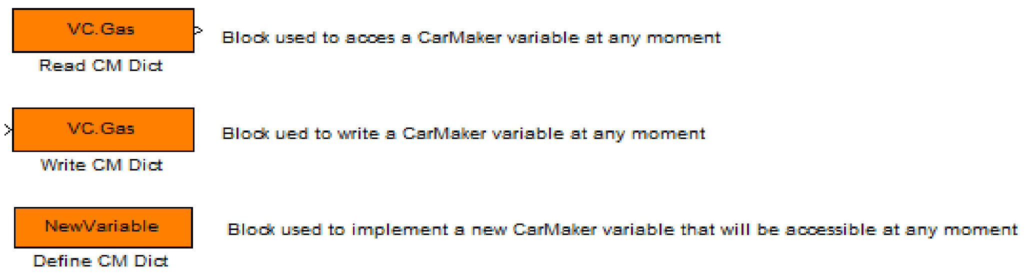

VC.Gas
Read CM Dict
Block used to acces a CarMaker variable at any moment
VC.Gas
Write CM Dict
Block ued to write a CarMaker variable at any moment
NewVariable
Define CM Dict
Block used to implement a new CarMaker variable that will be accessible at any moment

**Figure 3.2.** CarMaker : specific library implemeted in simulink

- The first block makes it possible to access any environment variable, such as the speed, the steering angle, the different tires load...
- The second block makes it possible to write any environment variable. It has mainly be used to implement the electric torque applied to the wheels. One must stay carefully with its use since it overwrites any calculated environment variable and thus can lead to simulations that are not physically valid.
- The third block makes it possible to create a new environment variable that can be used to store a variable until the next cycle for instance.

## 3.2 Power train modeling

The power train is modeled in a simple way. As shown on figure 3.3, it has been divided into two main sub blocks. The first gray block is dedicated to the torque vectoring. It is not a physical block but has to be exported and implemented on the logic board in the car. It shows the different inputs required by the torque vectoring such as the wheel speed, or the different accelerations. The second blue block represents the motor interface with the implementation of the field weakening. It takes into account the throttle command sent by the torque vectoring module and tries to give the matching $i_d$ and $i_q$ current for each motors.

The gain block (red triangle) on the left are used to simulate a sampling frequency lower than the natural frequency of simulink. Indeed, CarMaker is able to give instantaneously (the frequency is actually 10kHz) all parameters to simulink while sensor (for instance wheel speed sensors) require some time between two sampling. The gain blocks on the right represent some physical parameters of the car such as the link between $i_q$ and the torque generated, and the final gear ratio.

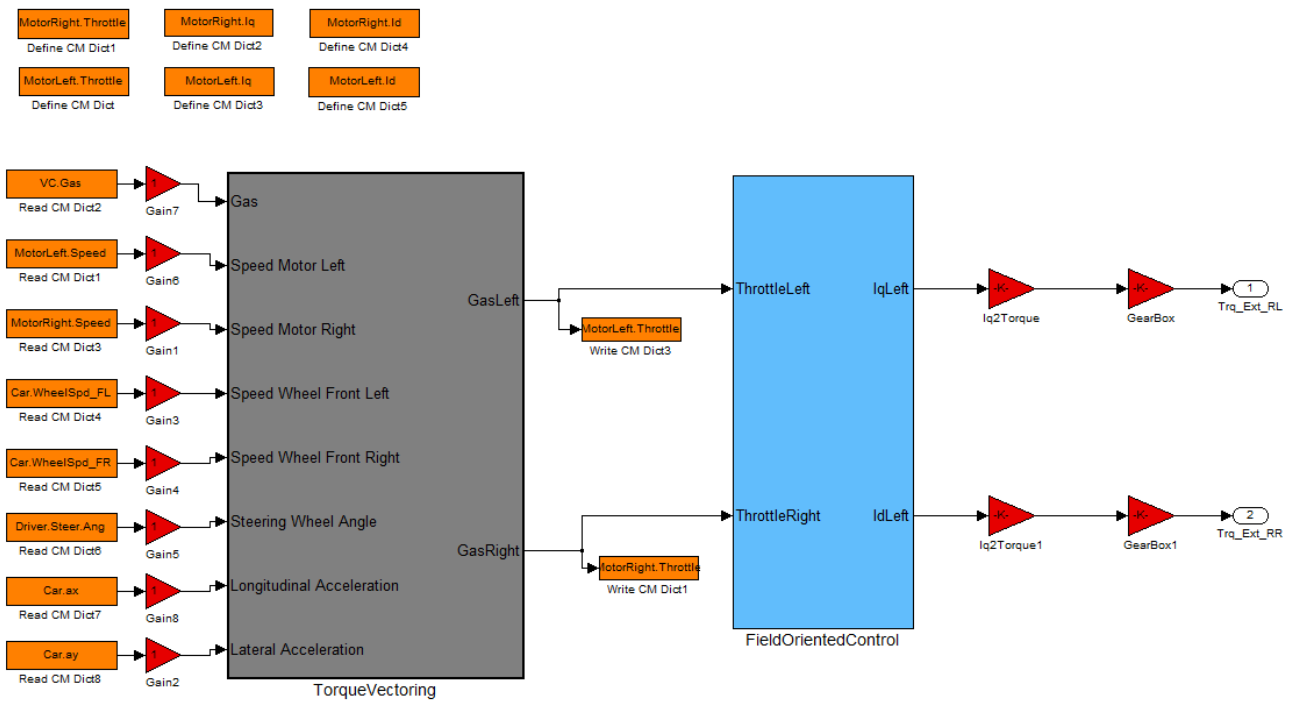


**Figure 3.3.** CarMaker : high level view of the power train model

**Torque vectoring module :** represented by the gray box, its role is to calculate the amount of torque sent to each wheel. Details about this block is given on figure 3.4.

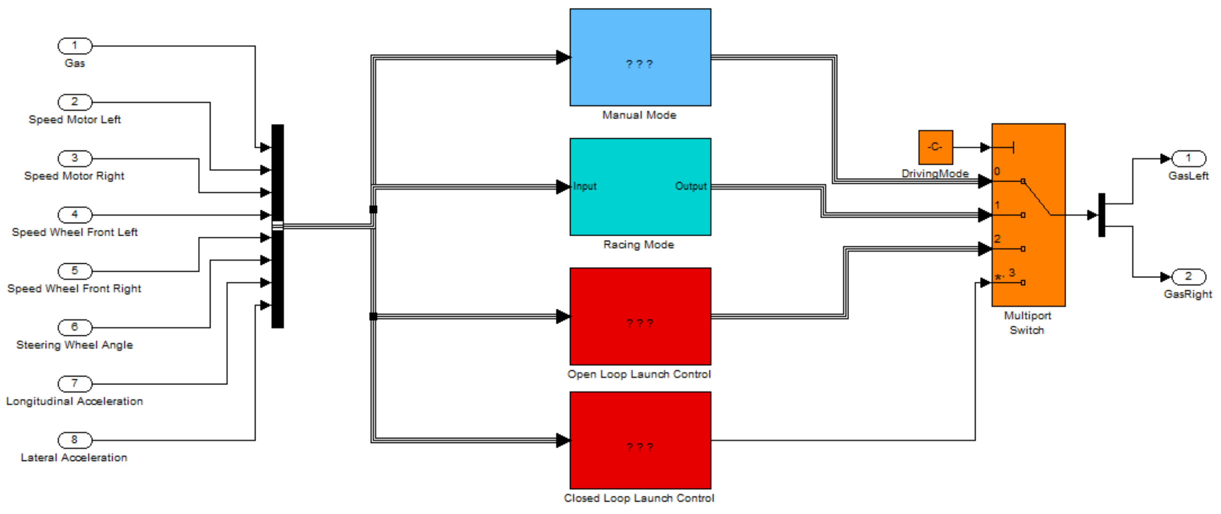


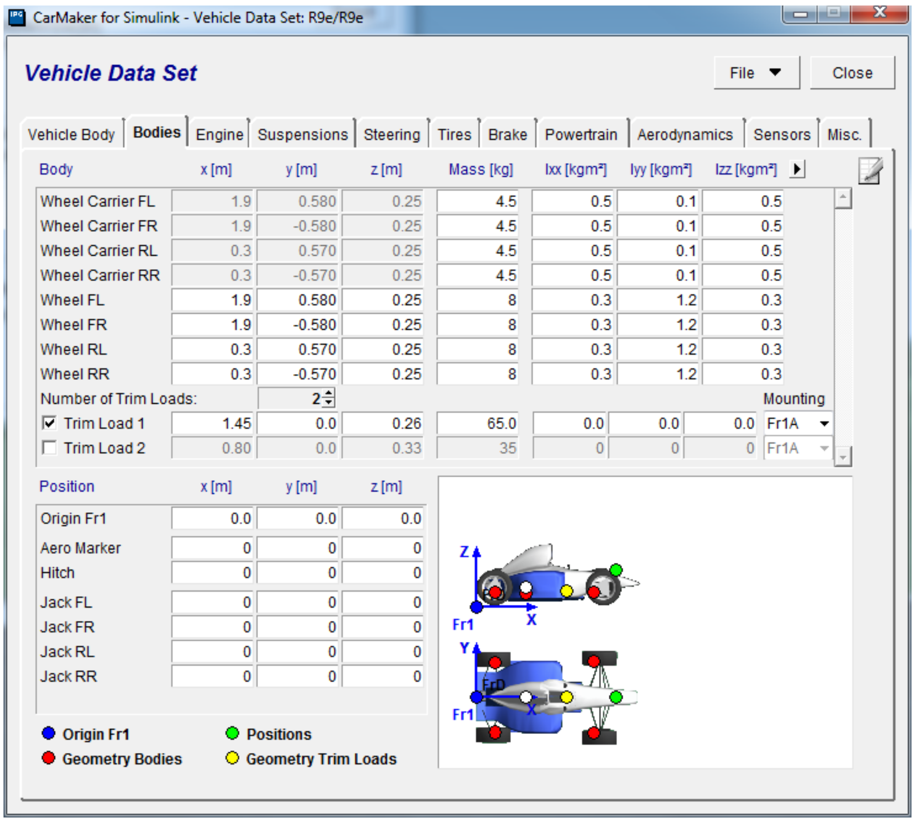
**Figure 3.4.** CarMaker : control level of the torque vectoring

The blue block represents a bypass mode where the control of the motors is only made by the driver. The blue-green box is the racing box where torque is limited according to the situation, and the two red box are two different launch control devices made to optimize the longitudinal acceleration. The orange switch has a physical sense since it makes it possible to select different profiles for the torque vectoring. These different profiles are selected using a rotary switch on the steering wheel.

