Torque Vectoring for a Formula Student Prototype

João Pedro Marques Antunes
joao.pedro.antunes@tecnico.ulisboa.pt

Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal

June 2017

Abstract

Torque Vectoring (TV) has the objective to substitute the need of a mechanical differential, while also improving the handling and response of the Formula Student vehicle. This thesis addresses the design of a torque vectoring system in an rear wheel driven formula student prototype. The proposed solution resorts to a PI controller for yaw rate tracking with an evenly distributed torque to each wheel. Also an LQR scheme is discussed, for tracking the yaw rate and the lateral velocity. To assess and design, first a 7 degree of freedom (DOF) non linear model is constructed, followed by a linear 2 DOF model, both models are validated with real data. The linear model, is used to design and simulate the proposed controllers. When the controller is within the desired parameters it is tested in the non linear model. Tests with the vehicle are performed to verify the contribution of the controller to the overall performance of the vehicle.

Keywords: Torque Vectoring, Formula Student, PI controller, LQR controller, vehicle model

1. Introduction

Each year that goes by sees an increase in sales of personal use of electric vehicles (battery EVs, plug-in hybrids and regular hybrids), all of which rely on electric motors as a basis or an aid to propulsion. With this increase, manufacturers are starting to explore new ways of implementing electric motors. They are favoring the use of 2 or 4 motors instead of just a single motor. Using more motors is advantageous: they are smaller, and lighter when compared to a combustion engine. It also opens the opportunity for vehicle stability control systems directly at the motors, like Electronic Stability Program (ESP) and Anti-lock Braking System (ABS) which gives the vehicle better stability and manoeuvrability. This paper addresses another type of vehicle stability control called Torque Vectoring (TV). By controlling the amount of torque distributed to each driven wheel, the system has the potential to improve both the stability and response of a vehicle without compromising safety and drivability, as well as reducing the weight and packaging.

Torque vectoring has made a big impact on Formula Student events. Since its appearance in 2011, electric cars have consistently been in the top position every year. 2016 FS competition once again showed that a four wheel drive electric vehicle is the winning concept.

Figure 1: FST06e Electric vehicle

Developing a torque vectoring system can be tackled resorting to several different approaches. The majority [7, 5] use the yaw rate of the vehicle as the reference for the controller. More advanced solutions [8, 10, 6] use the sideslip angle and/or a combination of yaw rate and sideslip angle. The choice on the strategy will depend on the available sensors.

A variety of control laws can be used for torque vectoring. The most basic method is to distribute the left and right torque, proportional to the amount of steering input $\Delta T = f(\delta)$.
The proportional integral derivative (PID) controller is the classic control structure and the most commonly used in practical applications. It is a straight forward method to implement and tune

[8, 11].
Sliding mode control is a non linear control design methodology used by several researchers to achieve the objectives of tracking the yaw rate and slip angle [8, 10].
Predictive control estimates the future states of the vehicle in order to find the best control input [2, 3]. Some authors also try to implement Fuzzy control to create a set of rules for the allocation of the torque [13].

2. Vehicle Model

Vehicle dynamics is the area devoted to the development of models that describe the behavior of a vehicle for any given set of inputs and disturbances. Modelling this type of system is a very complex and challenging task. A lot of different approaches and models can be required depending on the needs of the user. A complex multi-body system (with 20+ degrees of freedom), or a simple two degree of freedom model [4, p.6].

The vehicle model used to study a torque vectoring control system will typically have seven degrees of freedom. The lateral and longitudinal velocities of the vehicle (v_x and v_y respectively) and the yaw rate $\dot{\psi}$ constitute three degrees of freedom related to the vehicle body. The wheel velocities of the four wheels, the front left wheel (w_{fl}), front right wheel (w_{fr}), rear left wheel (w_{rl}), and rear right wheel w_{rr}) constitute the other four degrees of freedom [9].

The dynamic model described is complex. When studying lateral dynamics, one of the most common models used is the linear bicycle model. This model is a simplification in which it is only being considered the lateral velocity v_y and yaw rate $\dot{\psi}$ on the model [4, 7, 5, 13].

2.1. Non Linear Model

The FST06e is powered by two *Siemens* permanent magnetic synchronous motors each one with an RPM range of 0 to 8000, and producing a maximum torque of 107 Nm. With a planetary gear set fixed with a gear ratio of 4.1:1, amplify at the wheel with a total of 876 Nm. The model of the vehicle is divided in two parts. Where more information is available (parametric values and/or behavior), a more accurate and complex model is used, otherwise a more simplified approach is applied(ex: tire model). The full model consists in a horizontal model which describes the model position and orientation of the vehicle. A model of the steering kinematics, which describes the relation between the steering wheel and the actual steering of each wheel. A simplified tire model to calculate the forces acting on the wheels. A balance at the wheel between the applied torque and the tire longitudinal force. Figure 2 shows the block diagram of the model.

