UNIVERSITÀ POLITECNICA DELLE MARCHE

Faculty of Engineering

Bachelor of Science in Computer and Automation Engineering

Bachelor's Thesis

Control oriented modelling of four wheeled electric vehicles

Advisor:
prof. Andrea Bonci

Candidate:
Sebastian Nicolas Giles

July 2018

Acknowledgements

Thanks to someone

Sommario

I motori elettrici non contengono parti mobili all'infuori dell'assieme dell'albero motore, questo li rende più compatti e più semplici rispetto ai motori termici di prestazioni simili. Per questo motivo l'utilizzo di più motori meno potenti per la trazione dei veicoli è fattibile senza incorrere in eccessivo peso o complessità. L'impiego di un motore per ogni ruota motrice può ridurre peso, dimensioni e complessità della trasmissione. Questo è dato dall'assenza dei differenziali e dal fatto che la potenza meccanica possa essere prodotta più vicino alle ruote.

I motori possono inoltre essere controllati individualmente per produrre coppie differenti, risultando in un momento di rotazione attorno all'asse verticale, questa strategia di controllo è chiamata torque vectoring. Il torque vectoring può essere usato per corregere una vettura sovrasterzante o sottosterzante, migliorando la sicurezza nei veicoli stradali e la maneggevolezza nelle auto da competizione. Il controllo della trazione è inoltre ottenibile indipendentemente per ogni ruota, permettendo di percorrere curve a velocità maggiori.

L'obiettivo di questo studio è la derivazione di un modello matematico di un veicolo a quattro ruote utilizzabile in simulazione e per lo sviluppo di controllori per il torque vectoring e il controllo della trazione nei veicoli elettrici.

L'interesse per lo sviluppo del modello è sorto in vista di una prossima partecipazione alle competizioni di Formula Student Electric. Il modello è quindi concepito per applicazioni competitive. L'alta rigidezza sospensiva, diffusa nelle vetture che competono in queste competizioni, è stata tenuta in considerazione per ottenere alcune approssimazioni.

La modellazione delle dinamiche di rollio rende prevedibili i transitori di trasferimento di carico, garantendo maggiore precisione durante le manovre ad alta frequenza come i cambi di corsia, le piccole chicane e gli stretti slalom che tipicamente compongono i percorsi degli eventi di Autocross ed Endurance in Formula Student. L'inclusione del

comportamento di beccheggio e rollio permette anche di studiare la correlazione tra i trasferimenti di carico e l'escursione delle sospensioni, essendo quest'ultima una grandezza fisica facilmente misurabile in tempo reale e quindi utilizzabile come ingresso al sistema di controllo del veicolo. Le dinamiche di sterzo per le ruote posteriori sono state incluse nella seconda versione presentata di questo modello poichè il regolamento Formula Student permette esplicitamente l'utilizzo di quattro ruote sterzanti e la modellazione non ha richiesto alcun impegno aggiuntivo.

Il capitolo 3 descrive fisicamente un modello a sei gradi di libertà, successivamente ne viene fornita la formulazione lagrangiana. Nel capitolo 4 il modello viene esteso con 6 gradi di libertà aggiuntivi per la descrizione della rotazione delle ruote e delle dinamiche di sterzo anteriore e posteriore.

La formulazione lagrangiana è stata costruita simbolicamente utilizzando la MATLAB Symbolic Math Toolbox per facilitare le derivate e le trasformazioni di coordinate. Sono state scritte delle funzioni per risolvere algebricamente le equazioni di Lagrange e fornire una rappresentazione in spazio di stato del sistema. Tutti gli script sono scritti in forma vettoriale rendendo il codice più compatto e leggibile. Le simulazioni sono state eseguite in ambiente simulink, in cui i due modelli matematici ottenuti sono stati implementati in forma modulare, accompagnati da un modello empirico per le forze di attrito generate dagli pneumatici. L'utilizzo di un'animazione 2D della vettura ha facilitato la risoluzione dei problemi rendendo più intuibile il siginificato dei risultati numerici.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

The automotive market share held by hybrid and electric vehicles is growing. This growth is mainly possible thanks to advances in energy storage technologies and manufacturing processes. While there are many environmental advantages, an electric powertrain also offers greater flexibility during the design process of the automobile.

In comparison to similarly performing internal combustion engines, electric motors are smaller and simpler as they have only one moving part that is the rotor and driveshaft assembly, this makes it possible to use more than one motor without adding too much manufacturing complexity.

Using one motor for each driven wheel can reduce the size, weight and complexity of the trasmission. This is beacuse mechanical power can be produced closer to the wheels and the need for differentials is eliminated.

These motors may also be individually controlled to produce different amounts of torque (see figure 1.1) resulting in moment about the vertical (yaw) axis, this kind of control strategy is called torque vectoring. Torque vectoring may be used to correct an oversteering or understeering vehicle, benefitting consumer vehicles with increased safety and race cars with improved handling.

Individual wheel traction control is also possible, allowing greater cornering speed in the racing environments that allow it.

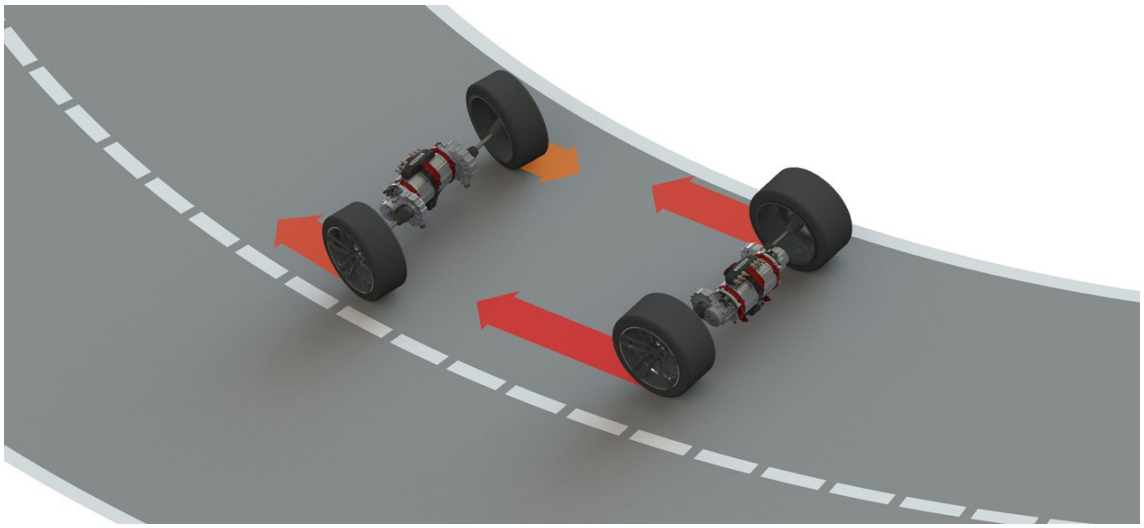

Figure 1.1: The Rimac Concept One EV powertrain applying different amounts of force at each road contact point.

1.2 Objectives

The objective of the present work is to derive a mathematical model to be used in simulations and in the development of control algorithms for traction control and torque vectoring applications in electric vehicles. Future participation in the Formula Student electric competitions has been the main motivation behind the development of the model. It is therefore conceived with racing in mind. The tipically high suspension stiffness of cars competing in this category is taken into account when making approximations.

Modelling of the roll dynamics renders lateral load transfer transients predictable, allowing for greater accuracy in high frequency manouvers such as lane changes, small chicanes and slaloms which make up the Formula Student Autocross and Endurance courses (like the one shown in figure 1.2).

Inclusion of the roll and pitch behaviour in the models also allows correlation of vertical forces to the suspension travel, the latter being an easily measurable physical quantity that can be added as an input to the vehicle control system.

The rear steering dynamics were included in the second model as Formula Student rules explicitly allow it and their modelling did not cost any extra effort.

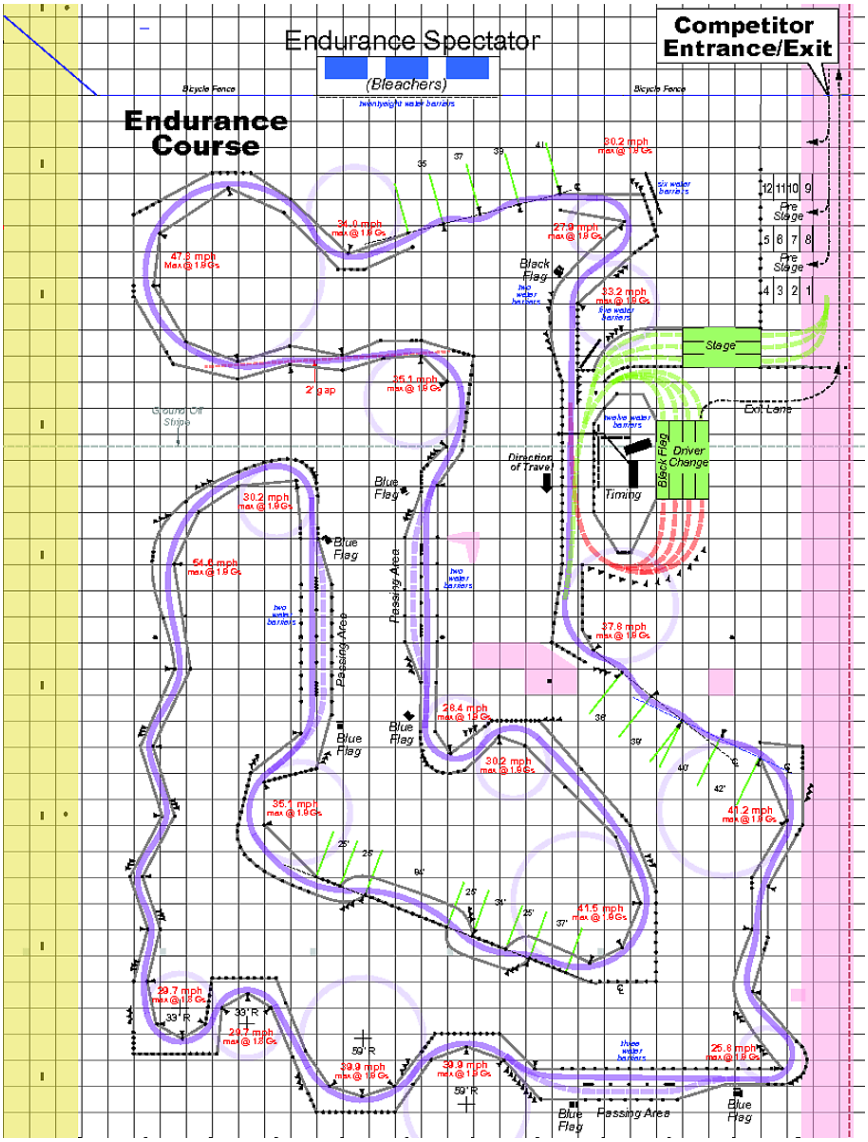

Figure 1.2: Example of a Formula SAE Endurance Course.