**Electrical machine :** The two electrical machines used are two PMSM designed by Siemens and whose reference is 1FE1-082WP (see table 2.1 page 8). Explanations about their characteristics and control was given in part 1. The whole process of the vector control is not detailed in the simulink program used to drive the Carmaker module. Although it is possible to implement a complete control of a PMSM in simulink, it is very CPU time consuming and leads to a model with a refreshment rate of 0.05x (it takes 100 seconds to simulate a 5 seconds run). The main interest of CarMaker is to be able to perform quick simulation with rendering at a refreshment rate up to 10x (a simulation run of 100 seconds takes only 10 seconds to simulate). Furthermore, direct modelling of the behavior of the PMSM is not useful. This is the reason why a system box was created whose input are the four current ($i_d$ and

$i_q$ for each machine) and the speed of each machine. Input is given as a torque according to a 3D lookup table. This method gives the advantage to be one hundred times faster than the whole modeling of the machines and gives very good results.

**Main parameters adjustment :** CarMaker makes it possible to implement a formula student car very quickly thanks to the FS car model given. One can directly adjust parameters such as global weight, load distribution, the different inertia, drag coefficient and the suspensions settings. The trim load 1 (see figure 3.5) makes it very easy to study the influence of the driver mass for instance.



| Body | x [m] | y [m] | z [m] | Mass [kg] | Ixx [kgm²] | Iyy [kgm²] | Izz [kgm²] | |
|---|---|---|---|---|---|---|---|---|
| Wheel Carrier FL | 1.9 | 0.580 | 0.25 | 4.5 | 0.5 | 0.1 | 0.5 | |
| Wheel Carrier FR | 1.9 | -0.580 | 0.25 | 4.5 | 0.5 | 0.1 | 0.5 | |
| Wheel Carrier RL | 0.3 | 0.570 | 0.25 | 4.5 | 0.5 | 0.1 | 0.5 | |
| Wheel Carrier RR | 0.3 | -0.570 | 0.25 | 4.5 | 0.5 | 0.1 | 0.5 | |
| Wheel FL | 1.9 | 0.580 | 0.25 | 8 | 0.3 | 1.2 | 0.3 | |
| Wheel FR | 1.9 | -0.580 | 0.25 | 8 | 0.3 | 1.2 | 0.3 | |
| Wheel RL | 0.3 | 0.570 | 0.25 | 8 | 0.3 | 1.2 | 0.3 | |
| Wheel RR | 0.3 | -0.570 | 0.25 | 8 | 0.3 | 1.2 | 0.3 | |
| Number of Trim Loads: | | 2 | | | | | | Mounting |
| ☑ Trim Load 1 | 1.45 | 0.0 | 0.26 | 65.0 | 0.0 | 0.0 | 0.0 | Fr1A |
| ☐ Trim Load 2 | 0.80 | 0.0 | 0.33 | 35 | 0 | 0 | 0 | Fr1A |

| Position | x [m] | y [m] | z [m] |
|---|---|---|---|
| Origin Fr1 | 0.0 | 0.0 | 0.0 |
| Aero Marker | 0 | 0 | 0 |
| Hitch | 0 | 0 | 0 |
| Jack FL | 0 | 0 | 0 |
| Jack FR | 0 | 0 | 0 |
| Jack RL | 0 | 0 | 0 |
| Jack RR | 0 | 0 | 0 |

● Origin Fr1
● Positions
● Geometry Bodies
● Geometry Trim Loads

**Figure 3.5.** CarMaker : parameter adjustment of standard variables

## 3.3 Tire model

Using appropriate tire data is crucial in the model of a racing car. Without this, it becomes impossible to get proper results from simulation because the behavior of the tire has to be taken into account during the trimming process of the suspension for instance.

Data of the tires is not easy to create and requires expensive equipment. Miliken research associates [3] has provided for years specific data for racing tires. Raw data is given as a .csv files each containing large amount of measurement points for various vertical load and inclination angle. For each measurement set (fixed $F_Z$ and camber angle), a double slip angle sweep is performed while 21 different variables (load, moments, temperatures) are logged.

Most of the work consist in building a Matlab script able to identify the different measurement sets automatically and export the measurement points in a format accepted by CarMaker (code given in appendix A.2 page 55). CarMaker accepts two main input as tire model, either the Pacejka model, or the Tydex format [4]. This last format has been used since it does not require any modeling work and is a descriptive model. The *tireutil.exe* executable given with CarMaker converts then the Tydex file created by Matlab into an internal model used by CarMaker.

Figure 3.6 shows the result of the Matlab script after sorting all data sets and smoothing of the results.

## 3.4 Simulation

In order to be able to investigate several aspects of the car (acceleration, cornering speed, energy efficiency), several tracks and simulation procedures have been used :

### 3.4.1 Acceleration race

To be as close as possible to the real competition, the acceleration is tested on a 75m straight line. Different friction coefficients can be tested in order to study the efficiency of the launch control systems depending on the weather for instance.

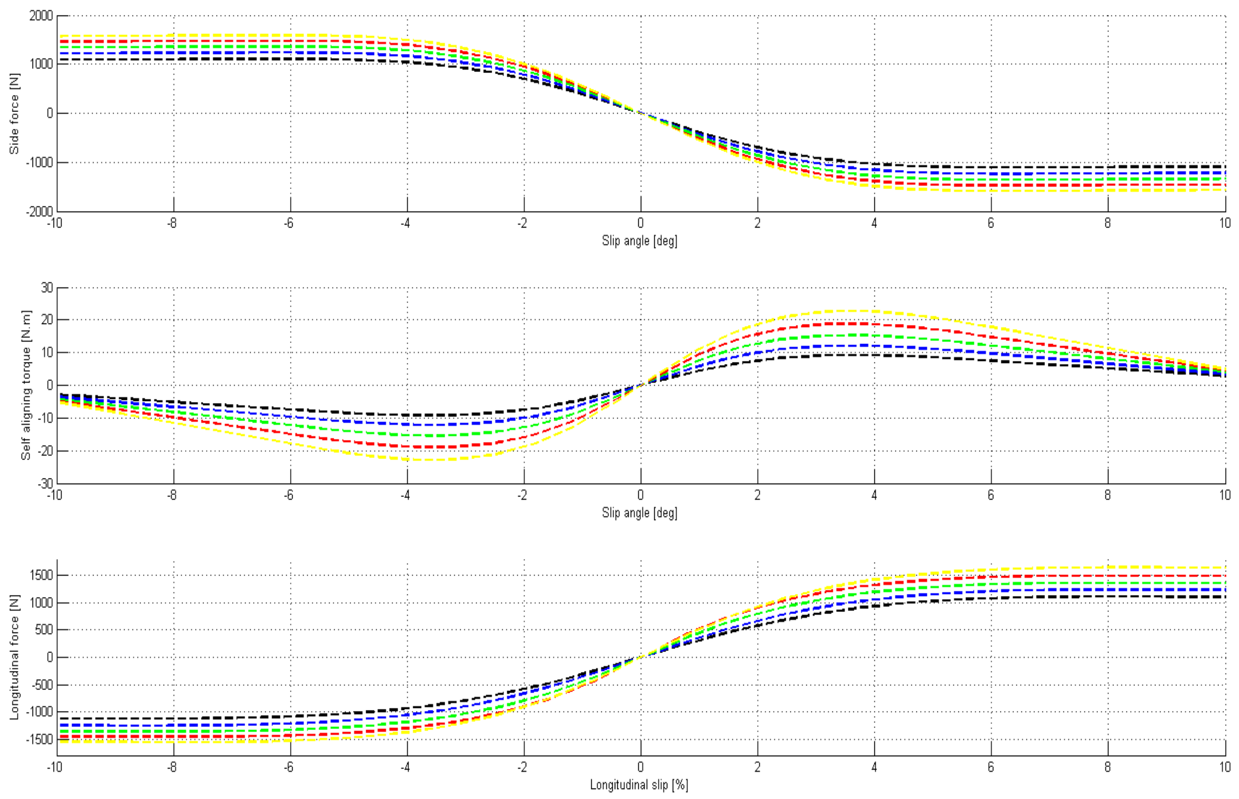


**Figure 3.6.** Tire model : characteristics of the Hoosier tires at 14PSI (vertical load : yellow : 1100N, red : 1000N, green : 900N, blue : 800N, black : 700N)

### 3.4.2 Endurance and sprint race

A real track (see figure 3.7) has been simulated in order to be able to put the car in real racing conditions. The length is rather small (a bit less than 1km) with a maximum speed achieved around 100km/h. These characteristics perfectly match with the kind of tracks actually used during the competition.