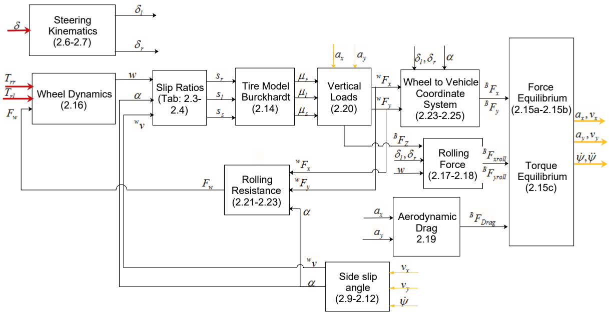

Figure 2: Simulink Schematic of Non Linear Model

2.2. Linear Model

The model described until now is complex. The goal is to increase the lateral dynamics of the vehicle. When studying lateral dynamics one of the most common models used is the linear bicycle model. The way the model is presented, the equations only generate lateral force by the steering input. The goal with the torque vectoring is to generate yaw moment based on controlling the torque (longitudinal force) at the driven wheels. For this it is necessary to introduce a new term M_z that represents the additional yaw moment. Table 2.2 summarizes the values from the FST06e need for the model.

Term	Symbol	Value	Units
Yaw rate	$\dot{\psi}$	-	$[rads^{-1}]$
Velocity	v_{x0}	[0,40]	$[ms^{-1}]$
Rear Stiffness	$C_{y,r}$	21429	$[Nrad^{-1}]$
Front Stiffness	$C_{y,f}$	15714	$[Nrad^{-1}]$
Inertia moment	I_{zz}	120	$[Kgm^2]$
Mass	m	356	$[Kg]$
front wheelbase	l_f	0.873	$[m]$
rear wheelbase	l_r	0.717	$[m]$
steering angle	δ	[-3.3,3.3]	$[rad]$
Yaw moment	M_z	-	$[Nm]$
Lateral velocity	v_y	-	$[ms^{-1}]$
Gear ratio	G_r	4.4	-
Half track	t_r	0.65	m
Wheel Radius	R_w	0.265	m

Table 1: Linear model values

$$\dot{x} = Ax + Bu_1 + Eu_2$$

$$\dot{x} = \begin{bmatrix} \dot{v}_y \\ \ddot{\psi} \end{bmatrix}; \qquad u_1 = M_z; \qquad u_2 = \delta$$

$$A = \begin{bmatrix} -\dfrac{C_{y,f} + C_{y,r}}{m v_{x0}} & \dfrac{-l_f C_{y,f} + l_r C_{y,r}}{m v_{x0}} - v_{x0} \\ \dfrac{-l_f C_{y,f} + l_r C_{y,r}}{I_{zz} v_{x0}} & -\dfrac{C_f l_f^2 + C_r l_r^2}{I_{zz} v_{x0}} \end{bmatrix} \tag{1}$$

$$B = \begin{bmatrix} 0 \\ \dfrac{1}{I_{zz}} \end{bmatrix} \tag{2}$$

$$E = \begin{bmatrix} \dfrac{C_{y,f}}{m v_{x0}} \\ \dfrac{l_f C_{y,f}}{I_{zz}} \end{bmatrix} \tag{3}$$

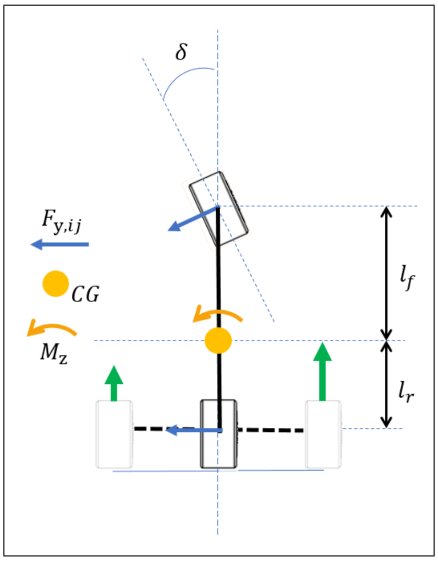

Figure 3: Vehicle's linear model free body diagram

2.3. Validation

The constant radius turn (skidpad) is defined by ISO 4138 [1]. According to the norm, the test should be performed with a minimum circle radius of 30m. For the FST06e, the available test track is at the campus in front of the main build in the parking lot, with a maximum circle radius of 7.5m. Another test track is at the International Kartodromo in Palmela, with a maximum radius of 15m. Therefore, the constant radius test had to be modified.

In the Formula Student competition, one of the disciplines is to perform a constant radius turn in a track with a radius of 8.75 m. To be more close to the competition conditions, the tests are modified and performed with a range of 5-9m radius.

In total three data sets are available. These tests were done in different days and in two different places. "Test 1" and "Test 2" were both done at the campus. The radius of the circle is the same, but the driver varied the velocity. "Test 3" is a test with a bigger radius done in Palmela. Table 2.3 summarizes the tests.

	Test 1	Test 2	Test 3	Unit
Radius of circle	5.62	5.62	9.02	$[m]$
Global Velocity	7	8.5	9.3	$[ms^{-1}]$
Steering Angle	110	100	71	$[deg]$

Table 2: Test setups

During the tests some data points are being monitored. The inputs are the global velocity, the steering wheel angle. The output values are the longitudinal, lateral acceleration and yaw rate. Although, many more data can be logged from the car, these are sufficient to conclude if the models are accurate enough [12, p. 345]. Table 3 summarizes the variables being monitored.