1.3 Approach

In Chapter 3 a six degree-of-freedom model is physically and mathematically presented. The lagrange formulation of the system is then given. Chapter 4 extends the model to 12

degrees-of-freedom describing the four wheel rotations and the front and rear steering dynamics. The Lagrange formulation is symbolically constructed using the MATLAB Symbolic Math Toolbox to facilitate co-ordinate transformations and differentiations. Functions were written to algebraically solve the Lagrange Equations and generate the state space representation of the system. All scripts are written in vectorial form allowing for compact and more readable code. Simulations were run in the Simulink environment with a 2D graphical representation of the car as a troubleshooting tool. A brief overview of tyre modelling was required in order to run realistic simulations.

Chapter 2

Tyres

Tyres are the main source of external forces and moments acting upon any road vehicle[8], for this reason, understanding their behaviour is key in making accurate predictions about vehicle dynamics. The tyre is a very complex system exhibiting high non linearities, especially in the wide operating range imposed by racing. A realistic model of the road-vehicle interaction is clearly necessary in order to obtain significant results from simulations but the detailed study of tyres is far beyond the objectives of this work. The present chapter provides only the fundamental concepts behind tire modelling needed to introduce the numerical model that was used.

2.1 Tyre modelling basics and definitions

All forces acting between the vehicle and the road are distributed over the surfaces of the so called *tyre contact patches*. As the tyre contact patches size and shape are dependent on the operating conditions [9], for each wheel, we consider a *tyre contact point*, C, as the point at which an applied vector can approximate all forces acting between the tyre and the road.

The definitions used in this work are consistent with the SAE J670 standard, although simplified.

Various definitions for this point may be used to include more or less complicated effects due to tyre compliance, such as dynamic caster or centrifugal deformation. Modelling of such phenomena would require extensive experimental data and is considered to be outside the scope of this research, thus, the following simplified definition for the tyre contact point

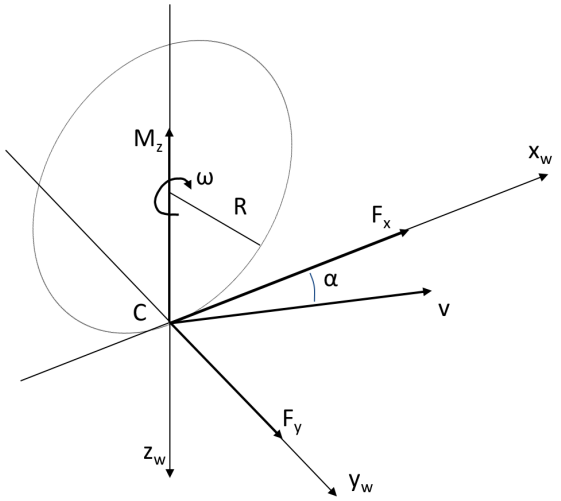

Figure 2.1: Forces and velocities in the wheel reference system.

was deemed satisfactory. The road is assumed to be a flat and planar surface and the tyre is to be considered as an infinitely thin rigid disk tangent to this plane. The tangent point will be considered as the tyre contact point.

The wheel co-ordinate system is required for the definition of all the related physical quantities. See figure 2.1.

We consider a reference system with origin in C, whose x axis lies along the intersection between the road plane and the wheel plane, such that it is generally oriented in the direction of motion, and whose z axis is pointing down, perpendicular to the road plane, the y axis is determined by asserting $Cxyz$ to be a right-handed cartesian coordinate system.

The angle between the xz plane and the tyre plane is known as the camber angle, it is assumed to be zero throughout this work.

The x, y and z axes are respectively associated with the *longitudinal*, *lateral* and *vertical* dimensions of the wheel.

As tyre compliance is neglected in this work, the vertical force F_z acting through the tyre and its contact point is considered as indipendent of tyre behaviour. This choice also excludes all transient responses of the rubber. The longitudinal and lateral friction forces, F_x and F_y, are to be computed since these are what effectively make the vehicle accelerate

and corner and are of primary interest.

Consider a wheel having radius R and angular velocity ω about it's spin axis, passing through the wheel center and orthogonal to the wheel plane. We consider ω as positive when the upper half of the wheel is moving in the positive x direction. The lowest point of the wheel is then moving with velocity $v_w = -\omega R$ in the wheel x direction. By considering the velocity vector of the point C in a road-fixed reference system and letting v_x and v_y be its components along the wheel x and y axes the longitudinal velocity of the road with respect to the wheel system is $v_{rx} = -v_x$. It is therefore possible to define the longitudinal slip as the difference between these two velocities at the contact point:

$$v_{sx} = v_w - v_{rx} = v_x - \omega R.$$

We further define the slip ratio as

$$\kappa = -\frac{v_{sx}}{v_x} = \frac{\omega R}{v_x} - 1.$$

The slip angle α is the angle formed between the contact point velocity vector in the road frame and the wheel x axis. It can be formally defined by the equation

$$\tan \alpha = \frac{v_y}{v_x}.$$

F_x and F_y scale almost linearly with the vertical load exerted on the tyre, for this reason it makes sense to define the longitudinal and lateral friction coefficients as

$$\mu_x = \frac{F_x}{F_z} \qquad \mu_y = \frac{F_y}{F_z}$$

κ and α are the primary variables affecting the friction coefficients and are used to estimate μ_x and μ_y. However, there are other factors that define the steady state tyre behaviour, some of which are relatively straight-forward, such as camber angle, vertical load, wheel rotational velocity, tyre pressure and temperature and others which are extremely difficult to quantify, such as asphalt characteristics and rubber aging.

The typical correlation between friction coefficient and slip ratio is shown in figure 2.2, ideally, all curves pass through the origin, due to the fact that without slip the speed of the wheel is the same as that of the contact point, and no friction forces can be generated. In reality, there is a slight offset caused by rolling resistance and asymmetric tyre contructions [9].

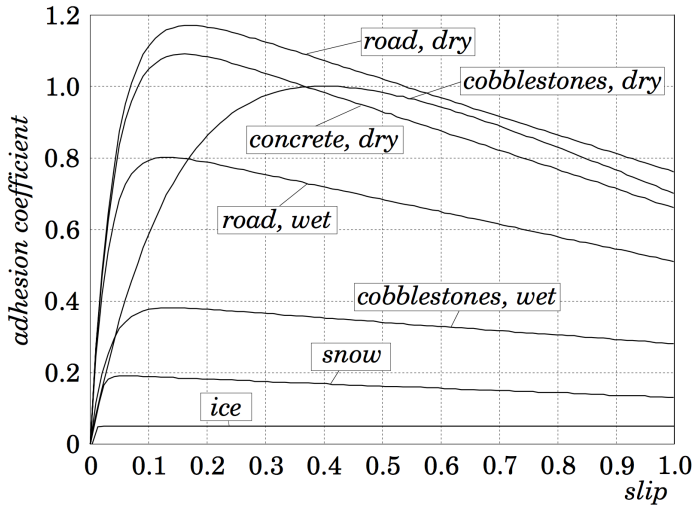

Figure 2.2: Longitudinal friction coefficient under various conditions. [source: [6]].

It is also clearly noticeable that in order to obtain maximum longitudinal traction their will be an optimal amount of slip ratio. Limiting the tractive torque to not exceed this value is the exact purpouse of launch control systems.

Another effect of the pneumatic tyre is the so called self aligning torque, this moment is exerted about the tyre z axis and is an important contributor to the steering effort which is applied to the steering wheel, this will be relevant to the steering dynamics introduced in the 12 DoF model presented in Chapter 4.

2.2 Magic Formula tyre models

A series of semi-empirical tyre models was developed during the past 30 years by Hans B. Pacejka of Delft University.

These models are based on the same parametric mathematical expression, nicknamed *Magic Formula* because of the good fitting properties it offers despite the lack of a physical foundation. The general form of these functions is

$$R(k) = d\sin(c\arctan(b(1-e)k + e\arctan(bk)))$$

where R is the physical quantity to be estimatated and k is the most relevant variable whilst d, c, b, e and k are fitting coefficients, which may in turn depend on other variables. The set of equations released in 1996, called *Deft Tyre 96* [10] was the first to include

combined effects of longitudinal and lateral slip, subsequent editions of the model have been conceived to model tyre pressure changes [1] and more. The version which has been chosen in this work is the PAC2002 model [7], as formalized by MSC Software and released as part of their Adams CAR multibody simulation package, which includes a set of tools for obtaining the fitting coefficients from experimental datasets. The standard file format used for the storage of the tyre coefficients is text based and can also be interpreted by an appropriate MATLAB script[4]. The input variables of this tyre model are the slip ratio, slip angle, vertical force and camber angle. Outputs are the longitudinal and lateral forces and the self aligning torque.

2.3 Tyre Data

Experimental tyre data that is specific to Formula SAE and Formula Student tyres has been gathered by the FSAE Tyre Test Consortium at the Calspan test facility. The expensive tests are funded by the one-time fee the student teams pay to access the information. As access to experimental data was not possible during the preparation of this work, a sample tyre model was used. As the coefficients were given, the fitting process was not necessary. To verify that the tyre model was consistent with the physical intuitions developed so far, some plots were drawn (shown in figure 2.3), these both exhibit a plausible correlation between the slip quantities and the friction forces.

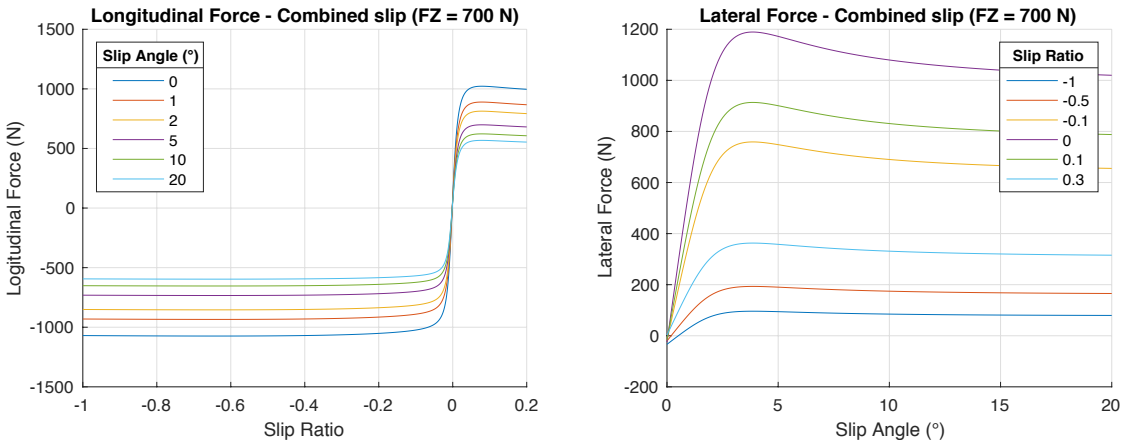

Figure 2.3: Longitudinal and Lateral forces in various combined slip conditions.