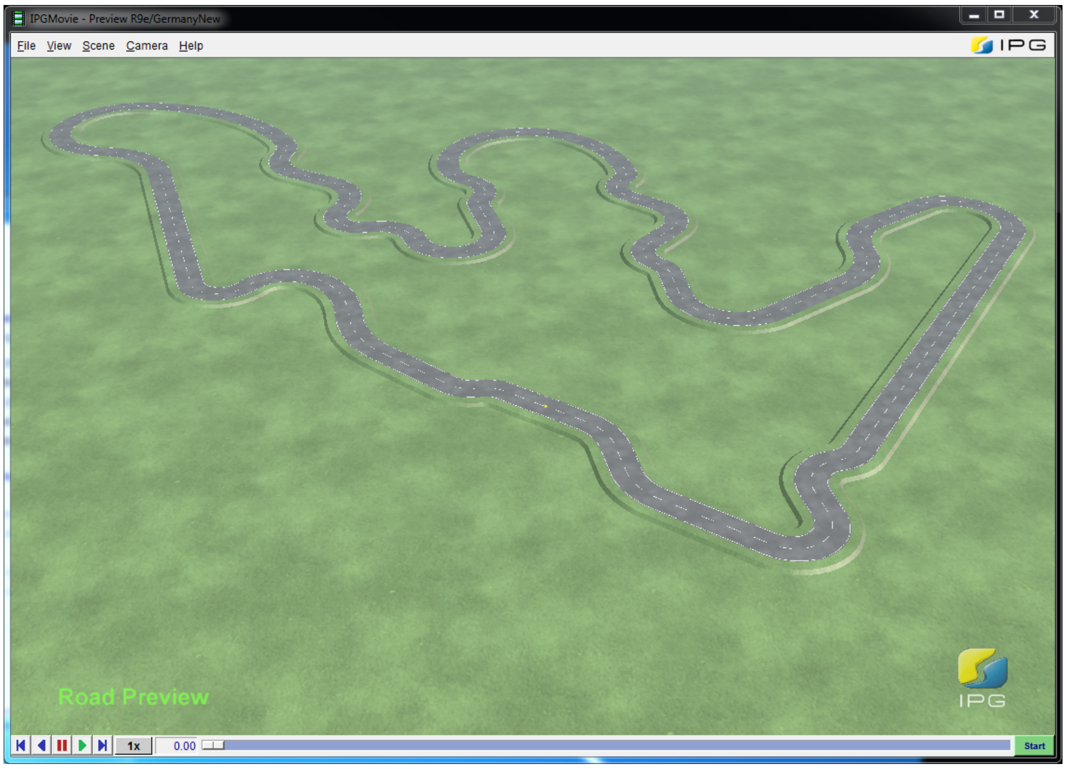

IPGMovie - Preview R9e/GermanyNew
File View Scene Camera Help
IPG
Road Preview
IPG
1x
0.00
Start

**Figure 3.7.** CarMaker: track used for sprint and endurance races

### 3.4.3 Tools developed

Some virtual tools had to be developed to be able to monitor and study some non trivial parameters :

- An energy-meter : As said in the introduction, energy efficiency is one of the major aspects of the competition. One should not forget that an electric car is very much limited in autonomy by its battery capacity. Being able to monitor the amount of energy used according to the weather condition, torque vectoring strategy, is then primordial
- A power-meter : Rules are extremely strict about the power that can be used to power the car. On an electric car, it is very easy to exceed such a limit by making a small control mistake. During the competition, this power consumption is constantly monitored by an official device, which had to be modeled.

# Chapter 4

# Optimization of the performance through simulation

The format of the competition gives a very important weight to acceleration and corning performances. These two different properties do not require the same kind of control of the torque and that is the reason why compromises have to be found. In order to build the torque command for the inverters, an EVK 1100 board is used, making it possible to export Simulink models for instance.

Because of lateness in the parallel project of building the real car, real test have not been conducted to validate the results of this study. This is the reason why a way to keep a track on improvements was needed. As an arbitrary reference, the exact same model was used, with same weight, tire, suspensions, but with a dummy torque vectoring, giving to both rear wheels the same torque command.

## 4.1 Acceleration optimization

### 4.1.1 Main principle and theory

Acceleration performances are judged on a 75m straight race. Suspension settings being given, the only way of improvement is the control of the torque, and precisely, the slip ratio of the powered wheels. A good racing driver can perform well in this exercise, but a computer assistance can give better results, ensuring the perfect slip ratio all way long.

In this quasi uni dimensional movement, the equation of the car can be simply written :

$$M.a_x = F_w - f_r \tag{4.1}$$

where $M$ stands for the total inertia mass of the vehicle (with equivalent mass of inertia such as rotating mass in the power train), $F_w$ is the force due to the torque on the powered wheel, and $f_r$ the overall resistance force (non powered wheel, air drag, friction). Maximize the acceleration will always give the best speed so it is important to keep the force created by the powered wheel at their maximum.

As seen in the previous part, the longitudinal force $F_w$ created by a tire is a function of its longitudinal slip ratio $s_r$. As shown on figure 3.6 (page 26), the curve of this function is convex and has a maximum for a slip ratio of 8%.

### 4.1.2 Launch control device implementation

Two different launch control strategies based on two different principles have been developed.

#### Model based launch control

The first is a model based control. It uses the CarMaker simulation to predict the behavior of the car. If the model is precise enough, it is possible to create some lookup table that will make the control of the car easy. To each state of the car (acceleration, speed...), the torque vectoring device can link an optimal torque distribution and apply it to the rear wheels.

The easiest open loop control made still requires to measure the longitudinal acceleration, which is easily accessible thanks to the 2D data logger. Assuming that each measurement point is a steady state point for the suspension, it is then possible to calculate the load distribution of the car. Simulation of acceleration of the car have been made using the IPG Driver and an acceleration command to model the relation between load distribution and longitudinal acceleration. It can be shown that a linear approximation is good enough and the load distribution on the rear wheel is then given by :

$$F_{Z,R} = d.a_x + F^0_{Z,R} \tag{4.2}$$

where $F^0_{Z,R}$ is the static load on the rear wheels that can be measured with the car laying still. $d$ can be calculated using the height of the center of gravity and the wheelbase, but here, CarMaker simulations has been used and $d$ is an experimental factor.

Thanks to the precise tire model used, the maximum torque that can be transmitted to the road through the wheel can be calculated and the command sent to the inverter. For the range considered (vertical load between 700N and 1200N for each rear wheel), a linear approximation can also be used to determine the maximum longitudinal force that can be transmitted, and thus the maximum torque for the motor (using the gear ratio and the wheel diameter).

Finally the model based control is given by this simple equation :

$$T_{motor} = f_r^{corr} \times \left( K a_x + T^0_{motor} \right) \tag{4.3}$$

where $K$ is an experimental factor describing the slope of the curve for a road friction coefficient of 1 and $T^0_{motor}$ the torque that can be transmitted when the car is standing still. The factor $f_r^{corr}$ is used to be able to trim the model according to the driving conditions since maximum transmitted forces for a tire are proportional to the road friction coefficient.

An inverse mapping of the motor (blue boxes on figure 4.1) is still needed to determine the throttle command corresponding to the desired torque and motor rotation speed. It is made my inverting the equation $T = f(throttle, rpm)$ to give the throttle needed to achieve a specified torque and rpm.

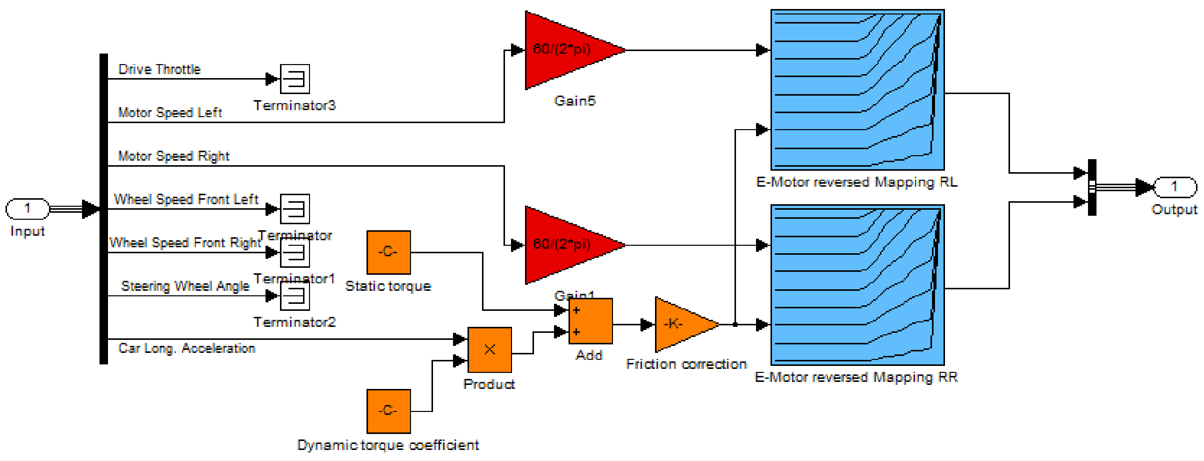


**Figure 4.1.** Simulink - Open loop model based launch control

## Closed-loop device

The second launch control device implemented uses a closed loop control on the longitudinal slip ratio. The number of inputs it needs is significantly higher since it requires to know the four wheels speed with a very good accuracy. Racing tires usually give their maximum longitudinal force for a slip ratio around 6 or 7%, but a simple deviation of 2% would decrease the force generated dramatically. Sampling speed of the sensors is also important and has been set to the reasonable value of 20Hz.
Advantage of the closed loop control is the fact that a model is not required to get good performance. No lookup table is used during the process, saving memory space, but a much higher CPU time is required due to the calculation of sums and integrals done constantly.

General schematic of the closed loop designed is shown on figure 4.2. One can identify three main parts that have been colored :

- The green part represents the activating part preventing the device to work for a speed lower than a few meters per second. At low speed, due to sampling rate, calculations may go wrong (division by zero for instance) and this would lead to a very bad behavior of the PID controller. This is the reason why the PID is not activated until a certain speed is reached

- The orange blocks represent some pre-calculation that are necessary for the PID controller. The vehicle speed is calculated using the average speed of the two front wheels. There is no speed sensor on the rear wheels, but since the power train uses a fix ratio gear box, the motors speed is used to calculate each of the rear wheel speed. The longitudinal slip ratio is then calculated using the the vehicle speed and each of the rear wheel speed.

- The blue blocks represent the active parts of the PID control. A constant is defined as a target (it should be the optimum longitudinal slip ratio of the tires) and according to the actual slip ratio, the PID will try to create a throttle command to get and stay as close as possible from the target.