Model Variables		
Input	Steering angle	δ
	Torque/Velocity	T, v_{CG}
Output	Longitudinal acceleration	a_x
	Lateral acceleration	a_y
	Yaw rate	$\dot{\psi}$

Table 3: Measured variables

Based on the tests, the first step is to verify if the data provided from the sensors are accurate. Based on the gathered data, presented in table 2.3 knowing the radius of the track as well as the vehicles' speed, the yaw rate is calculated. Which is then compared to the measured yaw rate. In a steady state cornering the yaw rate can be calculated by the following expression.

$$\dot{\psi} = \frac{v_{CG}}{Radius} \tag{4}$$

$$\dot{\psi} = \frac{9.16}{8.3875} \approx 1.09 rad \approx 62 deg$$

	Real Data [deg/s]	Theoretical [deg/s]	Difference [%]
Test 1	72.3	71.37	1.29
Test 2	72	86.65	20
Test 3	58.6	63	6.98

Table 4: Yaw Rate comparison between theoretical and real data

Based on the values presented in table 4 it can be inferred that the sensors used in the car are working and giving correct values. "Test 2", the big difference presented is that for the car to be able to maintain that velocity and steering angle, the trajectory of the vehicle (around the track) had to be bigger.

The data is then compared with the model. Figure 4 shows the comparison between the yaw rate from the vehicle and the yaw rate from both linear and non linear simulations. The simulated values are very close to the real data. The data presented illustrates that the vehicle made 4 turns. Two to the left and two to the right ($\approx$ 5s each). Negative values of yaw rate represent the car cornering to the right (in a clockwise way) while positive values represents the car cornering left (counter clockwise).

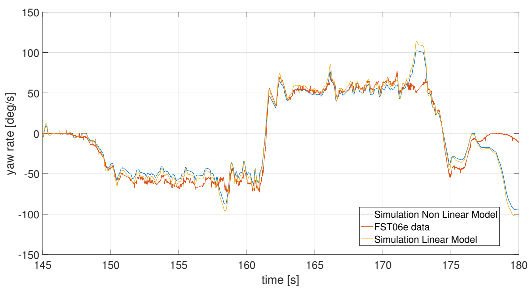

Figure 4: Comparison between real data form the fst06e and both linear and non linear simulation during a skidpad to the left and right from test1

3. Control System Design

Based on the availability, sampling time and quality of the sensors, the strategy for the calculation of the reference value is presented, followed by the choice and tuning of the controllers.

3.1. Reference Value

Before the introduction of the control algorithm, it is necessary to define a reference signal. Various authors purpose the calculation of the reference based on the velocity and steering angle of the car. It is assumed that the vehicle is in a steady state [9].

$$K_u = \frac{l_r m}{C_{y,f}(l_f + l_r)} - \frac{l_f m}{C_{y,r}(l_f + l_r)} \tag{5}$$

If the under-steer gradient, K_u is positive ($K_u > 0$): The car is said to have an under-steer behavior (under yaw rate).
If K_u is negative ($K_u < 0$): The car is said to have an over-steer behavior (over yaw rate). And if $K_u = 0$, it means that the car is neutral-steer (ideal yaw rate).

A neutral-steer vehicle has the smallest possible turning radius for a given velocity, which corresponds to optimal performance. Therefore, it would be assumed that a neutral-steered vehicle should be chosen as the reference. However, this approach would put the car on the verge of over-steer instability, so a slightly under-steered is taken as reference. This reference is much closer to neutral-steered than the actual car, so the controller can improve the yaw rate. The ideal or desired yaw rate can be defined by the velocity and the radius of the curve:

$$\dot{\psi}_{desired} = \frac{v_{CG}}{R} \tag{6}$$

Given the velocity and steering angle of the car and with the known steer gradient and wheelbase it is possible to know the turning radius.

$$\frac{1}{R} = \frac{\delta}{(l_r + l_f) + K_u v_{CG}^2} \tag{7}$$

Combining both expressions 6 and 7, the under-steer gradient and the road radius, gives the desired yaw reference.

$$\dot{\psi}_{desired} = \frac{v_{CG}}{(l_r + l_f) + K_u v_{GC}^2}\delta \tag{8}$$

The desired yaw rate is a function of the velocity, steering and characteristic of the car. The under-steer gradient K_u can be tuned in for each driver preference. The smaller the under-steer gradient the bigger the difference between the desired and actual yaw rate, more will the car have neutral steer characteristics.

To determinate the values of the under-steer gradient it is necessary to know the cornering stiffness of both the front and rear tires. The best approach is to test the tires in a tire testing facility, derive a tire model and from it retrieve the cornering stiffness.

For Formula Student teams, data for the tires are available from the TTC (Tire Test Consortium). The data is only available as raw data and therefore a lot of pre-processing and model fitting is necessary, it is possible to get a simple graph that correlates the vertical load with the cornering stiffness directly from the raw data.