Chapter 3

Six degree-of-freedom vehicle model

3.1 6DoF Model structure

The state of the car is identified by the six degrees of freedom required to define the position and orientation of its chassis. The car interacts with the road through a simplified suspension system. The model outputs are the vertical forces on each of the four wheels and the velocities of the respective ground contact points, which may be used to calculate friction by an external tyre model. Inputs to the model are the longitudinal and lateral forces acting on each wheel and the steering angle. The latter effectively modifies the slip angle and slip ratio of the front wheels and rotates the consequent force vectors thereby allowing direction control. The block diagram in figure 3.1 shows the intended flow of information for this model working in conjuction with a tyre model.

3.2 Chassis description

The road is assumed to be a horizontal plane. The xyz inertial reference system is defined by z pointed downwards, in the direction of gravity, whilst x and y lie on the road surface and may be arbitrarily chosen as long as the right-handedness of the reference frame is guaranteed.

The chassis is represented by a rigid body whose mass m and rotational inertia matrix I resemble those of the entire vehicle, inclusive of the driver and the wheels.

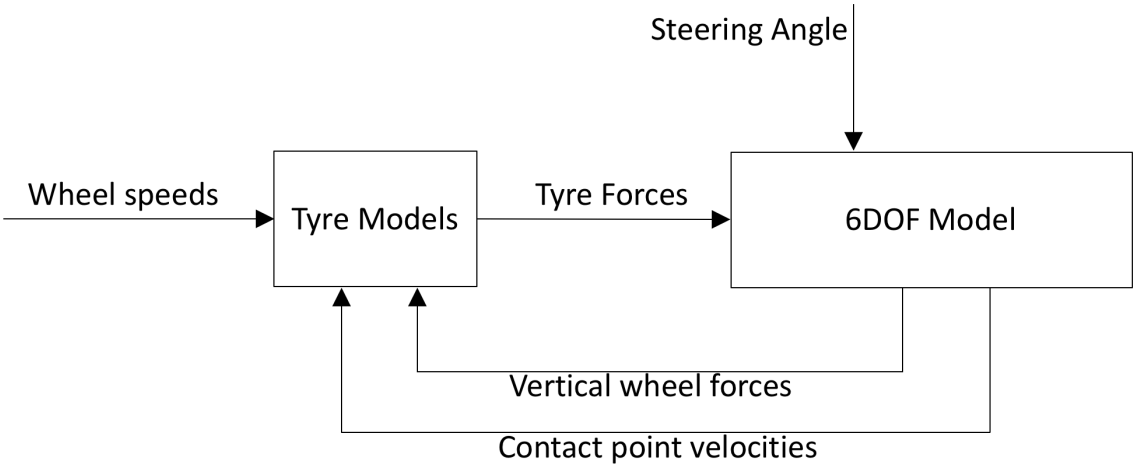

Figure 3.1: Information flow between the tyre model and the 6dof vehicle model.

The body reference system $x'y'z'$ is fixed to the chassis and originates at the center of mass. An unambiguous definition of the axes is made in the static equilibrium orientation when the suspension system is bearing the weight of the car and no other forces are at play (this will be later referenced as the *resting condition*). In this condition z' is defined to be parallel to z and likewise pointed downwards, x' is directed forwards, parallel to the road plane, and the y' axis is oriented to the right of the car as seen by the driver. The chassis reference system here defined is in line with the SAE J670[11] definition depicted in figure 3.2.

The attitude of the chassis is defined via the Tait-Bryan angles mapping the inertial system axes to the chassis axes: ψ (yaw), θ (pitch) and ϕ roll). Note that the previously given definition for the chassis reference system implies zero roll and pitch at static equilibrium.

An undercarriage reference system is also defined as having origin in the CoG projection to the road, its axes $x_u y_u z_u$ are obtained by applying only the yaw rotation to the inertial reference system.

The R_ψ and R matrices can be used to calculate inertial system vector co-ordinates from undercarriage system or vehicle system co-ordinates respectively.

$$R_\psi = \begin{bmatrix} cos(\psi) & -sin(\psi) & 0 \\ sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

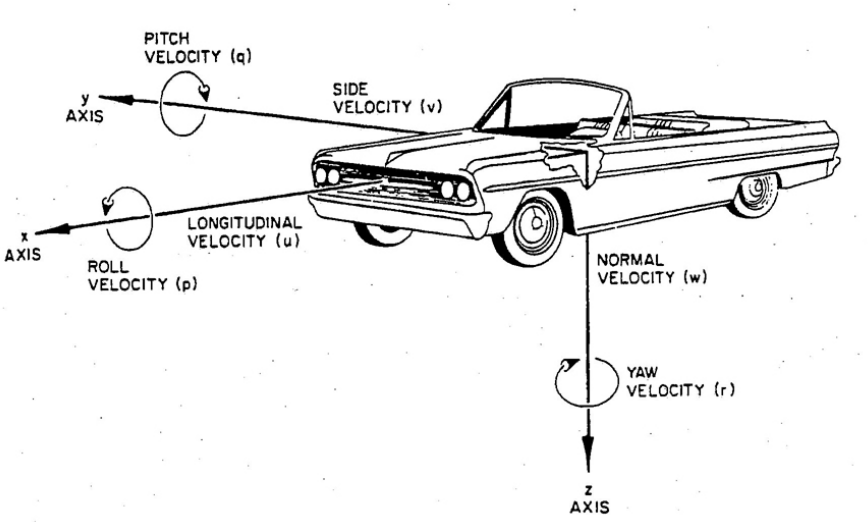

Figure 3.2: The chassis reference system as defined in the SAE J670 standard.

$$R = \begin{bmatrix} cos(\psi) & -sin(\psi) & 0 \\ sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} cos(\theta) & 0 & sin(\theta) \\ 0 & 1 & 0 \\ -sin(\theta) & 0 & cos(\theta) \end{bmatrix} . \begin{bmatrix} cos(\phi) & -sin(\phi) & 0 \\ sin(\phi) & cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} .$$

Geometric point coordinates can be brought into the inertial frame by rotating the point co-ordinates using the approriate rotation matrix and adding the original reference system origin co-ordinates.

The chassis is assumed to be symmetrical about the $x'z'$ plane, this identifies the y' axis as a principle axis of inertia and eliminates all the related product of inertia terms from the inertia matrix evaluated in the chassis reference system.

$$I = \begin{bmatrix} I_{xx} & 0 & I_{xz} \\ 0 & I_{yy} & 0 \\ I_{xz} & 0 & I_{zz} \end{bmatrix} .$$

Quantities relating to each of the four wheels or the correpsonding suspension systems will be distinguished by subscripts according to Table 3.1, the w subscript is used when referring generically to any one of them.

Table 3.1: Wheel and suspension systems subscripts

Subscript	Pertaining wheel or suspension system
FR	Front Right
FL	Front Left
RR	Rear Right
RL	Rear Left

3.3 Suspension

Suspension is represented by four vertical linear spring-damper systems associated with each wheel of the car.

The upper end of each suspension system is attached to an appropriate point P_w statically defined with respect to the chassis, the lower end point is obtained by projecting P_w onto the road plane. The spring length h_w is given by the distance between P_w and the road plane, so it is equal to the opposite of the inertial system z coordinate of P_w.

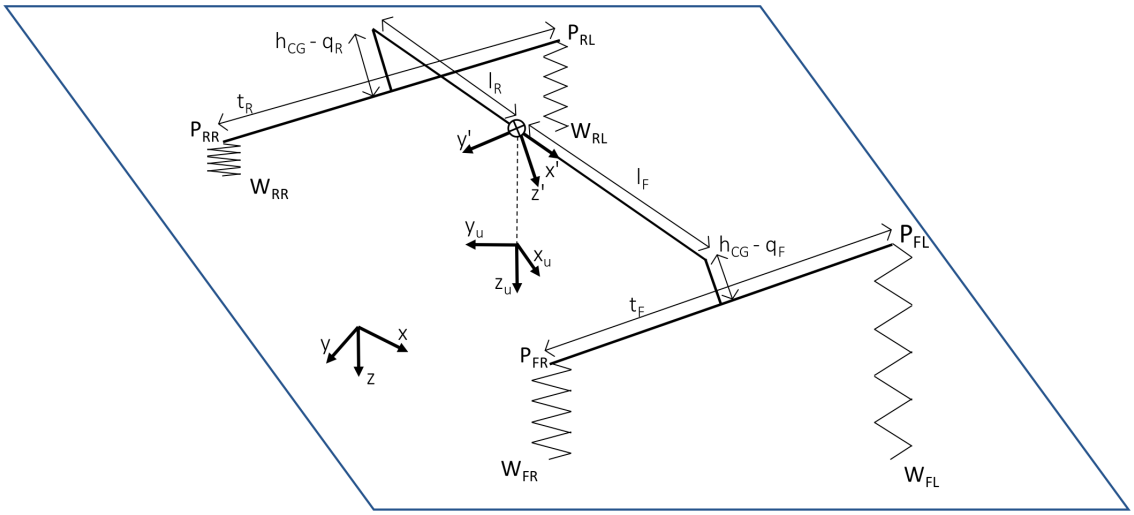

Figure 3.3: Description of the 6 DoF model geometry.

The correct weight transfer effects are obtained by assigning the corresponding wheel contact point x and y coordinates to each of the P_w points. These are easily calculated from the front and rear track lengths (t_F and t_R) and the distances of the front and rear axles from the CoG (l_F and l_R).

The z co-ordinates of the P_w points with respect to the chassis system are finally chosen in order to obtain realistic roll dynamics.

Automotive suspension systems can be characterized by a roll center for each axle. This is the point at which lateral forces acting on the wheels are reacted to the chassis. The roll center is usually below the center of mass, causing vehicles to lean towards the outside of turns when driving, this is in line with the situation shown in figure 3.4, where the lateral force at the center of mass is the centrifugal force acting in the non inertial reference frame of the car, balanced by friction acting on the wheel contact points.

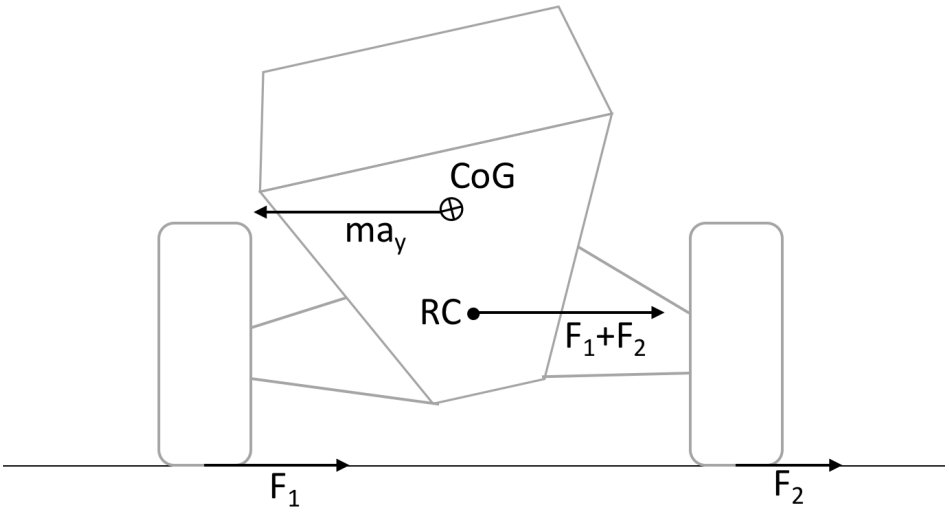

Figure 3.4: Body roll during cornering.