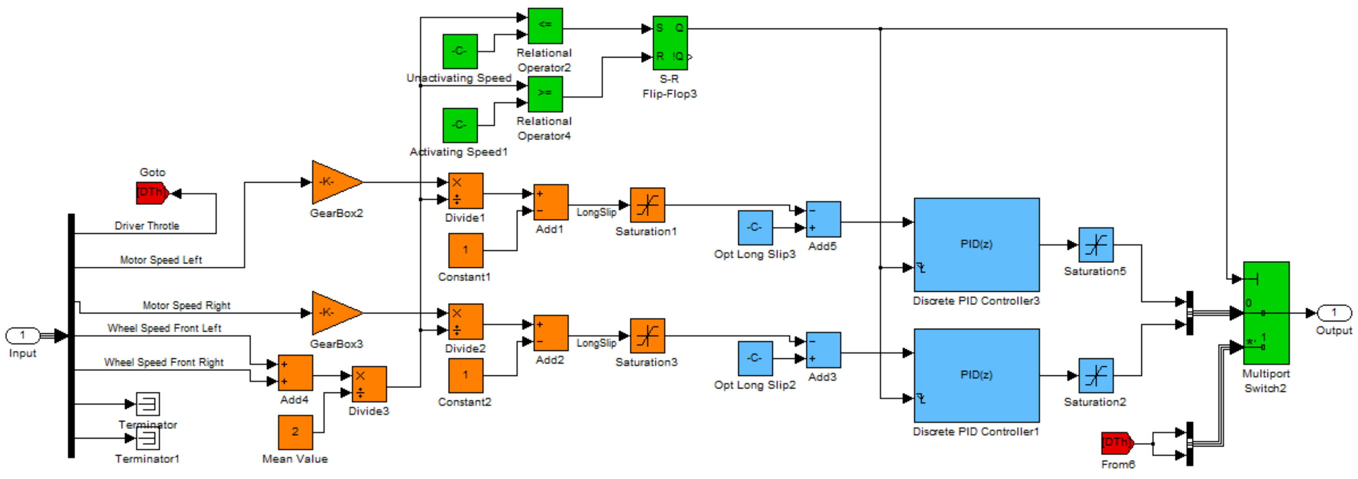


**Figure 4.2.** Simulink - Closed loop launch control

## 4.1.3 Results

### Model based device

Two different ways have been developed and compared to control the acceleration. The first is a simple model based control which does not require complicated software/hardware implementation while the second uses closed loop control using PI controllers. If not specified, tests have been conducted on a 75m race with a road friction coefficient of 1.0 and the same vehicle powertrain model, the only modifications being the control of the throttle command. Here is the comparison of the results of these two:

The open loop model gives very good results with a final time of 4.05s for a road friction coefficient of 1. A time of 3.8s can be achieved for road friction coefficient higher than 1.2 but no lower time has been achieved since this already corresponds to a race with full throttle during the whole run.
In figure 4.3, it can be seen that there are three main parts in the acceleration run.

- A first short transition period (0 to 0.2s) where the control variables are not steady and where the acceleration is increasing quickly until it reaches a steady value of $9.5m.s^{-2}$.

- Then comes a period (0.2 to 2.5s) where all variables are steady. This corresponds to the zone where the electric motor is able to provide a constant torque, and thus a quasi constant acceleration.

- Then comes a third period where the acceleration decreases despite an increase of the throttle command. This corresponds to the field weakening effect. As explained in the PMSM control theory part, the motor is rotating too fast, and higher speeds are only accessible by reducing the torque, leading to a lowered acceleration rate.

It is also interesting to study the influence of the road friction coefficient. Indeed, since this control module is model based, it cannot adapt automatically to different driving conditions (wet track for instance). The friction correction coefficient $f_r^{core}$ introduced in equation 4.3 is supposed to make quick adjustments easy (using for instance an Analog/Digital conversion using a potentiometer on the steering wheel). Simulations have been performed for a wide range of road friction coefficient and friction correction coefficient in order to study the influence of this pair and results are shown on the contour plot in figure 4.4

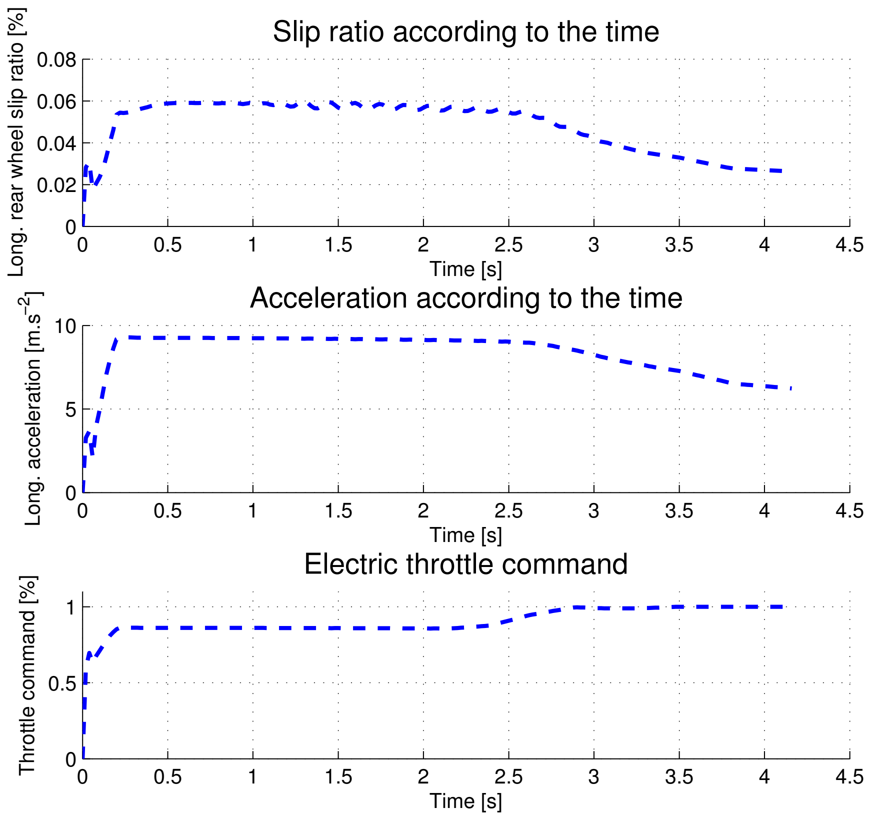


**Figure 4.3.** Simulink - Open loop model based state variables

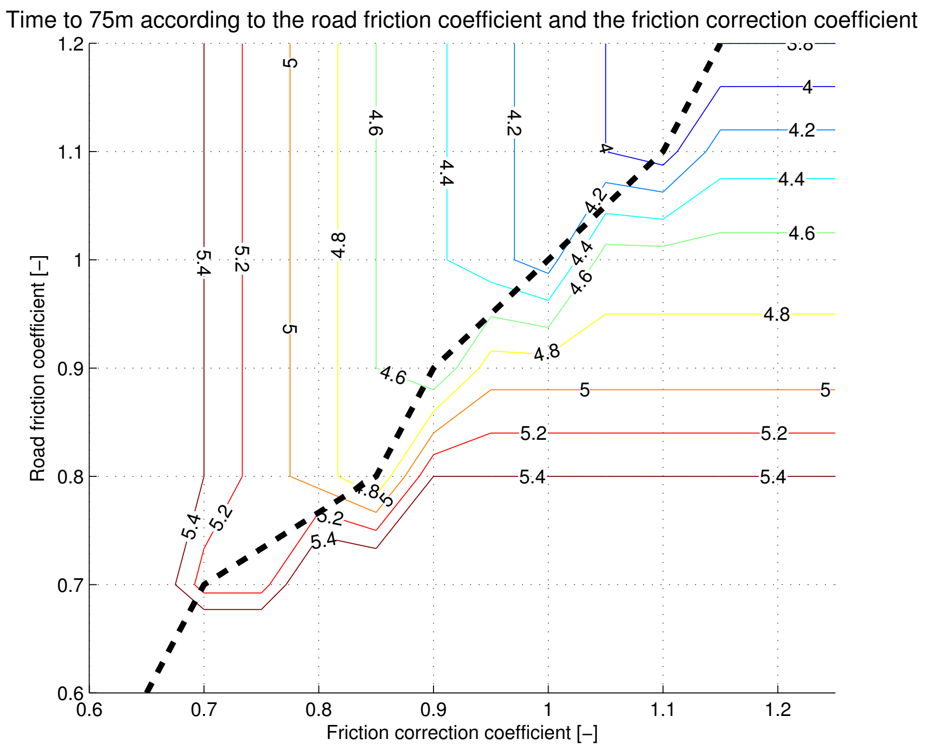


**Figure 4.4.** Influence of the friction coefficient

The black line in figure 4.4 symbolizes the best friction correction coefficient according to the road friction coefficient. It is interesting to note that it can honestly be approximated using $f_r^{corr} = f_r^{road}$, making adjustments very easy.

## Closed-loop device

The closed loop control device managed to give some very satisfying results too. Acceleration time for a road friction coefficient of 1 is 4.1s which is slightly slower than the open loop one.
As one can see on figure 4.5, three phases can also be identified like for the open loop device.