As a first approximation, the vertical load is simply the weight of the car distributed on each wheel. Because the car has a weight distribution of 40% front and 60% rear the vertical load will be respectively 697N in and 1046N.

$$C_{y,f} = 546N/deg \qquad C_{y,r} = 745N/deg$$

3.2. Maximum Yaw Value

With all the various possible implementations and control strategies, some limitations are valid for all torque vectoring controllers. These limitations are related to the physical properties of the vehicle like, the maximum yaw rate possible, the maximum tire adhesion, microprocessor computing time, etc. Depending on the entry speed of the car, it will be able or not to achieve the desired yaw. If entering in a corner to fast the road may be unable to provided the necessary tire forces, and the car just goes forward, thus under-steering. The solution is to bond by it's limiting factor the tire-road coefficient.

$$v_{CG}\dot{\psi} + a_x\beta + \frac{v_{CG}\dot{\beta}}{\sqrt{1+\tan\beta^2}} \leq \mu g \tag{9}$$

In equation (9), if considering that the car has a small heading angle, the equation can be further simplified and reduced to:

$$\dot{\psi}_{max} = \sigma\frac{\mu g}{v_{CG}} \tag{10}$$

where σ represents a tunability factor to take into account changes in the friction coefficient from different types of pavement. The yaw rate reference is used as long as it doesn't pass the maximum possible yaw rate.

$$X(m,n) = \left\{ \begin{array}{ll} \dot{\psi}_{des}, & |\dot{\psi}_{des}| \leq \dot{\psi}_{max}| \\ \pm\dot{\psi}_{max}, & \text{otherwise} \end{array} \right\} \tag{11}$$

3.3. Proposed Controller

The added momentum results from the difference between the left wheel torque T_{rl} and the right wheel torque T_{rr}. This difference multiplied by the half track of the car will be the additional yaw momentum M_z in the model. If the right wheel as more torque than the left wheel the car will have a positive yaw momentum, thus turning to the left, if the opposite happens the car while have a negative momentum and will turn right.

$$M_z = (T_{rr} - Trl)t_f \tag{12}$$

But the torque at the wheel is not the same torque at the motor. Between the motor an the wheel there is a planetary gear set, this gear set multiplies by a gear ratio G_r the torque from the motor $T_{wheel} = Gr * T_{motor}$. Then the torque at the wheel has to be divided by the wheel radius R_w to obtain the torque at the ground. These values can be found in table 2.2.

Putting together all this information and with equation 12 the yaw moment from the difference in torque is given by.

$$\Delta T = \frac{R_w}{2t_rG_r}M_z = \frac{0.265}{2*0.65*4.4}M_z = 0.05M_z \tag{13}$$

Thus rewriting the state space equation from 1 in order of the delta torque instead of the yaw moment.

$$u_1 = \Delta T \tag{14}$$

$$B = \begin{bmatrix} 0 \\ 1 \\ \hline 0.05 * I_{zz} \end{bmatrix} \tag{15}$$

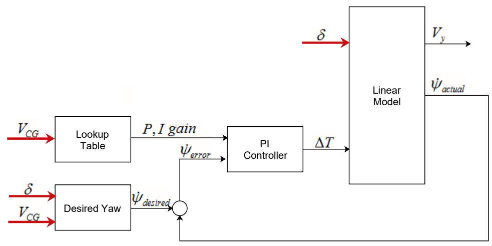

Figure 5: Schematic of simulink linear model

3.4. LQR

Linear quadratic regulator (LQR) is an optimal control strategy for linear systems. In the design of this type of control an optimal gain K is calculated based on the performance index J. The advantage of the PI controller is its simplicity in implementation and understanding of what is happening in terms of allocated torque, but this simplicity also has its drawbacks. If the tire-road friction is wrong or the calculated reference yaw rate is excessive for the current state of the vehicle, the vehicle behavior may become unstable. To further improve the torque control, a LQR controller is presented. The controller will also use both models, both lateral velocity v_y and yaw rate $\dot{\psi}$ are considered state variables and ΔT the control input. The difference is that now it will also be monitored the lateral velocity, which will give a more robust control.

$$\Delta T = K_r\dot{\psi} + K_v v_y \tag{16}$$

The performance index may be written in the following way:

$$J(u) = \frac{1}{2}\int_{t_0}^{t_f} [(X_d - X)^T Q(X_d - X) + u^T Ru]dt \tag{17}$$

where: u = M_z, R = weight factor of the control effort, Q = Penalization Matrix for the states (v_y,r), $\dot{\psi}_{des}$

$$J = \int_{t_0}^{t_f} \left[\frac{1}{2}(\dot{\psi} - \dot{\psi}_{des})^2 + \frac{1}{2}w\Delta T^2\right]dt \tag{18}$$

Where $\dot{\psi}_{des}$ is the desired yaw rate of the vehicle. Minimizing this will lead to a vehicle with very close to neutral steer behaviour. Not forgetting that has discussed, the control effort ΔT must be constrained both due to the maximum torque possible and the tires limit.