The roll center height, as defined by the SAE in J670, is determined by suspension kinematics and dynamic load distribution so it is almost never constant. In the developed model, each roll center is approximated by the midpoint of the left and right P_w points of the corresponding axle. Appendix A justifies this approximation.

The roll center heights (q_F and q_R) are a common design specification in race cars. Their values can be obtained by means of multibody simulation, experimental tests or kinematic approximations [5]. q_F and q_R are given with respect to ground in the resting condition even though they are relevant when considered in respect to the CoG height and tend to move with the chassis of the vehicle. When the model is in the resting condition the height of each of the P_w points is made to match the corresponding roll center height by completing their definition as in table 3.2.

In appendix A it is also shown that the linear spring rates for the model should be chosen as

$$k_F = \frac{\phi}{F} = 2\frac{h_{CG} - q_F}{K_{\phi F} t_F^2}.$$

Table 3.2: Coordinates of the suspension force application points

Chassis system co-ordinate	P_{FR}	P_{FL}	P_{RR}	P_{RL}
x	l_f	l_f	$-l_r$	$-l_r$
y	$\frac{t_f}{2}$	$-\frac{t_f}{2}$	$\frac{t_r}{2}$	$-\frac{t_r}{2}$
z	$h_{CG} - q_F$	$h_{CG} - q_F$	$h_{CG} - q_R$	$h_{CG} - q_R$

$$k_R = \frac{\phi}{F} = 2\frac{h_{CG} - q_R}{K_{\phi R} t_F^2}.$$

To make the model of the car sit in the resting position with zero pitch angle and the correct CoG height the unloaded spring lengths must be

$$h_{F0} = q_F + \frac{mg}{2k_F}\frac{l_R}{l_R + l_F}$$

$$h_{R0} = q_r + \frac{mg}{2k_R}\frac{l_F}{l_R + l_F}$$

3.4 Lagrangian formulation

Equations of motion were obtained by solving the lagrangian formulation of the system with respect to the inertial reference frame. The chosen lagrangian coordinates are, in order, the yaw, pitch and roll angles and the coordinates of the chassis CoG

$$q = \begin{bmatrix} \psi \\ \theta \\ \phi \\ x_{CG} \\ y_{CG} \\ z_{CG} \end{bmatrix}.$$

The gravitational potential energy is

$$U_{grav} = -mgz_{CG}$$

The potential energy stored in the suspension springs is

$$U_{spring} = \frac{1}{2}k_F[(h_{FR} - h_{F0})^2 + (h_{FL} - h_{F0})^2] + \frac{1}{2}k_R[(h_{RR} - h_{R0})^2 + (h_{RL} - h_{R0})^2]$$

The total potential energy function is

$$U = U_{spring} + U_{grav}$$

The damping effect of the shock absorbers is modelled by the Rayleigh dissipation function

$$D = \frac{1}{2}b_F(\dot{h}_{FR}^2 + \dot{h}_{FL}^2) + \frac{1}{2}b_R(\dot{h}_{RR}^2 + \dot{h}_{RL}^2)$$

b_F and b_r are the effective damping coefficients for vertical movements.

The translational kinetic energy of the car is given by

$$T_{trans} = \frac{1}{2}m(\dot{x}_{CG}^2 + \dot{y}_{CG}^2 + \dot{z}_{CG}^2)$$

E is the mapping between the Tait-Bryan angle derivatives and the angular velocity vector

$$E = \begin{bmatrix} 0 & -sin(\psi) & cos(\theta)cos(\psi) \\ 0 & cos(\psi) & cos(\theta)sin(\psi) \\ 1 & 0 & -sin(\theta) \end{bmatrix}.$$

The angular velocity vector with respect to the inertial frame is obtained from the roll, pitch and yaw rates as

$$\omega' = E \begin{bmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix}$$

Computing the kinetic energy requires that the angular velocity vector ω must be expressed with respect to the same axes as the inertia matrix, i.e. the chassis reference frame. This is easily done by

$$\omega = R^{-1}\omega'$$

The rotational kinetic energy is

$$T_{rot} = \frac{1}{2}\omega^T I \omega$$

The total kinetic energy function is then

$$T = T_{rot} + T_{trans}$$

The steering angle input δ acts like a time varying parameter in the model, influencing the wheel system rotation matrices. The tyre horizontal force component inputs F_wx F_wy are to be expressed with respect to each of the wheel reference systems.

To obtain the generalized forces vector Q the external forces expressed in vector form in the wheel frames are rotated into inertial frame co-ordinates and applied to the P_w points.

3.5 6DoF Dynamics equation set

The Lagrange Equations may be expressed in the form

$$M(\theta, \phi)\ddot{x} = f(q, \dot{q}) + Q(q, \delta)u$$

$$\begin{aligned}
M_{\psi,\psi} &= \mathrm{I_{zz}} \cos(\theta)^2 \cos(\phi)^2 + \mathrm{I_{yy}} \cos(\theta)^2 \sin(\phi)^2 - 2\,\mathrm{I_{xz}}\, \cos(\theta)\, \cos(\phi)\, \sin(\theta) + \mathrm{I_{xx}} \sin(\theta)^2 \\
M_{\psi,\theta} &= \sin(\phi)\, (\mathrm{I_{xz}}\, \sin(\theta) + \mathrm{I_{yy}}\, \cos(\theta)\, \cos(\phi) - \mathrm{I_{zz}}\, \cos(\theta)\, \cos(\phi)) \\
M_{\psi,\phi} &= \mathrm{I_{xz}}\, \cos(\theta)\, \cos(\phi) - \mathrm{I_{xx}}\, \sin(\theta) \\
M_{\theta,\theta} &= \mathrm{I_{yy}} - \mathrm{I_{yy}} \sin(\phi)^2 + \mathrm{I_{zz}} \sin(\phi)^2 \\
M_{\theta,\phi} &= -\mathrm{I_{xz}}\, \sin(\phi) \\
M_{\phi,\phi} &= \mathrm{I_{xx}}
\end{aligned}$$

$$M(\phi, \theta) = \begin{pmatrix}
M_{\psi,\psi} & M_{\psi,\theta} & M_{\psi,\phi} & 0 & 0 & 0 \\
M_{\psi,\theta} & M_{\theta,\theta} & M_{\theta,\phi} & 0 & 0 & 0 \\
M_{\psi,\phi} & M_{\theta,\phi} & M_{\phi,\phi} & 0 & 0 & 0 \\
0 & 0 & 0 & m & 0 & 0 \\
0 & 0 & 0 & 0 & m & 0 \\
0 & 0 & 0 & 0 & 0 & m
\end{pmatrix}$$

3.6 Output functions

The vertical wheel load outputs are calculated by the forces acting in the suspension systems, as a sum of the elastic and linear damping force.

$$Z_{FR} = -k_F(h_{FR} - h_{F0}) - b_F\dot{h}_{FR}$$

$$Z_{FL} = -k_F(h_{FL} - h_{F0}) - b_F\dot{h}_{FL}$$

$$Z_{RR} = -k_R(h_{RR} - h_{R0}) - b_R\dot{h}_{RR}$$

$$Z_{RL} = -k_R(h_{RL} - h_{R0}) - b_R\dot{h}_{RL}$$

The contact point velocities output is given by the time derivatives of the W_w points' co-ordinates in the inertial frame rotated into wheel system co-ordinates. The W_w point coordinates are defined with respect to the undercarriage reference frame in table 3.3.

Table 3.3: Coordinates of the tyre contact points

Undercarriage system co-ordinates	W_{FR}	W_{FL}	W_{RR}	W_{RL}
x	l_f	l_f	$-l_r$	$-l_r$
y	$\frac{t_f}{2}$	$-\frac{t_f}{2}$	$\frac{t_r}{2}$	$-\frac{t_r}{2}$
z	0	0	0	0

Chapter 4

The twelve degree-of-freedom Model

4.1 12DoF Model structure

The model in the previous chapter is extended by integrating wheel and steering dynamics. The additional degrees of freedom are given by wheel angular positions and (front and rear) steering angles.

Note that the additional degrees of freedom still do not allow the wheels to tilt. This means that the wheel angular velocity vectors are constrained to a horizontal plane and gyroscopic effects do not emerge.

The steering angle can no longer be an input as it is part of the vehicle state. Direction control can then only be obtained by applying steering forces. Inputs for tyre self aligning moments are added to model the steering resistance. Motors are modelled by inputs corresponding to the torques they generate at each wheel axle. Braking is represented by negative motor torque. As the wheel angular velocities are included, the slip quantities are calculated within the model. The inputs and outputs are those seen in figure 4.1.

4.2 Unsprung mass

During the development of this second model the unsprung part of the vehicle was modelled to improve accuracy of the roll dynamics. It's mass m_u was considered separately from that of the body. The unsprung mass does not move vertically so it is not accounted for

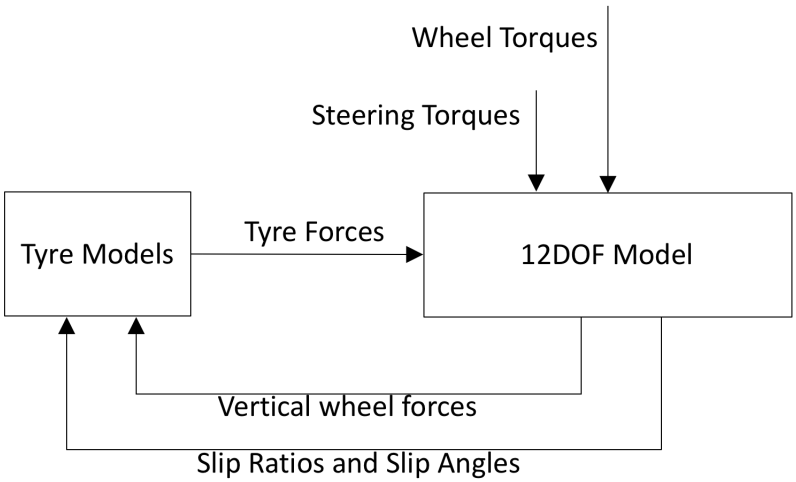

Figure 4.1: Intended Information flow for the 12 DoF Model.

in the gravitational potential energy. The associated vertical kinetic energy is also zero.

The rotational inertia I_u of the unsprung mass is also separated. The latter is a scalar because the unsprung mass only rotates around the yaw axis and does not pitch or roll. The unsprung components weight does not act through the suspension by definition so it's contribution to the vertical wheel forces must be added separately. Assuming it is distributed equally over all four wheels the correct wheel loads are simply

$$Z'_w = Z_w + \frac{m_u}{4} g.$$

4.3 Wheels

The wheels are represented by their angular positions. The input longitudinal force on each tyre generates a moment on the wheel axis. This is summed to the motor torques and assigned to the generalized forces component associated with the corresponding wheel angular position.