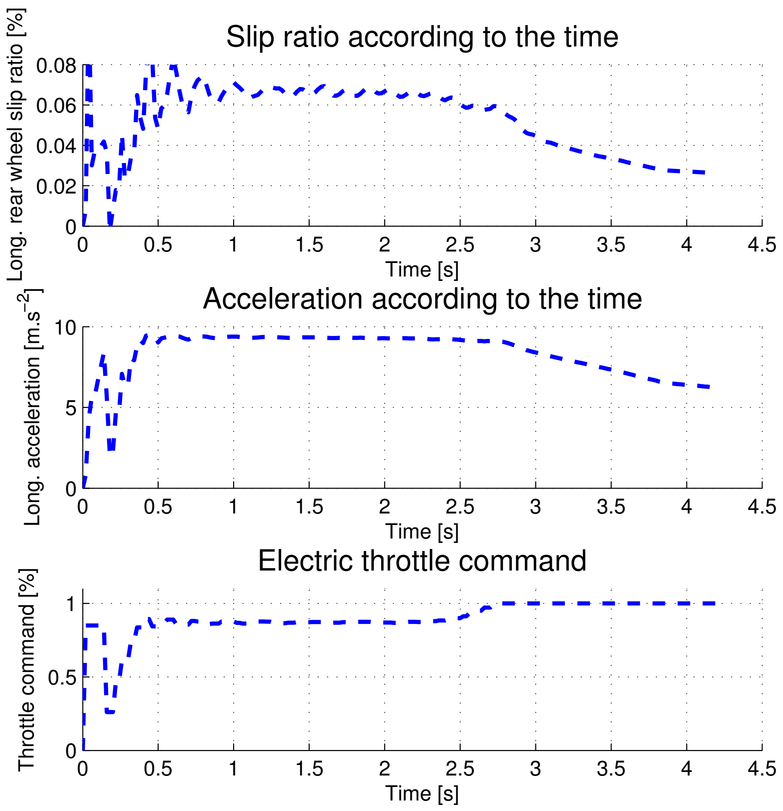


**Figure 4.5.** Simulink - Closed loop launch control state variables

The first transition phase where the acceleration is not maximum is longer though and is the reason for the lower performance. Due to the fact that divisions by zero that may occur for low speed, the throttle is fixed at 85% for a few meters until the PI control can give good results. When this occurs (at about 0.15s), it takes about 0.15s for the PI control to get back to its optimal value. This rising time can be dramatically lowered by trimming the $K_i$ and $K_p$ coefficient, however, better speed implies less stability. On figure 4.5, it is possible to see some parasitic oscillations between 0.5s and 1.5s, but which almost vanish after that. A better rising time would imply these oscillations not to vanish and leading to a dramatical loss of performance.

As promised, the closed loop control gives good results in every condition without any adjustment effort. As shown on figure 4.6, optimal time seems to be reached from a road friction coefficient of 1.2 and then stays steady. It is important to remember than this curve is done without any adjustment of the PI control. The closed loop certainly gives results that are not as good as the model based device due to lags at the beginning and parasitic oscillations, but always gives very consistent results, no matter the racing conditions. May the weather conditions change suddenly, acceleration will always stay close to the maximum achievable while the model based control will require an accurate value for the road friction coefficient to give optimum results for instance.

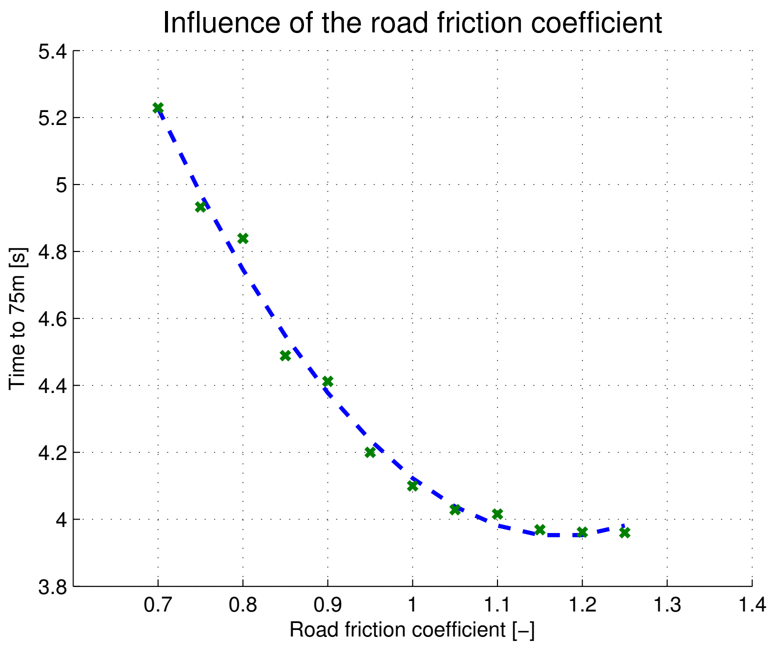


**Figure 4.6.** Road friction coefficient influence on the closed loop control

### 4.1.4 Parameters setup and risk aversion

It was shown in the result part that the open loop launch control device gave very good results provided that its friction correction coefficient is correctly set. However, measurement of extrinsic variables such as the road friction coefficient is not always easy and it might not be reasonable that it is known with a perfect accuracy, and this is the reason why influence of error margin should be studied.

To show this influence, different runs have been simulated with a road coefficient of 1 and results are given by the blue curve on the figure 4.7

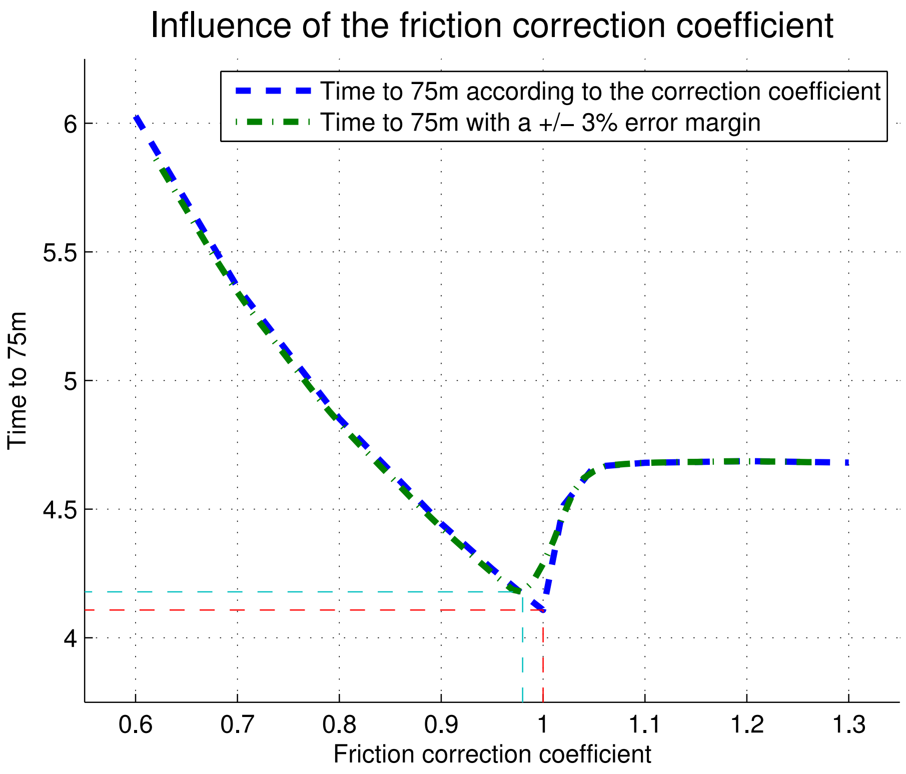


**Figure 4.7.** Influence of the error margin on the friction coefficient

As predicted, the best time is observed when the friction correction coefficient is equal to 1 but time dramatically increases for correction coefficient slightly higher. To model a 6% error margin, real average run time have been computed using this averaging formula :

$$t_{mean}(f_r^{corr}) = \frac{1}{1.03 - 0.97} \int_{f_r^{road}=0.97}^{1.03} t\left(f_r^{corr}\right) df_r^{road} \tag{4.4}$$

Results of this calculation are shown by the green curve. One can notice that the optimal setting for the correction coefficient is now lower than 1. Due to the risk to have overestimated the friction coefficient and that could have dramatic consequences on the final time, it is sensible to aim for an slightly higher best time, but with a better expected value.

## 4.2 Racing performance

### 4.2.1 Torque vectoring main principle

During cornering, load transfer is quite inevitable for a car due to the fact that its center of gravity is above the ground. Its main effect is that the load on the inner wheel is reduced and on the outer wheel the load is increased. Moreover, the maximum force that can be applied by a tyre can be calculated with the friction circle as a first approximation (see fig. 4.8).

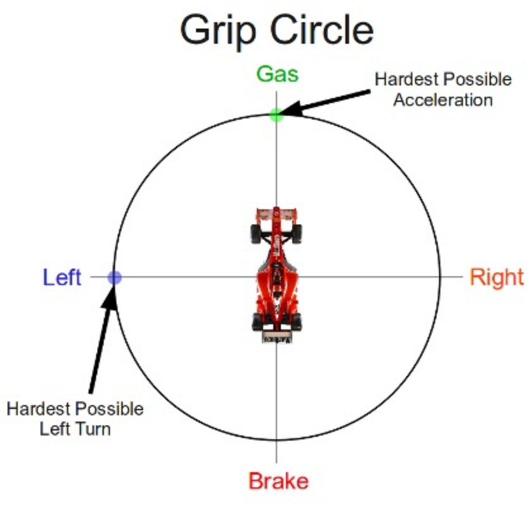


**Figure 4.8.** Schematic of a friction circle for a tire

When cornering to the right, the right rear wheel is unloaded (its friction circle is smaller), and any longitudinal force (acceleration or braking) may be enough to get the wheel spinning. On the opposite, the left wheels are extra-loaded and are able to withstand an extra force.

The main principle of the torque vectoring is then to apply longitudinal force preferentially on the outer wheel (see [7]). The inner wheel can then "focus" on lateral forces and is prevented from spinning. Maximum lateral force is thus increased and it is possible to get higher lateral acceleration.

One can notice than by applying higher longitudinal force to the outer wheel, an other effect will appear :

- If this longitudinal force is used to accelerate the vehicle, then it also tends to make it turn the right way since the unbalance in force between the two rear wheels creates a torque that tends to make the car turn around its $z$ axis the same way that the front wheels tend to

- However if this unbalance is applied in braking, then the torque tends to make the car turn around its $z$ axis the opposite way that the front wheels. This could lead to massive understeer and has not been investigated here since the torque vectoring is only acting in a motor way.

Torque vectoring is not new and the technology and is currently at the core of many research topics (see [7], [8] and [9]). Nowadays, it is already implemented in some luxury cars such as the Porsche 911 Turbo S, or the Audi S5. However, this kind of technology is forced to use mechanical ways to control the torque, such as applying braking force on the inner wheel or using clutches and electronically commanded rear differential. This results in extremely complicated and expensive rear differential (see picture 4.9) that may never appear on common vehicles.