Using the linear model, from which the gains will be calculate, it is necessary to make a small change to the state space matrix. A new entry was added because the system doesn't have an integrator. So the signal is fed back to the input.

$$\dot{x} = \begin{bmatrix} \dot{v}_y \\ \ddot{\psi} \\ \dot{\psi} \end{bmatrix} \tag{19}$$

$$A = \begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{20}$$

The K gains are calculated by solving the Riccati equation, choosing appropriated values of Q and R. The controller is tuned by varying both values, First R and then Q.

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 \tag{21}$$

Solving for our case will be:

$$\begin{aligned} &2a_{11}p_{11} + 2a_{21}p_{12} - R^{-1}\frac{p_{12}^2}{I_{zz}} = Q_{11} \\ &(a_{11} + a_{22})p_{12} + a_{21}k_{22} + a_{12}k_{11} - R^{-1}\frac{p_{12}p_{22}}{I_{zz}} = Q_{12} \\ &2a_{12}p_{12} + 2a_{22}p_{22} - R^{-1}\frac{p_{22}^2}{I_{zz}^2} = Q_{22} \end{aligned} \tag{22}$$

This system can be solved and the optimal feedback gain matrix K will be:

$$K = R^{-1}B^T P \tag{23}$$

$$K = R^{-1}\begin{bmatrix} 0 & \frac{1}{I_{zz}} \end{bmatrix}\begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix} \tag{24}$$

$$K = R^{-1}\begin{bmatrix} \frac{p_{12}}{I_{zz}} & \frac{p_{22}}{I_{zz}} \end{bmatrix} \tag{25}$$

Resorting to *matlab* to solve equation 22 (with the use of the command "*lqry(sys,Q,R)*"), The matrix is Q = $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 10^6 \end{bmatrix}$

And R value of R = 10^{-6}

4. Implementation

The FST06e works in ADC values, which in practice means that what is going to be controlled is the ADC value that comes from the pedal (requested by the driver). This value enters the Torque Vectoring PCB, and the algorithm then decides the value which should go to each motor, sending an ADC value based on the current state. The first one is a lower limit dead zone, which ensures that, although the pedal spring may not always return to the same position, the car will not start as soon as one hits the pedal. The second is a saturation of the maximum value of torque that can go to the motors.

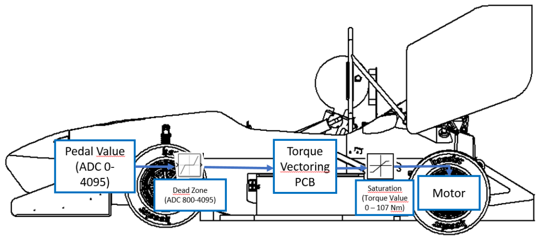

Figure 6: Scheme from the Pedal Value to the Motor Value

4.1. Communication

The FST06e communication protocol is CAN protocol. A Controller Area Network (CAN) is a specialized internal communications network designed to allow microcontrollers to communicate with each other without the use of a host computer. Once received the sensor data the first step is to convert the data into SI units like radians, torque... After all the calculations are done, it is necessary to convert to send back to the CAN line. Figure 7 illustrates this process.

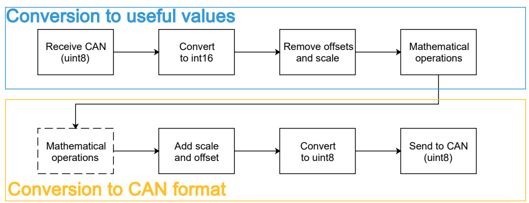

Figure 7: Conversion process from CAN to SI units and back to CAN

The use of this protocol is made possible by using a self develop board with the DSpic6012A microcontroller. This microcontroller can run up to 112 MHz, and it has a floating point unit, which is particularly useful when performing calculations.

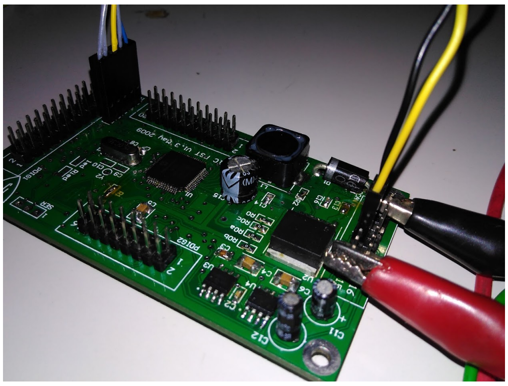

Figure 8: DSpic6012A microcontroller plugged in for testing

4.2. Sample Time
All sensors have different sampling period. Table 4.2 shows the sample time from each one. Based on the maximum sampling time that each sensor was able to provide. It is advantageous to have more data points available and discard/average them, than having less points.

Sensor	Aquisition Time (ms)
IMU	8
Enconders Rear	50
Enconders Front	100
GPS	40
Steering Angle Enconder	40
Pedal	10

Table 5: Sensors form the FST06e

4.3. Filter Sensors
Figure 9 shows the yaw rate from the IMU. After some testing it is concluded that it is necessary to use a 3 point median filter and a 1st order low pass filter. The median filter is necessary to remove the outliers, then a low pass filter with a cutoff frequency (f_c) of 3Hz proved to be sufficient to remove the noise (because the car is a mechanical model, the working frequency is very low).