Whenever the wheels are driven by the motors a reaction torque between each wheel and the chassis is generated. Each of the motor torques is expressed in vector form with respect to the inertial reference frame.

The torques are then vectorially summed. The effect of this resulting torque on the lagrangian coordinates is given by left-multiplying it with the rotational Jacobian Matrix of the chassis. The rotational Jacobian is equal to the inverse of the E matrix shown in

chapter 3.

4.4 Steering

The wheels on each axle are assumed to be parallel to each other. The front and rear steering angles, δ_F and δ_R, are defined as the angles the front and rear wheel planes form with the xz plane of the undercarriage reference system. Steering dynamics are characterized by the steering system inertia used to calculate the associated kinetic energy term in the Lagrangian. The steering system inertias, I_F and I_R, may be approximated by the rotational inertia of the wheel and wheel hub systems about the kingpin axes (assumed as vertical).

The steering torque inputs are summed to the self aligning tyre moment inputs and assigned to the generalized forces components associated with the corresponding steering angles.

The steering reaction to the chassis is modelled by adding the sum of the steering torques to the generalized forces component associated with the yaw angle. The rotational Jacobian can be bypassed as the steering torques act parallel to the undercarriage reference frame z axis which is always coincident with yaw axis.

Chapter 5

Model implementation workflow

This chapter shows the process behind the implementation of the 12 DoF Model. The 6 DoF model can be obtained by removing the appropriate degrees of freedom and related features. To make the developed model easy to handle as part of more complex projects it was implemented in the form of a simulink library. The library comprises two blocks, a tyre block and a Body block as in Figure 5.1.

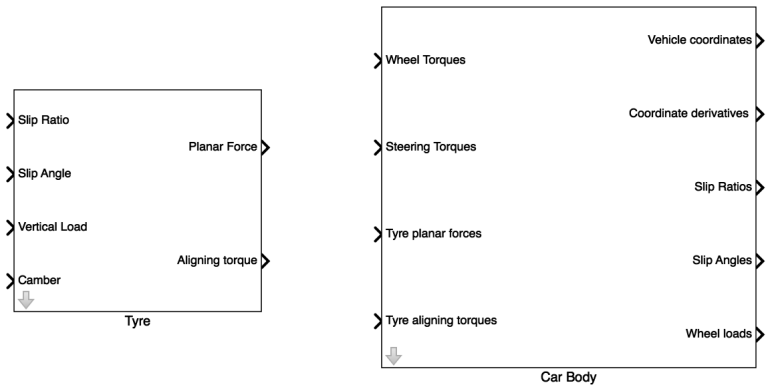

Figure 5.1: Simulink blocks containing the tyre model and the 12DoF model.

5.1 Tyre block

The Tyre model is implemented as a MATLAB function. The inputs are the slip ratio, the slip angle, the vertical force and the camber angle. The outputs are the horizontal force 2D vector (F_x, F_y) and the aligning torque. If all the tyres of the car are the same a single

tyre block can be used to model all four wheels of the car, this is done by concantenating the inputs horizontally into a matrix. The forces corresponding to each tyre can then be extracted as columns of the outputs. The outputs are computed by evaluating the Magic Formula equations described in [7]. The coefficients representing the tyre for the simulations must be given in the Adams Car ".TIR" format. The format is human readable as a plaintext file, as described in [7]. The tire model file must be located in the Matlab working directory. The name of the file can be entered in the "block mask" accessible by double clicking the block (Figure 5.2).

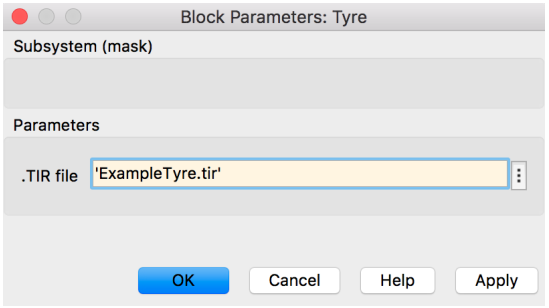

Figure 5.2: Simulink mask for the tyre block allows selection of the tyre model.

The parameters are loaded from the file into the tyre block workspace during the initialization phase by means of a script which takes advantage of the "loadTIR" function[4].

Note that tyres are not always simmetrical. The model used in these simulations reflects this property and yields non-zero lateral force at zero slip angle. For this reason care should be taken when modelling tyres on opposite sides of the car. The ".TIR" files contain a field specifying for which side the empirical coefficients were fitted. To model tyres on the other side all inputs and outputs can be redefined in a simmetrical reference system. This simply means the slip angle and camber angle signs have to be changed before entering the tyre model and the lateral force and aligning moment signs have to be changed after they are output.

5.2 Vehicle Body Block

The internal block diagram for the 12 DoF Model is shown in figure 5.3. The *dynamics* block employs the state space representation of the system to evaluate the state derivative $\dot{x}$ from the state and the inputs. The state derivative, $\dot{x}$, is then integrated to close the physical feedback loop. The expressions used by the *dynamics* block are loaded from

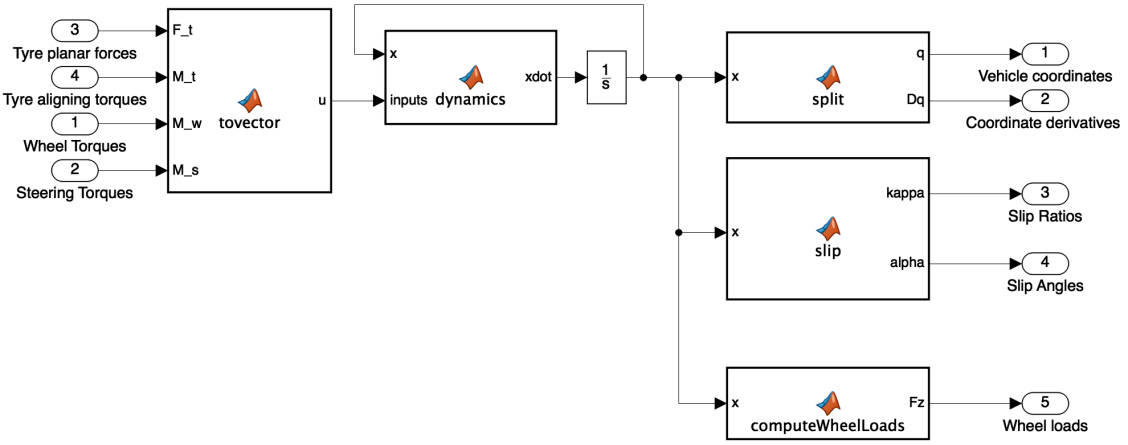

Figure 5.3: Simulink block diagram for the 12DoF model.

a MATLAB function file created by the *carmodel.m* script (Appendix C.2) discussed in section 5.3. The *tovector* block simply assembles the incoming signals into the input vector for the dynamic system. The *split* block disassembles the state vector x into the lagragian coordinates and their derivatives. The *slip* block takes the vehicle coordinates as an input, calculates the contact point velocities and returns the slip angles and slip ratios. The *computeWheelLoads* block evaluates the vertical wheel forces using the formulas in section 3.6.

5.2.1 Parameters and initial state

The initial state for the integrator and the vehicle body parameters are taken from text files. The text files are in the format shown in figures B.1 and B.2. The file names are entered in the block mask as seen in Appendix B.

5.3 Mathematical model implementation script

Appendix C.2 contains the full MATLAB script for the construction of the Lagrange equations of the 12DoF model and their solution. The script is based on the symbolic math toolbox.

The script begins by defining symbols for vehicle parameters and inputs. Lagrangian coordinates and the necessary derivatives are declared as symbolic functions of the time variable `t`. Several groupings of these variables are made into one dimensional vectors. An

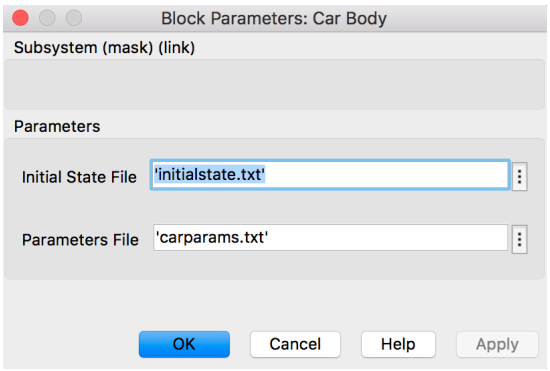

Figure 5.4: Simulink mask for the body dynamics block.

example is given by `q` and `Dq` which respectively list symbols for the lagrangian coordinates and their first order derivatives.

The rotation matrices and CoG coordinates vector are defined to help in the coordinate transformations between inertial, undercarriage and chassis reference systems. Matrices are also defined to rotate wheel forces and velocities from the wheel frames to the undercarriage frame.

Geometric points, velocities, forces and other three dimensional physical entities are expressed as column vectors of cartesian coordinates with respect to some reference system, explicited in the source code comments.

Lengthy code repetitions were avoided by taking advantage of the properties of matrix multiplication. When dealing with the same quantity for each wheel of the car, such as the suspension travels or contact points, variables are grouped as columns of matrices. The order used is `FR - FL - RR - RL`. The following code snippet is provided as an example.

```
118 % spring to frame attachment point coordinates wrt body frame
119 P =[l_f       l_f      -l_r      -l_r
120     t_f/2    -t_f/2    t_r/2    -t_r/2
121     h_CG-q_f h_CG-q_f h_CG-q_r h_CG-q_r];
122 % Suspension attachment point coordinates wrt inertial frame
123 ps = p_CG + R*P;
```

Here `p_CG` is the CoG coordinates vector with respect to the inertial reference frame and `R` is the active rotation matrix from chassis to body frame.

When using the `diff()` function to differentiate the lagrangian coordinates with respect to time, the symbolic math toolbox uses place-holders such as `diff(x_CG,t)`. These

expressions are replaced by the script with the specifically defined symbols for the coordinate derivatives.

The `subs(expr, diff(q,t), Dq)` instruction is used to re-format the `expr` function containing time derivatives as it efficiently handles all the place-holders at once.

The function listed in appendix C.1 uses the virtual work principle[3] to find the generalized forces, Q_i for the lagrangian formulation of the system.

The kinetic and potential energies, T and U, are calculated and kept separate. The Langrangian is defined as

$$L = T - U$$

The Lagrange equations are given by evaluating

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} + \frac{\partial D}{\partial \dot{q}_i} = Q_i$$

where q_i takes the values of each of the lagrangian coordinates. These equations may be written in the form

$$\frac{d}{dt}\frac{\partial T}{\partial \dot{q}_i} = \frac{d}{dt}\frac{\partial U}{\partial \dot{q}_i} + \frac{\partial T}{\partial q_i} - \frac{\partial U}{\partial q_i} - \frac{\partial D}{\partial \dot{q}_i} + Q_i$$

Second order derivatives of the lagrangian coordinates can only occur in the left terms. Placeholders are initially used for the right hand sides. Solving the resulting linear system of equations to find the second order derivatives is done using the `solve()` function. The right hand sides of the equations are then evaluated and substituted to the placeholders in the solutions.