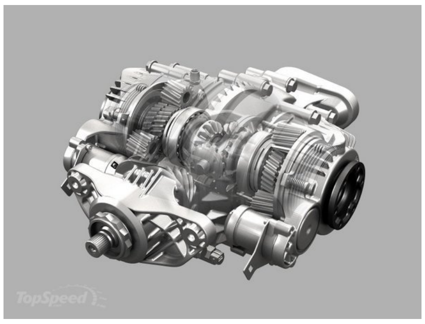


**Figure 4.9.** BMW rear differential designed for torque vectoring [10]

Electric powered car have now the opportunity to benefit from smaller engines dedicated to each wheel. Moreover, electric machines are usually meant to be used both as motors and regenerative brakes which gives the opportunity to perform some torque vectoring even when the longitudinal resultant force should be equal to 0.

### 4.2.2 Implementation of torque vectoring

As previously, a model based control have been used for several reasons:

- Simplicity : simulation can be performed easily and cheaply. The model only requires a precise knowledge of the behavior of the car, and the horizontal accelerations (accessible through the 3-axis accelerometer).
- Adaptivity : in case of small design change (mass transfer, pilot, tire, suspension), a new model can be quickly calculated and tested.

As for the acceleration open loop model, this mode uses the horizontal acceleration $a_y$.

$$throttleTV_{l/r} = throttleDV_{l/r}\,(1 \pm K a_y) \tag{4.5}$$

This model is based on the fact that the outer wheel can withstand an higher torque when it would be better to reduce the torque on the inner wheel. Since the load approximatively increases linearly with the acceleration, and that the torque sent to the wheel is proportional to the throttle command (see chapter on the PMSM), it is acceptable to keep this linear relation between the throttle and the acceleration.

The constant $K$ has been found using empirically. If it is chosen too high, then the outer wheel will spin in turns, while if it is too low, the inner wheel will start to spin first.

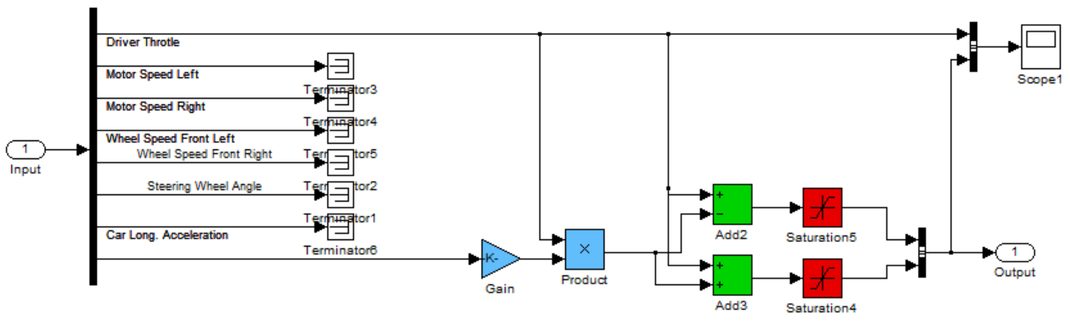


**Figure 4.10.** Simulink model of the open loop torque vectoring

Figure 4.11 shows one of the advantages to distribute the torque more on the outer wheel. One can notice that on the right picture, the torque delivered to the inner wheel is lowered. This adjustment has two main effects: because the torque is lowered, the wheel will not spin, and this lowered longitudinal effort gives the tire the possibility to transmit an higher lateral force as shown by the blue arrow.

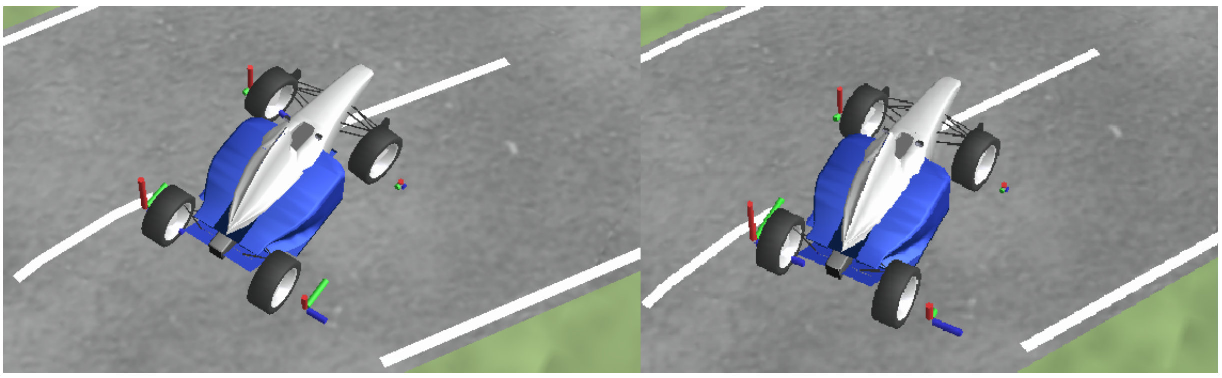


**Figure 4.11.** Same turn with and without torque vectoring (note the difference on the inner rear wheel)

### 4.2.3 Results

Three different tests have been performed to show the influence of the Torque Vectoring.

- In order to show the influence in steady states conditions, the car have been driven on a circular track with a rather small radius
- A step steer in order to show the influence during the transition phase
- Finally, a full lap simulation in order to study the benefits in real race conditions

#### Steady state cornering

The Skidpad test is done on a circular track whose radius is 7,625m to represent the official skidpad event. The input command for the driver is the lateral acceleration (which is directly linked to the speed by $a_y = \frac{v^2}{R}$ since $R$ is a constant here). The driver will try to match this lateral acceleration but may try to go too fast which does not give good results eventually.
Test have been performed on a ten laps basis, and the time is given is the average time (the first lap has been removed to remove the acceleration period)

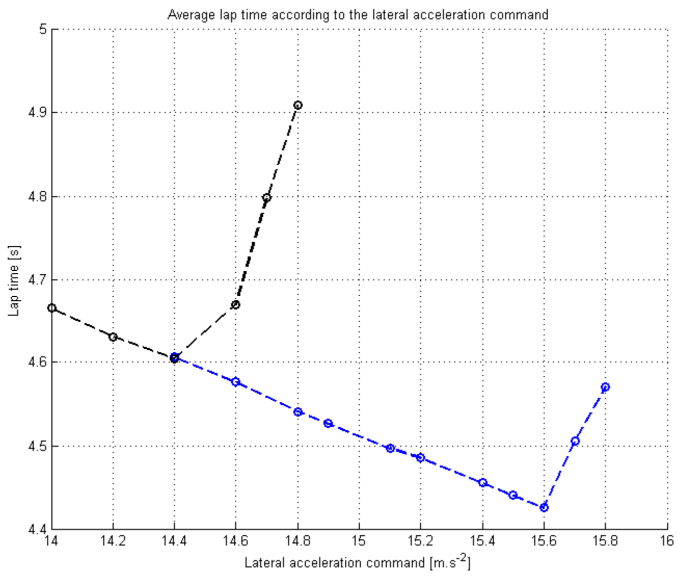


**Figure 4.12.** Lap time according to the lateral acceleration command, with and without torque vectoring

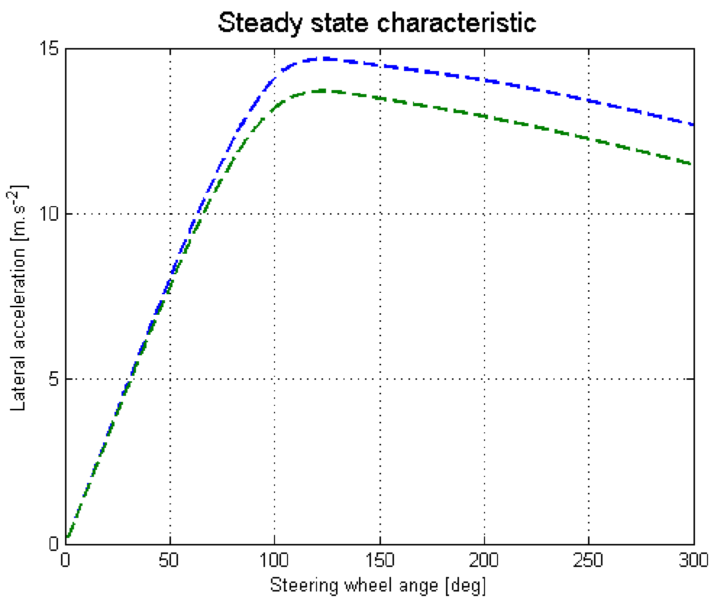


**Figure 4.13.** Steady state characteristic of the vehicle with (blue) and without (green) torque vectoring

As shown on figures 4.12 and 4.13, the torque vectoring gives a real advantage. Thanks to the better torque distribution, the inner wheel has a better contribution in lateral acceleration, and helps improving performances. The minimum lap time is decreased by 4% which is significant in racing and the maximum acceleration reached by 8%.

## Step steer test

The step steer test consists in a quick maneuver where the steering wheel angle is kept constant to 0, and then increases with a step (90deg step during 0.2s).

Figure 4.14 shows the evolution of relevant parameters during the maneuver. Since the torque vectoring implemented is using the lateral acceleration as input, and not the steering angle, there is a delay between the beginning of the step steer, and the beginning of the real torque vectoring process as shown on the second graph. During 0.4s, the torque distribution on the rear wheels is changing a lot. This probably means that the response is not optimal and that it could be improved by taking into account the steering wheel angle.
Nevertheless, the global response of the vehicle compared to a vehicle without torque vectoring is obviously better as one can see on the effective lateral acceleration, or the yaw angle.