The first order low pass filter is just the current yaw rate $\dot{\psi}(i)$ multiplied by a time constant(α) summed with the previous value $\dot{\psi}$ multiplied by $(1-\alpha)$. The time constant α is a value that is chosen and tuned based on the cut-off frequency (f_c) until the filter is within acceptable values. The time variation depends on the acquisition rate of the IMU, for the FST06e it is 8ms.

$$\dot{\psi}_i = \alpha\dot{\psi}_i + (1-\alpha)\dot{\psi}_{i-1} \tag{26}$$

$$\alpha = \frac{dt}{\frac{1}{2\pi f_c} + dt} \tag{27}$$

The time variation dt depends on the acquisition rate of the IMU, for the FST06e it's 8ms. The time constant (α) is a value that is tuned until the filter is within acceptable parameters. The goal is to have the data filtered with the least possible delay.

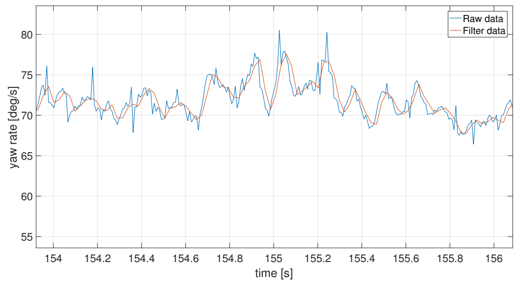

Figure 9: Results of the filter

4.4. Fail Safe System
After the code is written and checked it is necessary to create a system of fail-safes. These measures are necessary because prototypes can be dangerous, and when testing a new control algorithm for the first time it is necessary to implement fail safe systems. These types of systems not only ensure that in case of sensor failure or communication problems with the microcontroller, the controller will not became unstable, but also that when the car is stopped or the pedal value is zero there is no torque applied

- **Denominator 0** - The most basic fail safe is to guarantee that no division has the value 0 in the denominator (for instance the case of the yaw reference). To address this problem an if case is implemented, when the velocity is zero the yaw reference will return zero.

- **Velocity <5** - When the car has a speed lower than 5m/s, the output of the torque vectoring is zero. This ensures that when the car is at low speeds, ex: when parking or moving to the track, no torque vectoring is active.

- **Number of messages** - Because the CAN protocol is very complex and has a lot of messages and priorities to handle at once, it may happen that sometimes some messages can't be sent, or be corrupted. Given this possibility, some sensors can still have old values. To protect against this problem, a routine was created to check the total number of messages received. If the total number of messages received is below a certain threshold the Torque Vectoring is deactivated.

- **Maximum allowed torque** - Depending on the test track it is not always possible to have the maximum torque available. Therefore, a parameter that regulates the maximum torque that comes out of the controller is introduced.

- **Pedal torque:** - The last safety measure is to ensure that when the driver lifts his feet from the pedal, the torque vectoring is not active. This safety measure is implemented because a normal Formula Student driver is always pressing the throttle pedal while cornering.

4.5. Code

This section assesses the necessary steps in the development and implementation of the C code. The approach can be divided in the following steps:

- Receiving messages

    Conversion from ADC to SI Units

- Calculations

    Fail Safe System

    Reference Yaw

    Controller Torque Output

- Send Message

    Conversion from SI Units to ADC

Figure 10 shows the workflow of the implementation of the controller in C code. In the yellow box, the car sends its values in ADC, then converts to SI units. Every 20ms, it moves to the blue box where it verifies if all the necessary messages are present to calculate the torque. If there are not enough messages, it skips the calculations and the torque given from the pedal to the motors. If the necessary number of messages are present, the code executes the calculations, first computing the reference value followed by the torque. In the green box, it checks the value of pedal, checks the value of the pedal, and returns the new torque value if it is higher than zero or the previous value if it is zero.

5. Results

The results of the implemented controller are discussed. The car is tested in the same conditions as in section 2.3. The driver performs a skidpad with a radius of 5m trying to match the same conditions that are used to validate the model (same speed and steering angle). With the data acquired with the different controllers, a comparison is made, and the gain in vehicle performance evaluated.

5.1. No TV

Figure 11 shows the variables, desired yaw rate (blue color), current yaw rate (red color), and the speed of the vehicle (yellow color). The driver does 5 laps of approximately 5s each (80s-110s) to the left, cornering in a counter-clockwise way, and from 110s-120s the driver exists and starts