The lagrangian coordinates and their first order derivatives constitute the vehicle state. The state space representation is assembled accordingly

The `matlabFunction()` command is used to save the system laws of motion to a file, to be executed by the simulink blocks. The same is done for the outputs of the model.

5.4 The simulation configuration

To obtain interesting data from a vehicle dynamics simulation the car model needs to receive a realistic input. Inputs for testing the discussed model were generated in different ways.

The first tests were done without the intervention of the tyre model. Constant forces or torques were directly applied to the model inputs to verify the expected behaviour.

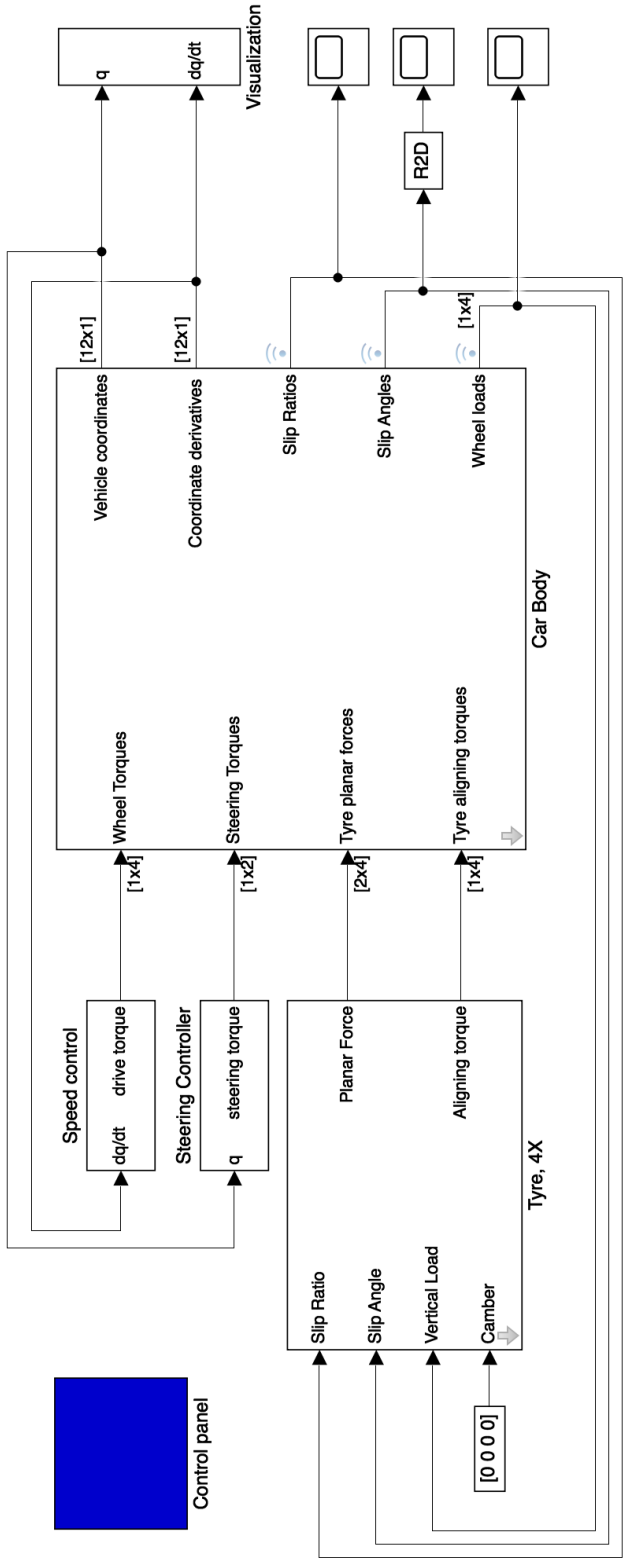

Figure 5.5: Block diagram with steering and speed controllers for test simulations.

After debugging the vehicle body model using this method the tyre model was connected, the resulting block diagram for the 12 DoF model is as shown in figure 5.5.

The steering and speed controllers may be implemented in various ways to obtain the desired test conditions.

The *visualization* subsystem includes several *scope* blocks to plot interesting data such as the vehicle state, lateral accelerations, velocities, slip angles and more. The *visualization* subsystem also contains a 2D animation feature adapted from [2], shown in figure 5.6.

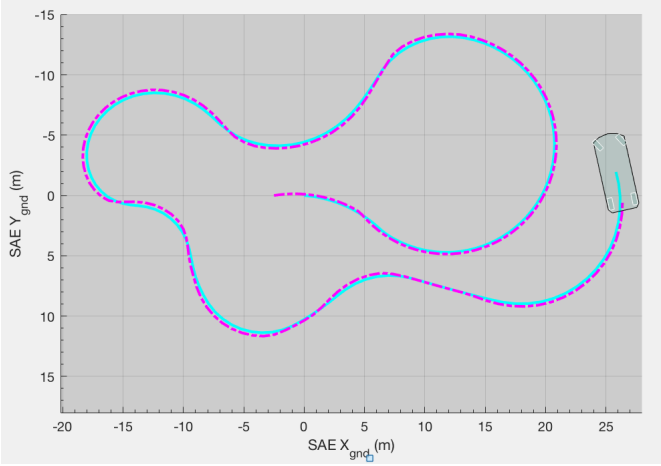

Figure 5.6: Simulink real time interface.

5.4.1 Sinusoidal Steering Input

A sinusoidal steering angle input was used to observe transients in the 6DoF vehicle model. Data resulting from this test is shown in figure 5.7. Various phase delays can be observed between the most important physical quantities involved in handling dynamics, including load transfer and side slip angle. The side slip angle is defined as the angle between the undercarriage x axis and the vehicle horizontal velocity $v_u = (\dot{x}_{CG}, \dot{y}_{CG}, 0)$. The delay between the roll angle and lateral acceleration is comparable to real experimental data captured during a slalom test, shown in figure 5.8. The test was run using the Peacock3 FSAE car built by Polimarche Racing Team.

5.4.2 Manual steering and acceleration

Simulink offers the possiblity to slow down simulations to real-time speed and graphical user interface block to modify simulation parameters during execution. These two features

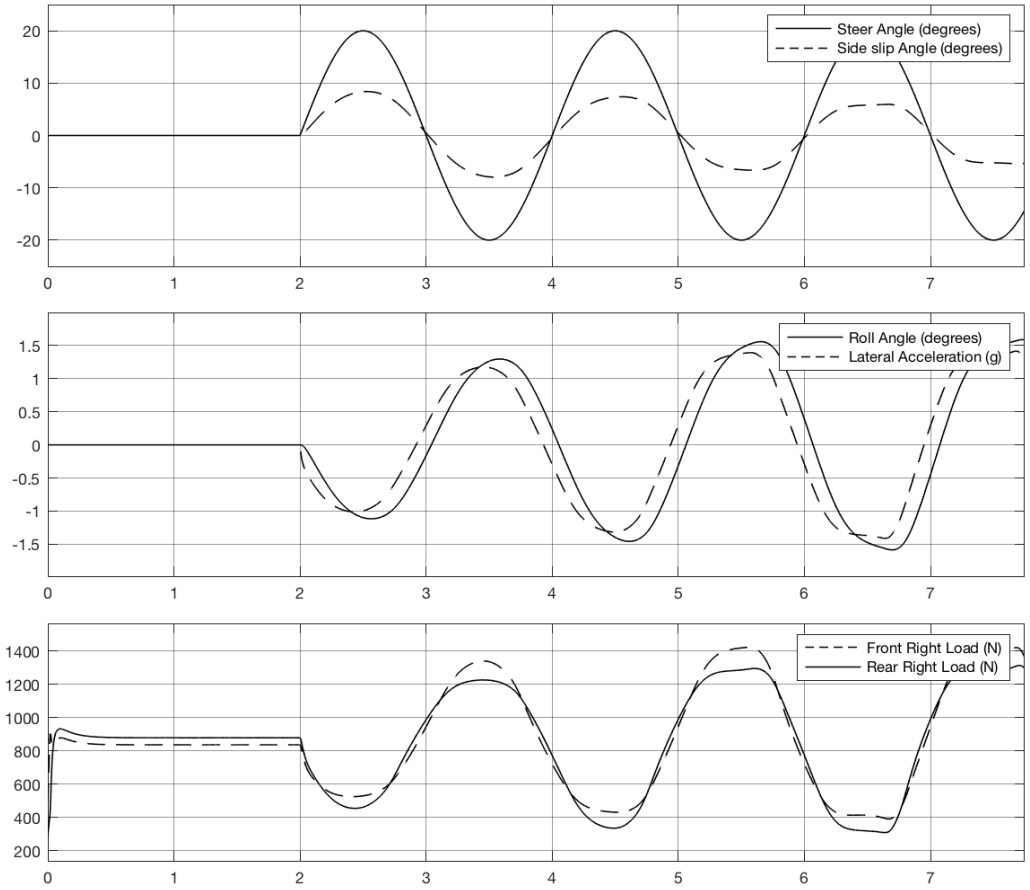

Figure 5.7: Vehicle response for 0.5 Hz sinusoidal steering input.

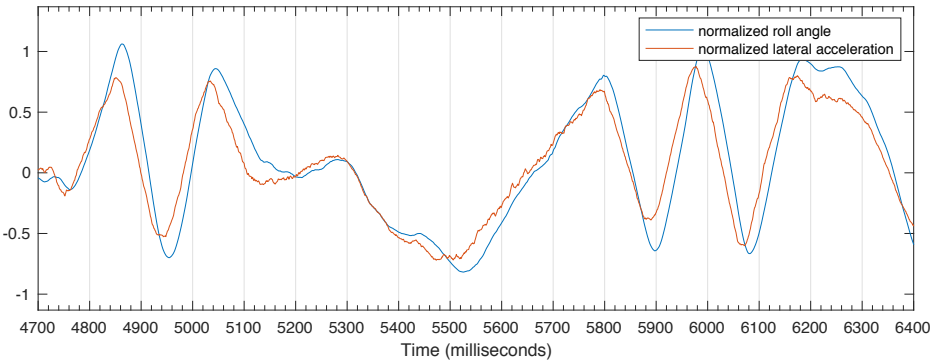

Figure 5.8: Roll angle and lateral acceleration during slalom tests. The vehicle was driven twice through the same 5 cone slalom, completing a U-turn in between. [Data courtesy of Polimarche Racing Team]

in combination with the 2D animation shown in figure 5.6 made it possible to manually drive the simulink model using the computer cursor, like a rudimentary videogame, whilst saving simulation data. The "driver" interface is shown in figure 5.9.