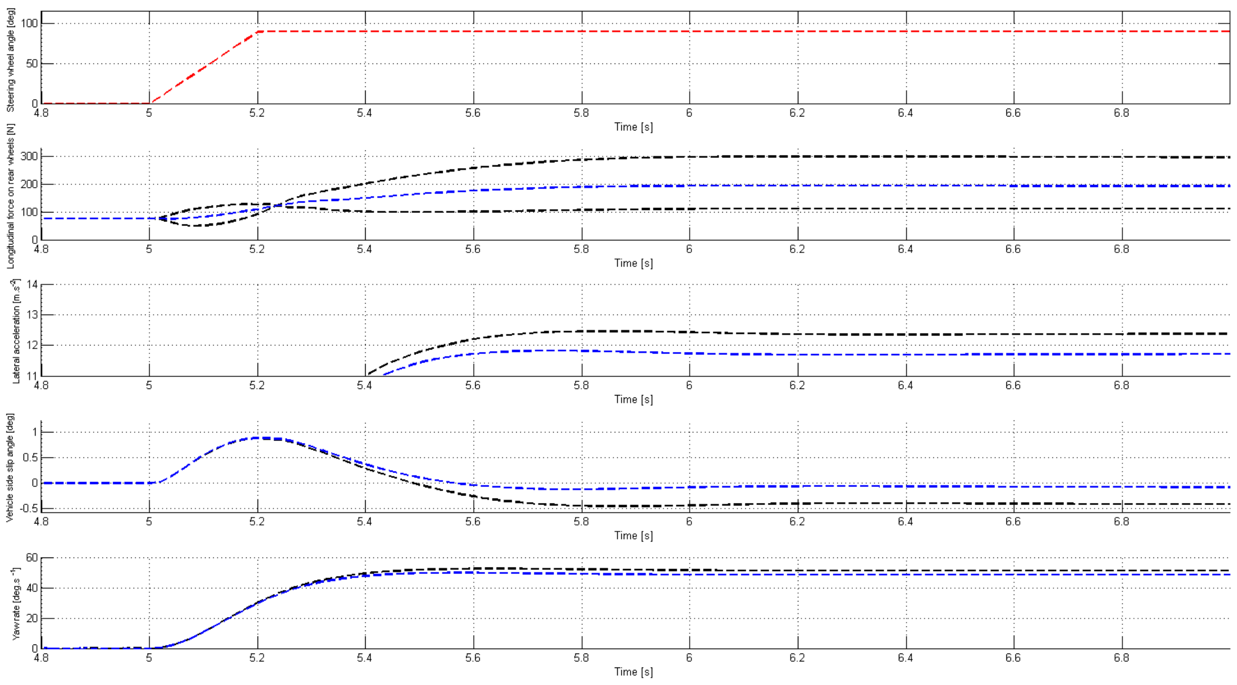


**Figure 4.14.** Vehicle behavior to a step steer (TV On : black, Off : blue)

## Racing test

Finally, the comparison has been done on a full racing lap with both models. The video interface of CarMaker makes this comparison very easy since two different laps can be shown at the same time, making it easy to identify where a certain model may loose some time or be out of the trajectory.

On a single lap, the difference is 0.4s which is shown on figure 4.15. One has to remember that the endurance race would include around 20 of these laps leading to a significant advantage.
It is also important to note that the model using the torque limitation device is easier to control (as shown during the two previous tests), even for an inexperienced driver. During the competition, the track is delimited by cones equally spaced. Each cone touched implies a penalty of 2s and a simple small loss of control during a cornering would lead to at least one pad upside down.

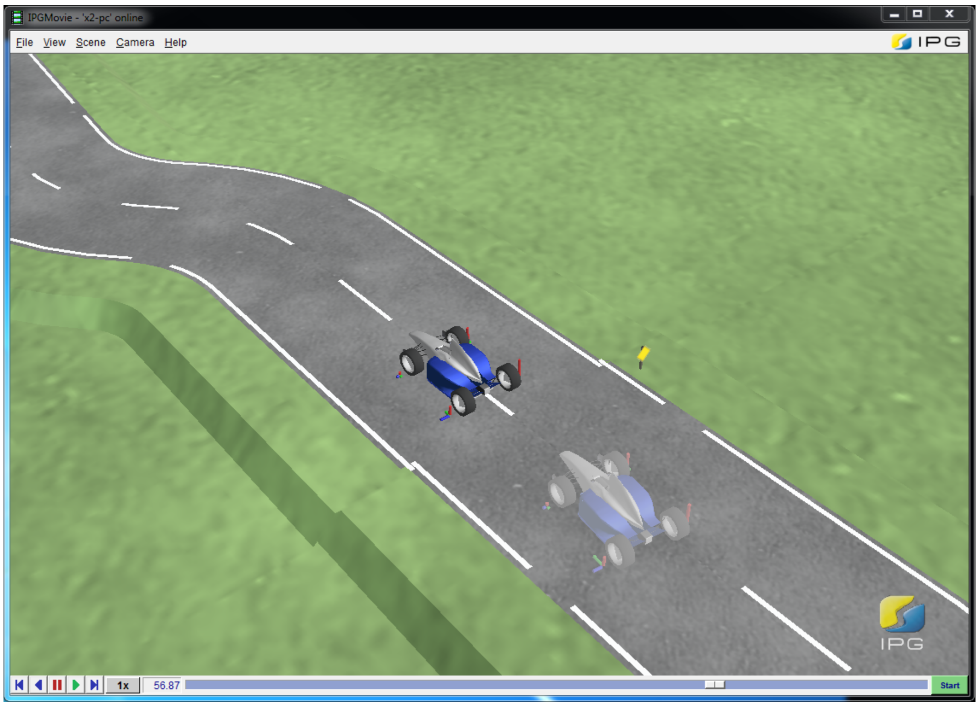


**Figure 4.15.** CarMaker : final position of the car with and without torque vectoring after a single lap ($\Delta t = 0.4s$)

## 4.3 Study of close design vehicles and optimization of the size of the power train

### 4.3.1 Simulation of close design vehicles

The competition rules specify a maximum power of 85kW. Nevertheless, to achieve this power implies the use of heavy motors, heavy gear box, and then more batteries. Knowing the type of tracks used for the competition, where cars barely reach 100km/h and where these 85kW might not be that useful, it is interesting to study the influence of a power decrease. The study was done simulating some close configuration from the R9e to investigate other close design solutions.

Several simulations have been performed using the exact same track on a two lap session ( about 10% of the 22km endurance race). In all case, final time, but also energy efficiency is given. Indeed, as written in the introduction, the total amount of energy used during the endurance event is used to encourage teams to produce a vehicle as efficient as possible.

| Total mass | 85kW | 85kW | 70kW | 70kW | 50kW | 50kW |
|---|---|---|---|---|---|---|
| 250kg | 1,27MJ | 157,8s | 1,24 MJ | 158,1s | 1,2MJ | 160,3s |
| 280kg | 1,45MJ | 158,7s | 1,28 MJ | 159,7s | 1,33MJ | 161,7s |
| 300kg | 1,57MJ | 159,4s | 1,49MJ | 160,9s | 1,42MJ | 162,8s |

**Table 4.1.** Time and energy used for two laps (10% of a race) according to the total mass (car + driver) and power of the vehicle

## 4.3.2 Feasibility of the optimal solutions

As expected, the heavier the vehicle is, the more energy it requires, and the more powerful it is, the faster it is. However, it can be interesting to consider the combined effect of a power reduction, implying a mass reduction in the power train, but also in the batteries required due to the lower consumption. According to the simulation, the current performance of the car lead to a lap time of 159,4s, while it would be possible to reach 158,1s with a 50kg weight reduction. Let's investigate if this could be achievable easily :

- For a very rough approximation, let's consider that the power available for the power train is proportional to its weight, and that the energy stored by the batteries is also proportional to its weight. According to the simulation, a saving of 50kg results in a saving of 19% of energy consumed which represents 11,4kg of an original battery box weighting 60kg.
- The original power train is rated as 85kW for the competition, but is actually a 100kW power train (both electrical machine, and inverters can support it), and its weight is about 60kg. A downsizing if it was possible, would bring it to 42kg (saving of 18kg).
- Total weight saving is less than 30kg, while the original assumption was to save 50kg (to be able to save the 19% of energy).

These simulation show that the current design is at least a local optimum (Power train downsizing would not allow sufficient weight savings to improve performances and efficiency enough).

Other weight savings (by using 10in. wheels, carbon fiber monocoque for instance), would probably lead to more optimal configurations such as the DUT12 developed by the DUT Racing team (Delft, Netherlands) and granted as best electric car in 2012. Its designs relies on an extremely light concept (148kg) with a four wheel drive system.

**Figure 4.16.** DUT12 - Prototype competing in the same category that the R9e and winning the FSG 2012

# Chapter 5

# Conclusion

Electric power trains and torque vectoring technologies will probably be part of the future in automotive and not only in racing and this study succeeded to show some of their advantages in this very specific domain.

The three different parts of the study presented some interesting results: The modelling in Simulink and implementation of an electric power train was achieved and tested with success on a bench in one of the laboratory. The complete model of an electric power train and its control module in CarMaker, taking into account the specificity of the vehicle, of its power train and suspensions was done and tested although it has not been possible to compare it with a real vehicle. And finally, the model of the car was used to test different launch strategies, and to test one torque vectoring method. Compared to the standard electric car model, these strategies lead to substantial improvements in performance and have the advantage not required heavy equipments to be implemented.

The main limit of this study stands in the accuracy of the CarMaker model. Due to lateness in the building of the real vehicle, tests could not be conducted to compare results and fine-tune the different parameters (such as suspension, mass..). Furthermore, CarMaker makes it possible to perform some hardware-in-the-loop tests and should be considered as an axis of improvement.

As a further study, I would recommend to fine tune the CarMaker model once the physical vehicle is ready. Moreover, the current model could be used to simulate a lighter version of the vehicle, with four wheels drive, and thus try to find an other optimum in the power train design.