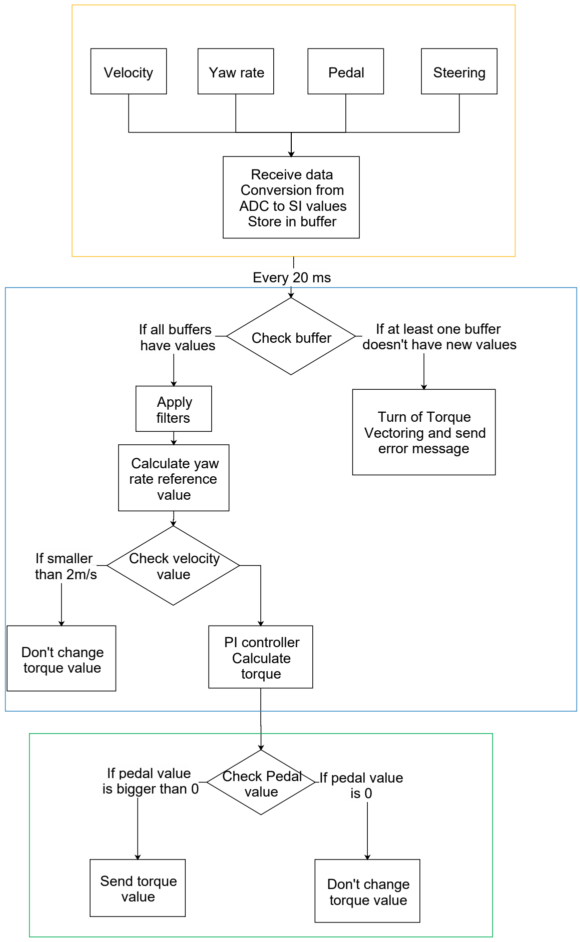

Figure 10: Schematic of the implementation of the C code

cornering in the opposite way in a clockwise way, from (120s-150s).

The vehicle speed is also plotted in the graph to confirm if the driver is maintaining it's speed, recalling equation 8 the reference is calculated based on the velocity and the steering angle, if both are constant then the desired yaw will also be constant. Because the velocity is constant, the desired yaw rate variations that appear are the steering corrections made by the driver. It can be observed that the first laps (80s-100s) the inconsistency of the driver is diminishing until at (100s-110s) the driver is constant. The same happens from (140s-150s).

Focusing now just on the current yaw rate value and desired yaw value it can be observed that the current value of the yaw rate is 70 deg/s, and the desired yaw rate is 81deg/s, which means that with torque vectoring there can be a possible gain of 11deg/s. Also, it is important the see that if we look at test1 from table 4, the yaw rate is quite similar, which makes sense because the test track is the same.

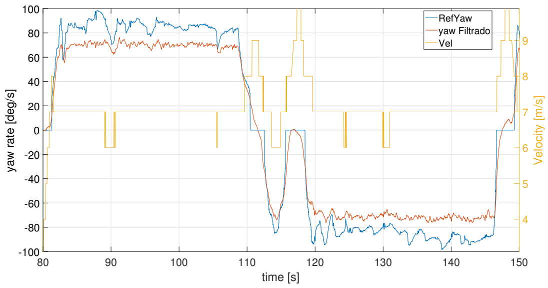

Figure 11: Data logged form the vehicle during the test with no torque vectoring. The variables presented are: Yaw rate reference, yaw rate, and global velocity

5.2. TV

After the baseline test is performed, the test with the controllers are performed. Figure 12 shows the same variables as in figure 11, but this time the torque vectoring controller is acting on the vehicle. The data presented is from the car cornering to the right. As the driver is starting the corner, like in figure 11 at first (525s-535s) the driver is inconsistent but starts to became consistent the more time he is in the corner (535s-555s). That is when it can be seen that the controller is improving the yaw rate of the car.

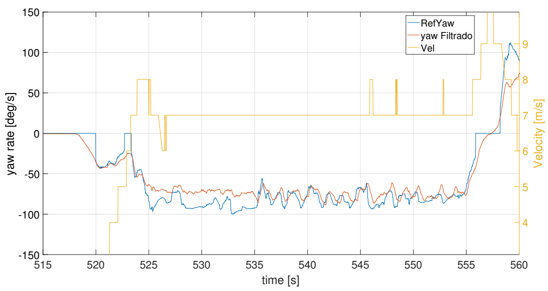

Figure 12: Data logged from the vehicle during the test with the PI controller. The variables presented are: Yaw reference, yaw rate, and global velocity

Figure 14 shows the torque given to each wheel. It can be seen that effectively the controller is changing the torque in each wheel depending on the drivers input, when the car is producing to much yaw rate the controller puts more torque in the wheels, when the car has to much yaw rate, the controller removes torque. Also in the same figure are the simulated torque requested in the non linear simulation for the same steering and velocity inputs as in the test.

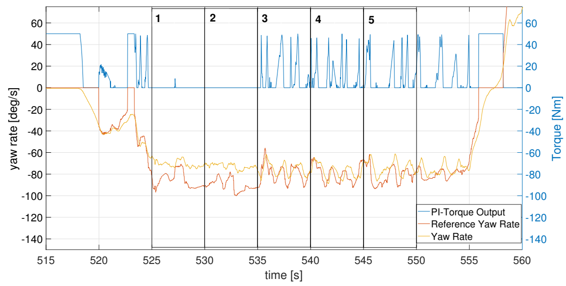

Figure 13: Data logged from the vehicle which shows, PI - controller output, desired yaw rate and yaw rate

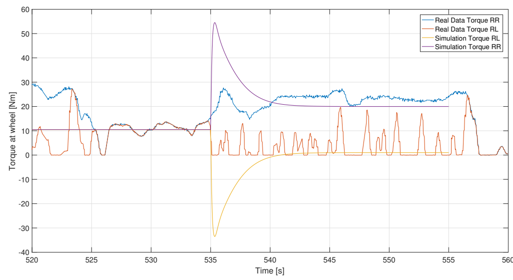

Figure 14: Comparison of torques between simulation and real data from the car

Table 6 show that the proposed controller contributes to an increase in lateral performance, the vehicle as more yaw rate, allowing to achieve a higher cornering speed, which translates in a reduction of 0.38s per lap.