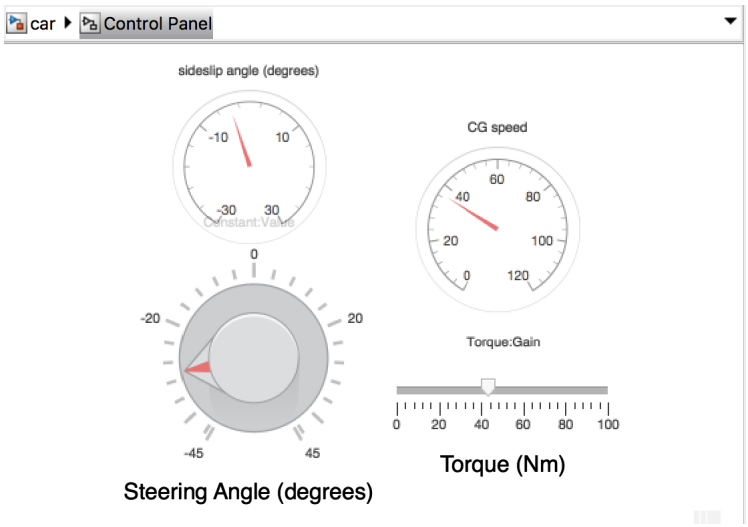

Figure 5.9: Simulink real time interface.

5.4.3 Skidpad test

A skidpad test consists in driving the examined vehicle in a circle with constant radius and at constant speed. This kind of test is easy to reproduce experimentally and is thus useful when validating vehicle dynamics models.

Maintaining constant speed is done by using a PID controller whose output is the torque equally applied to the rear wheels.

Following a circular trajectory can be seen as holding a constant distance from the origin of the inertial reference system. This was accomplished by using two controllers.

The first PID controller acts in a servo loop to hold a target steering angle. Note that this controller is not needed in the 6DoF model as the steering angle does not have dynamics of its own.

The second PID controller generates the target steering angle to hold the desired distance from the reference point.

Chapter 6

Conclusion

6.1 Future Work

6.1.1 Aerodynamic effects

6.1.2 Driving algorithm

Appendix A

Roll center approximation

The SAE definition of roll center is

"The point in the transverse vertical plane through any pair of wheel centers at which lateral forces may be applied to the sprung mass without producing suspension roll"

This definition directly relates the P_w points to the roll center height. Validity for chosen roll center approximation can be shown by modifying a generic equilibrium situation such as the one described in figure [TODO:la stessa di prima] and applying a new side force F' acting on the midpoint M_F of P_{FR} and P_{FL}, if the tyres are capable of generating the necessary friction forces (F_1 and F_2), the system will settle in a new equilibrium state. The sum of vertical forces acting on the tyres does not change as it is equal to

$$F_z := F_{z1} + F_{z2} = mg.$$

As F_z does not change, the increase in side friction force is entirely due to changes in the lateral friction coefficients, which are both supposed as being equal to μ_y.

The fact that the spring systems are constrained in the vertical orientation means the horizontal forces acting at the ground contact points effect the chassis directly at the points P_{FR} and P_{FL} respectively.

The horizontal force balance is given by

$$F + F' = F_1 + F_2 = \mu_y(F_{z1} + F_{z2}) = \mu_y mg$$

To find the equilibrium roll angle the torque balance about point M_F must be solved

$$\frac{t_F}{2}\sin\phi(F_1 - F_2) - \frac{t_F}{2}\cos\phi(F_{z1} - F_{z2}) - F(h_{CG} - q_F)\cos\phi = 0$$

substituting the friction forces yields

$$\frac{t_F}{2}(F_{z1} - F_{z2})[\mu_y \sin\phi - \cos\phi] - F(h_{CG} - q_F)\cos\phi = 0$$

The vertical forces act through the springs (spring rate k_F) . As Hooke's law is linear, the difference between them is proportional to the difference between the spring lengths which is in turn obtained as a function of the roll angle.

$$F_{z1} - F_{z2} = -k_F t_F \sin\phi$$

The moment balance then becomes

$$\mu_y k_F \frac{t_F^2}{2}\sin^2\phi - k_F \frac{t_F^2}{2}\sin\phi\cos\phi - F(h_{CG} - q_F)\cos\phi = 0$$

[TODO:find source for roll angles in passenger cars being a few ° max] The equation may be linearized with respect to ϕ.

$$-k_F \frac{t_F^2}{2}\phi - F(h_{CG} - q_F) = 0$$

Doing so makes the solution for ϕ independent of μ_y, the only quantity influenced by F'. The height of point M_F is then effectively the roll center height according to the SAE definition.

The roll rate K_ϕ is defined as roll angle per unit of lateral force applied to the CG [TODO:find source], applying this definition to the linearized equation gives the expression to be used to derive spring rates for the 6DoF model in order to match the roll-rate of the car.

$$K_\phi = \frac{\phi}{F} = 2\frac{h_{CG} - q_F}{K_F t_F^2}$$

Appendix B

Model Parameters files

```
# car parameters file, variable names must match those used in carmodel.m
# all units are metric
g          9.81       # gravity
t_f        1.2        # front track
t_r        1.2        # rear track
l_f        0.8        # Front axle to CG distance
l_r        0.8        # Rear axle to CG distance
q_f        0.09       # front roll center height
q_r        0.09       # rear roll center height
h_CG       0.3        # CG height
I_f        1          # Front Steering inertia
I_r        1          # Rear steering Inertia
k_f        35000      # front virtual spring stiffness
k_r        35000      # rear virtual spring stiffness
b_f        2000       # front virtual suspension damping
b_r        2000       # rear virtual suspension damping
m          300        # sprung mass (including driver)
m_u        50         # unsprung mass
Ixx        30         # Moment of inertia about longitudinal axis (sprung only)
Iyy        60         # Moment of inertia about lateral axis (sprung only)
Izz        50         # Moment of inertia about vertical axis (sprung only)
Ixz        0          # vertical-longitudinal inertia coupling
I_u        50         # unsprung mass rotational inertia about vertical axis
r_0        0.26       # wheel radius
I_w        0.3        # wheel rotational inertia
```

Figure B.1: Structure for the vehicle parameters file

```
# initial state file, variable names must match those used in carmodel.m
# all units are metric SI
y         0        # initial yaw
p         0        # initial pitch
r         0        # initial roll
x_CG      0        # initial CG coordinates
y_CG      0        # initial CG coordinates
z_CG     -0.3      # initial CG coordinates
delta_f   0        # initial front steer angle
delta_r   0        # initial rear steer angle
gamma_fr  0        # initial wheel angular position (FR)
gamma_fl  0        # initial wheel angular position (FL)
gamma_rr  0        # initial wheel angular position (RR)
gamma_rl  0        # initial wheel angular position (RL)
Dy        0        # initial yaw rate
Dp        0        # initial pitch rate
Dr        0        # initial roll rate
Dx_CG     1        # initial CG velocity wrt inertial frame
Dy_CG     0        # initial CG velocity wrt inertial frame
Dz_CG     0        # initial CG velocity wrt inertial frame
Ddelta_f  0        # initial front steering rate
Ddelta_r  0        # initial rear steering rate
Dgamma_fr 0        # initial wheel speed (FR)
Dgamma_fl 0        # initial wheel speed (FL)
Dgamma_rr 0        # initial wheel speed (RR)
Dgamma_rl 0        # initial wheel speed (RL)
```

Figure B.2: Structure for the initial vehicle state file

Appendix C

Matlab code

C.1 Generalized forces function

```
% apply virtual work principle to find effect of external forces on
% langrangian coordinates
function Q = genforces(f,p,q)
% number of DOFs
n = length(q);
% number of external forces
m = length(f);

Q = sym(zeros(n,1));

for j = 1:m
    Q = Q + f(j) * functionalDerivative(p(j), q);
end
end
```