# Bibliography

[1] *2012 Formula SAE rules*. SAE International, 2012,
visible at : http://students.sae.org/competitions/formulaseries/rules/2012fsaerules.pdf

[2] *Formula Student Electric Rules 2012*. Formula student germany, March 7th 2012 (v1.01),
visible at : http://www.formulastudent.de/uploads/media/FSE_Rules_2012_v1.0.1.pdf

[3] *Milliken Research Associates, Inc*. Formula SAE tire test consortium,
visible at : http://www.millikenresearch.com/

[4] *TYDEX-Format* Description and Reference Manual - Release 1.3, January 9th 1997,
visible at : http://www.fast.kit.edu/download/DownloadsFahrzeugtechnik/_TY100531TYDEX_V1_3.pdf

[5] *Field Weakening control of PMSM* Daniel Fita, Girma Mullisa (Prof.), January 2005

[6] *Fundamental of Electrical drives* G.K. Dubey, Narosa, Kanpur, 4th edition 1997

[7] *Improvement of Vehicle Dynamics by Right-and-Left Torque Vectoring System in Various Drivetrains* Mitsubishi Motors Technical Review, no. 20, 2008, K. Sawase, Y. Ushiroda

[8] *New Vehicle Functionality Using Electric Propulsion*, A. Arikere, 2012

[9] F. Cheli, F. Cimatti, P. Dellacha & A. Zorzutti (2009), *Development and implementation of a torque vectoring algorithm for an innovative 4WD driveline for a high-performance vehicle*, Vehicle System Dynamics: International Journal of Vehicle Mechanics and Mobility, 47:2, 179-193

[10] *BMW rear-axle differential*, visible at : http://www.topspeed.com/cars/car-news/bmw-models will-get-new-rear-axle-torque-vectoring-systems-ar48579.html

# Appendix A

# Appendix

## A.1 Simulink initialization file

```
% Initialisation file for the R9e CarMaker model
%
% Created by Sylvain Blaszykowski for KTH Racing - 2012

 clear all
 clc

 display('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
 display('%
%')
 display('%          KTH Racing R9e - CarMaker model
%')
 display('%
%')
 display('%         Developped by Sylvain Blaszykowski
%')
 display('%
%')
 display('%
%')
 display('%                 Season 2011 - 2012
%')
 display('%
%')
 display('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
display(' ')

display('Initialisation starts...')

%% Run parameters
%   0 for manual throttle
%   1 for racing throttle (torque limitation)
%   2 for acceleration mode (open loop)
%   3 for acceleration mode (closed loop)
DrivingMode = 0;

%% Geometry parameters
wheelBase = 1.55;          % wheelbase [m]
wheelRadius = 20.5*2.54e-2;    % wheel radius [m]

%% Electric motor parameters initialisation
% initialisation of the motor characteristics for the simulation
[iqMatrix,idMatrix,speedList,throttleList] = PMSM();

current2torque = 1; %% current to torque characteristics of the SIEMENS
eMotor = current2torque*iqMatrix;

% creation of the inverse mapping for the control of the PMSM
[I,J] = size(eMotor);
torqueList = [0:10:100];
for j=1:I
    torqueListTemp = eMotor(j,:);
    eMotorRev(:,j) = interp1(torqueListTemp,throttleList,torqueList);
end

eMotorRev(isnan(eMotorRev)) = 1;    % non accessible torque should lead

torqueList = [torqueList 9999];     % torque values higher than 100N.m
eMotorRev(end+1,2:end) = 1;


%% Gear box parameters :
final_ratio = 2.4*2.4;            % ratio between motor speed and wheel


%% programming parameters
```

```
SampleTimeSensor = 0.01;
WheelSpeedSensorNb = 1;
WheelSpeedSensorRate = 1;

%% launch control parameters (closed loop mode)
vMini = 0.5;                     % minimum speed before launch control a
longSlipLaunchControl = 0.08;    % optimal longitudinal slip ratio [%]
KpLaunchControl = 2;             % Kp coefficient of the PI control
KiLaunchControl = 50;            % Ki coefficient of the PI control

%% launch control parameters (open loop mode)
staticTorque = 55;               % optimal torque delivered to a wheel f
dynamicTorqueCoefficient = 2.9; % optimal torque delivered to a wheel a
frictionCorrection = 1.;         % friction coefficient correction

%% racing mode parameters
dynamicTorqueCoefficientLateral = 3;

clear I J j torqueListTemp;

display('All parameters initialised...')
```

## A.2 TYDEX format converter

```
% TYDEX conversion code for the R9e CarMaker model
%
% Created by Sylvain Blaszykowski for KTH Racing - 2012

clear all
clc

display('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
display('%
%')
display('%     KTH Racing R9e - TYDEX conversion code
%')
display('%
%')
display('%         Developped by Sylvain Blaszykowski
%')
```

```
display('%
%')
display('%
%')
display('%                    Season 2011 - 2012
%')
display('%
%')
display('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
display(' ')

hoosier = open('hoosier2058psilat.csv');
hoosier = hoosier.hoosier2058psilat;
hoosier = hoosier(1:22100,:);

[I,J] = size(hoosier);

memoireLoad = 0;
loadList = [];
camberList = [];
loadStart = [];
loadEnd = [];
for i=1:1:I
    newLoad = hoosier(i,11);
    if abs(newLoad-memoireLoad)>150
        memoireLoad = newLoad;
        loadStart = [loadStart,i];
    end
end

loadEnd = [loadStart(2:length(loadStart)),I];


for i=1:length(loadStart)
    derivLoad = diff(hoosier(loadStart(i):loadEnd(i),4));
    offsetStart = find(derivLoad>0.05,1,'first');
    offsetEnd = find(derivLoad>0.05,1,'last');

    loadEnd(i) = loadStart(i) + offsetEnd;
    loadStart(i) = loadStart(i) + offsetStart;
    loadList(i) = -mean(hoosier(loadStart(i):loadEnd(i),11));
    camberList(i) = mean(hoosier(loadStart(i):loadEnd(i),5));
end



fid = fopen('Hoosier.tdx','w');
%% constants
line101 = '**CONSTANTS';
line102 = 'NOMWIDTH                                    inch
7';
line104 = 'NOMDIAME                                    inch
20.5';
line105 = 'RIMDIAME                                    inch
13';
line106 = 'RIMWIDTH                                    inch
7';
line107 = 'MANUFACT
Hoosier';

fprintf(fid, '%s\n',line101);
fprintf(fid, '%s\n',line102);
fprintf(fid, '%s\n',line104);
fprintf(fid, '%s\n',line105);
fprintf(fid, '%s\n',line106);
fprintf(fid, '%s\n',line107);

%% measurchannels
line201 = '**MEASURCHANNELS';
line202 = 'FZH       Vertical force            N
1         0';
line203 = 'SLIPANGL  Slip angle                deg
1         0';
line204 = 'FYH       Lateral force             N
1         0';
line205 = 'MZH       Aligning Moment           Nm
1         0';
line206 = 'INCLANGL  Inclination angle         deg
0         0';

fprintf(fid, '%s\n',line201);
fprintf(fid, '%s\n',line202);
fprintf(fid, '%s\n',line203);
fprintf(fid, '%s\n',line204);
fprintf(fid, '%s\n',line205);
fprintf(fid, '%s\n',line206);

%% datas
line301 = '**MEASURDATA';
fprintf(fid, '%s\n',line301);

figure(1)
clf
leg = [];
leg2 = [];
suptitle('Hoosier - 20.5x7-13')
subplot(221)
title('F_Y vs slip angle and the vertical load (camber = 0deg)');
grid on
xlabel('Slip angle [deg]','fontsize',12)
ylabel('Lateral force [N]','fontsize',12)
subplot(222)
title('MZ vs slip angle and the vertical load (camber = 0deg)');
grid on
xlabel('Slip angle [deg]','fontsize',12)
ylabel('Aligning moment [N.m]','fontsize',12)

subplot(223)
title('F_Y vs slip angle and camber angle (F_Y = 1100N)');
grid on
xlabel('Slip angle [deg]','fontsize',12)
ylabel('Lateral force [N]','fontsize',12)
subplot(224)
title('MZ vs slip angle and camber angle (F_Y = 1100N)');
grid on
xlabel('Slip angle [deg]','fontsize',12)
ylabel('Aligning moment [N.m]','fontsize',12)

for i=1:length(loadStart)
    load = [];
    sa = [];
    Fy = [];
```

```
MZ = [];

measurePoints = loadEnd(i)-loadStart(i)+1;

load = -hoosier(loadStart(i):loadEnd(i),11);
sa = hoosier(loadStart(i):loadEnd(i),4);
Fy = -hoosier(loadStart(i):loadEnd(i),10);
MZ = hoosier(loadStart(i):loadEnd(i),13);
ia = hoosier(loadStart(i):loadEnd(i),5);

matr = [sa,load,Fy,MZ,ia];
matr = sortrows(matr);

sa = matr(:,1);
load = matr(:,2);
Fy = smooth(matr(:,3),10);
MZ = smooth(matr(:,4),20);
ia = 1/10*round(10*matr(:,5));

output = [load sa -Fy MZ ia];
output = output(1:80:end,:);

[K,L] = size(output);

if abs(mean(ia)-0)<0.2
    for k=1:K
        fprintf(fid,'%6.2f %6.2f %6.2f %6.2f %6.2f\n',output(k,:));
    end
end

if abs(mean(ia))<0.2
    subplot(221)
    hold all
    p(i) = plot(sa,Fy,'--','LineWidth',2);
    subplot(222)
    hold all
    plot(sa,MZ,'--','LineWidth',2);
    subplot(224)
    leg = [leg,mean(load)];
end
```

```
    if abs(mean(load)-1100)<50
        subplot(223)
        hold all
        plot(sa,Fy,'--','LineWidth',2);
        subplot(224)
        hold all
        plot(sa,MZ,'--','LineWidth',2);
        leg2 = [leg2,mean(ia)];
    end
end

subplot(221)
legend([num2str(leg(1),4), 'N'],[num2str(leg(2),4), 'N'],[num2str(leg(3
subplot(223)
legend([num2str(leg2(1),4), 'deg'],[num2str(leg2(2),4), 'deg'],[num2str

fclose(fid);

system('tireutil -if Hoosier.tdx -of Hoosier -ofbin Hoosier.bin')

if true
    system('move Hoosier C:\MasterThesisFinal\Data\Tire\FS\Hoosier');
    system('move Hoosier.bin C:\MasterThesisFinal\Data\Tire\FS\Hoosier.
end
```