Table 6: Comparison between torque vectoring and no torque vectoring for the PI controller

	Yaw rate [deg/s]	Velocity [m/s]	Time [s]
No TV	70	8.4	4.97
TV	74	8.8	4.59

5.3. LQR

Once the tests of the PI controller are done, the LQR controller is tested. Due to tire wear and electronic faiulure it was not possible to test the LQR controller in the car. Figure 15 compares the real data with the PI controller and the simulation of the LQR controller, it can be seen that the LQR controller would be very similar.

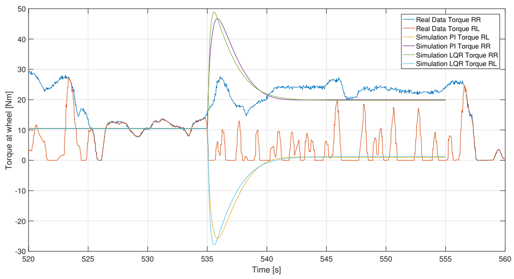

Figure 15: Comparison of torques between simulation and real data from the car for the LQR controller and PI controller

5.4. Conclusion

A nonlinear model for simulation has been presented, as well as a linearized model for yaw rate control through the application of a differential torque. Both models are validated with logged data from the electric prototype.

Two control strategies are presented. A gain scheduling PI controller for controlling the wheel torques based on the yaw rate. And an LQR approach for controlling the wheel torques based on the yaw rate and lateral velocity. It was concluded that for low velocities the models, are not much affected by the lateral velocity gain, and in terms of response it was quite similar to the PI controller. The main advantage of the LQR is its robustness and lack of a gain scheduling when compared to the PI controller.

The microcontroller developed by the team proved sufficient for receiving and filtering the data, and also run the controller every 0.2s. The fail safe system was crucial to ensure that the tests with the controller could be executed in safe conditions. Looking at all the different test procedures, it is clear that the implementation of the torque vectoring has an effect in the vehicle behaviour. When the controller is used a gain of 5% is achieved, which translates in a reduction of 0.38s in a skidpad when compared to the same situation but without the torque vectoring.

References

[1] *Passengers cars Steady-state circular driving behaviour Open-loop test methods ISO 4138:2004*. Fourth edition edition, June 2012.

[2] S. Anwar. Yaw stability control of an automotive vehicle via generalized predictive algorithm. *Proceedings of the 2005, American Control Conference, 2005.*, pages 435–440, June 2005.

[3] S. D. Cairano, H. E. Tseng, D. Bernardini, and A. Bemporad. Vehicle yaw stability control by coordinated active front steering and differential braking in the tire sideslip angles domain. *IEEE Transactions on Control Systems Technology*, 21(4):1236–1248, July 2013.

[4] R. B. Dieter Schramm, Manfred Hiller. *Vehicle Dynamics: Modeling and Simulation.* Springer, 1^{st} edition, 2014. ISBN 978-3-540-36045-2.

[5] J. Ghosh, A. Tonoli, and N. Amati. A torque vectoring strategy for improving the performance of a rear wheel drive electric vehicle. *2015 IEEE Vehicle Power and Propulsion Conference (VPPC)*, pages 1–6, Oct 2015.

[6] G. Kaiser, F. Holzmann, B. Chretien, M. Korte, and H. Werner. Torque vectoring with a feedback and feed forward controller - applied to a through the road hybrid electric vehicle. *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 448–453, June 2011.

[7] G. Kaiser, Q. Liu, C. Hoffmann, M. Korte, and H. Werner. Torque vectoring for an electric vehicle using an lpv drive controller and a torque and slip limiter. *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 5016–5021, Dec 2012.

[8] L. D. Novellis, A. Sorniotti, P. Gruber, and A. Pennycott. Comparison of feedback control techniques for torque-vectoring control of fully electric vehicles. *IEEE Transactions on Vehicular Technology*, 63(8):3612–3623, Oct 2014.

[9] R. Rajamani. *Vehicle Dynamics and Control, Mechanical Engineering Series.* Springer, 1^{st} edition, 2005. ISBN 978-0387263960.

[10] D. Rubin and S. Arogeti. Vehicle yaw stability control using rear active differential via sliding mode control methods. *21st Mediterranean Conference on Control and Automation*, pages 317–322, June 2013.

[11] A. Stoop. Design and implementation of torque vectoring for the forze racing car. Master's thesis, Delft Center for Systems and Control, 2014.

[12] L. N. Uwe Kiencke. *Automotive Control Systems: For Engine, Driveline and Vehicle.* Springer, 2^{nd} edition, 2005. ISBN 978-3-540-26484-2.

[13] C. Zhao, W. Xiang, and P. Richardson. Vehicle lateral control and yaw stability control through differential braking. pages 384–389, July 2006.