C.2 12DoF Model generation script

```
function [] = carmodel()
%% Car parametrization
disp 'defining car parameters'
% gravitational acceleration
syms g
```

```
6  % track widths
7  syms t_f t_r
8  % Front / Rear Axle Distance from CG
9  syms l_f l_r
10 % Front / Rear Roll center height
11 syms q_f q_r
12 % CG ride height
13 syms h_CG
14 % Front / Rear Steer Inertia
15 syms I_f I_r
16 % Front / Rear virtual Spring coefficients
17 syms k_f k_r
18 % Front / Rear virtual Damper coefficients
19 syms b_f b_r
20 % sprung mass
21 syms m
22 % unsprung mass (including wheels)
23 syms m_u
24 % sprung masses moment of inertia tensor wrt body frame
25 syms Ixx Iyy Izz Ixz
26 % unsprung masses moment of inertia about vertical axis through CG
27 syms I_u
28 % Nominal Wheel Radius
29 syms r_0
30 % Wheel rotational inertia
31 syms I_w
32
33
34 %% Inputs Definition
35 disp 'defining car input signals'
36 % Front / Rear steer torques
37 syms M_sf M_sr
38 % planar wheel forces
39 syms FX_fr FY_fr FX_fl FY_fl FX_rr FY_rr FX_rl FY_rl
40 % self aligning wheel torques
41 syms MZ_fr MZ_fl MZ_rr MZ_rl
42 % motor/brake torque
43 syms MW_fr MW_fl MW_rr MW_rl
```

```
44
45
46 %% TIME
47 syms t
48 %% CAR STATE COORDINATES
49 disp 'defining car coordinates'
50 % body orientation angles & derivatives (ZYX Euler - YPR - Tait Bryan)
51 syms   y(t)  p(t)  r(t)
52 syms  Dy(t) Dp(t) Dr(t)
53 syms DDy   DDp   DDr
54 % CG position wrt inertial frame & derivatives
55 syms   x_CG(t)  y_CG(t)  z_CG(t)
56 syms  Dx_CG(t) Dy_CG(t) Dz_CG(t)
57 syms DDx_CG   DDy_CG   DDz_CG
58 % front / rear steer angles
59 syms   delta_f(t)   delta_r(t)
60 syms  Ddelta_f(t)  Ddelta_r(t)
61 syms DDdelta_f    DDdelta_r
62 % wheel rotation angles
63 syms   gamma_fr(t)   gamma_fl(t)   gamma_rr(t)   gamma_rl(t)
64 syms  Dgamma_fr(t)  Dgamma_fl(t)  Dgamma_rr(t)  Dgamma_rl(t)
65 syms DDgamma_fr    DDgamma_fl    DDgamma_rr    DDgamma_rl
66
67 Dgamma = [Dgamma_fr Dgamma_fl Dgamma_rr Dgamma_rl];
68 % various handy vector combinations of the above
69 q   = [  y;   p;   r;   x_CG;   y_CG;   z_CG; delta_f; delta_r; ...
70      gamma_fr; gamma_fl; gamma_rr; gamma_rl];
71 Dq  = [ Dy;  Dp;  Dr;  Dx_CG;  Dy_CG;  Dz_CG; Ddelta_f; Ddelta_r; ...
72      Dgamma_fr; Dgamma_fl; Dgamma_rr; Dgamma_rl];
73 DDq = [DDy; DDp; DDr; DDx_CG; DDy_CG; DDz_CG; DDdelta_f; DDdelta_r; ...
74          DDgamma_fr; DDgamma_fl; DDgamma_rr; DDgamma_rl];
75
76 % position of CG wrt inertial frame
77 p_CG = [x_CG; y_CG; z_CG];
78 %% ROTATION MATRICES
79 disp 'defining matrices'
80
81 % yaw rotation matrix (ROTATION MATRIX from Inertial to undercarriage)
```

```
82 Rz=[cos(y) -sin(y) 0
83     sin(y) cos(y)  0
84     0      0       1];
85
86 % pitch rotation matrix
87 Ry=[cos(p)  0  sin(p)
88     0       1  0
89     -sin(p) 0  cos(p)];
90 % roll rotation matrix
91 Rx=[1  0      0
92     0  cos(r) -sin(r)
93     0  sin(r) cos(r) ];
94
95 % Full rotation matrix from Inertial frame to body frame
96 % this matrix carries all information about vehicle attitude,
97 % it can be used to rotate vectors in body to inertial frame
98 R = Rz*Ry*Rx ;
99
100 %% Steering Rotation Matrices
101 % from undercarriage frame to wheel frames (no Ackermann)
102 Sfr=[cos(delta_f) -sin(delta_f) 0
103     sin(delta_f)  cos(delta_f)  0
104     0             0             1 ];
105 Sfl=[cos(delta_f) -sin(delta_f) 0
106     sin(delta_f)  cos(delta_f)  0
107     0             0             1 ];
108
109 Srr=[cos(delta_r) -sin(delta_r) 0
110     sin(delta_r)  cos(delta_r)  0
111     0             0             1 ];
112 Srl=[cos(delta_r) -sin(delta_r) 0
113     sin(delta_r)  cos(delta_r)  0
114     0             0             1 ];
115
116 %% SUSPENSION
117 disp 'defining suspension'
118 % spring to frame attachment point coordinates wrt body frame
119 P =[l_f      l_f     -l_r      -l_r
```

```
120       t_f/2      -t_f/2     t_r/2      -t_r/2
121       h_CG-q_f h_CG-q_f h_CG-q_r h_CG-q_r];
122  % Suspension attachment point coordinates wrt inertial frame
123  ps = p_CG + R*P;
124  % suspension displacement
125  d = -[0 0 1]*ps - [h_f h_f h_r h_r];
126  % suspension velocities
127  Dd = subs(diff(d,t), diff(q,t), Dq);
128
129  %% ENERGY
130  disp 'defining energies'
131
132  % suspension damper-dissipated energy (Rayleigh dissipation function)
133  D         = 1/2 * Dd.^2 * [b_f b_f b_r b_r].';
134
135  % energy stored in suspension springs
136  U_spring = 1/2 *  d.^2 * [k_f k_f k_r k_r].';
137
138  % gravitational potential energy
139  U_g = -m*g*z_CG;
140
141  % Total potential energy
142  U = U_spring + U_g;
143
144  % translational kinetic energy
145  T_trans = 1/2 * m    * (Dx_CG.^2 + Dy_CG.^2 + Dz_CG.^2)  ...
146           + 1/2 * m_u * (Dx_CG.^2 + Dy_CG.^2);
147
148  E=[ 0 -sin(y)  cos(p)*cos(y) ;
149       0  cos(y)  cos(p)*sin(y) ;
150       1        0         -sin(p) ];
151
152  % Rotational velocity vector wrt inertial frame
153  w = E * [Dy(t); Dp(t); Dr(t)];
154
155  % Rotational velocity vector wrt body frame
156  W = R\w;
157
```

```
158 % Inertia Tensor
159 I=[ Ixx 0   Ixz;
160      0   Iyy 0;
161      Ixz 0   Izz];
162
163 % rotational kinetic energy of body
164 T_rot = 1/2*W.'*I*W + 1/2*I_u*Dy.^2;
165 % rotational kinetic energy of wheels
166 T_w = 1/2 * I_w * sum(Dgamma.^2);
167 % steering kinetic energy
168 T_steer = 1/2 * I_f * (Ddelta_f.^2) + 1/2 * I_r * (Ddelta_r.^2);
169
170 % total kinetic energy
171 T = T_rot + T_trans + T_w + T_steer;
172 %% WHEELS
173 disp 'defining road interface'
174 % contact point coordinates with respect to undercarriage reference
      frame
175 CP =[l_f       l_f      -l_r       -l_r
176      t_f/2    -t_f/2    t_r/2     -t_r/2
177      0          0         0          0 ];
178 % contact point coordinates with respect to inertial frame
179 cp = Rz*CP + [x_CG; y_CG; 0];
180 % wheel contact point velocities wrt inertial frame
181 v = subs(diff(cp, t), diff(q,t), Dq);
182 v = v(t);
183 % contact point velocities wrt corresponding wheel frames
184 v_w = [ Sfr\(Rz\v(:,1)) Sfl\(Rz\v(:,2)) Srr\(Rz\v(:,3))
      Srl\(Rz\v(:,4)) ];
185 % total planar force wrt inertial frame
186 f= [Rz*Sfr*[FX_fr; FY_fr; 0] ...
187      Rz*Sfl*[FX_fl; FY_fl; 0] ...
188      Rz*Srr*[FX_rr; FY_rr; 0] ...
189      Rz*Srl*[FX_rl; FY_rl; 0] ];
190 f=simplify(f(t));
191 ps=ps(t);
192 %% TORQUES ON BODY
193 disp 'defining torques'
```

```
194
195 % total motors/brakes reaction torques
196 M_motor = Rz*Sfr*[0; MW_fr; 0] + ...
197           Rz*Sfl*[0; MW_fl; 0] + ...
198           Rz*Srr*[0; MW_rr; 0] + ...
199           Rz*Srl*[0; MW_rl; 0] ;
200 M_motor = simplify(M_motor);
201 chi = [y(t);p(t);r(t)];
202 M_body = E \ (M_motor);
203 %% TORQUES ON WHEELS
204 M_w = [MW_fr; MW_fl; MW_rr; MW_rl] - r_0 * [FX_fr; FX_fl; FX_rr;
        FX_rl];
205 P_w = [gamma_fr(t); gamma_fl(t); gamma_rr(t); gamma_rl(t)];
206 %% TORQUE ON STEERING
207 M_f = M_sf + MZ_fr + MZ_fl;
208 M_r = M_sr + MZ_rr + MZ_rl;
209 %% LAGRANGE
210 disp 'calculating lagrange magic'
211 % all external forces/torques
212 F_ext = [f(:); M_body(t); M_w;          M_f;          M_r;   M_sf+M_sr];
213 % correpsonding application points/angles
214 P_ext = [ps(:);       chi; P_w; delta_f(t); delta_r(t);        y(t)];
215 % generalized forces
216 Q = genforces(F_ext, P_ext, q(t));
217
218 % Potential energy is independent of coordinate derivatives, so second
219 % order time derivatives will only appear in the kinetic term
220 LHS = diff(functionalDerivative(T, Dq),t);
221 % improve speed by replacing time derivatives with correpsonding
        symbols
222 LHS = simplify(subs(LHS, [diff(q); diff(Dq)], [Dq; DDq]));
223 % X1 ... X12 are placeholders for all other lagrange-equation terms
        (which
224 % do not contain second order derivatives)
225 X   = sym('X', [12, 1]);
226 % Solve for second order derivatives (vector space representation)
227 qdotdot = solve(LHS == X, DDq);
228 % convert struct to array
```

```
qdotdot = [qdotdot.DDy;      qdotdot.DDp;      qdotdot.DDr;
           qdotdot.DDx_CG;  qdotdot.DDy_CG;  qdotdot.DDz_CG;
           qdotdot.DDdelta_f;  qdotdot.DDdelta_r;
           qdotdot.DDgamma_fr;  qdotdot.DDgamma_fl;
           qdotdot.DDgamma_rr;  qdotdot.DDgamma_rl              ];
qdotdot = simplify(qdotdot);
% calculate values for placeholders
RHS = subs(diff(functionalDerivative(U, Dq),t),diff(q), Dq)  ...
    + simplify(functionalDerivative(T-U, q))                 ...
    - functionalDerivative(D, Dq)                            ...
    + Q;

% replace placeholders
qdotdot = subs(qdotdot, X, RHS);
%% Dynamic system definition
% inputs vector
u= [FX_fr FY_fr MZ_fr ...
    FX_fl FY_fl MZ_fl ...
    FX_rr FY_rr MZ_rr ...
    FX_rl FY_rl MZ_rl ...
    MW_fr MW_fl MW_rr MW_rl ...
    M_sf M_sr].';
% state derivative function
xdot = [Dq; qdotdot];

% wheel loads for friction calculations (obtained as suspension forces)
FZ = -k_f * d - b_f * Dd;

params = [g; t_f; t_r; l_f; l_r; q_f; q_r; h_CG; I_f; I_r; k_f; k_r;
    b_f; b_r; m; m_u; Ixx; Iyy; Izz; Ixz; I_u; r_0; I_w; h_f; h_r];

%% Function handles contruction and saving
% redefine lagrangian variables and derivatives so we don't need to
    worry
% about the time variable
syms y p r Dy Dp Dr x_CG y_CG z_CG Dx_CG Dy_CG Dz_CG
syms gamma_fr gamma_fl gamma_rr gamma_rl
syms Dgamma_fr Dgamma_fl Dgamma_rr Dgamma_rl
```

```
266 syms delta_f delta_r Ddelta_f Ddelta_r
267
268 %state vector
269 x= [y; p; r; x_CG; y_CG; z_CG;
270      delta_f; delta_r; gamma_fr; gamma_fl; gamma_rr; gamma_rl;
271      Dy; Dp; Dr; Dx_CG; Dy_CG; Dz_CG;
272      Ddelta_f; Ddelta_r; Dgamma_fr; Dgamma_fl; Dgamma_rr; Dgamma_rl ];
273
274 % replace with the new time independent coordinates
275 xdot_ = subs(xdot(t), [q(t); Dq(t)], x);
276 FZ_   = subs(  FZ(t), [q(t); Dq(t)], x);
277 CPV_  = subs( v_w(t), [q(t); Dq(t)], x);
278
279 disp 'writing body dynamics function to file'
280 delete BodyDynamicsFunction.m
281 matlabFunction(xdot_, 'Vars', {x, u,
        params},'File','BodyDynamicsFunction');
282
283 disp 'writing wheel loads function to file'
284 delete WheelLoadsFunction.m
285 matlabFunction(FZ_, 'Vars', {x, params},'File','WheelLoadsFunction');
286
287 disp 'writing contact point velocities function to file'
288 delete ContactPointVelocitiesFunction.m
289 matlabFunction(CPV_, 'Vars', {x,
        params},'File','ContactPointVelocitiesFunction');
```

Bibliography

[1] I. J.M. Besselink, A. J.C. Schmeitz, and H. B. Pacejka. "An improved Magic Formula/Swift tyre model that can handle inflation pressure changes". In: *Vehicle System Dynamics* 48.sup1 (2010), pp. 337–352. DOI: 10.1080/00423111003748088.

[2] Marc Compere. *Simple 2D kinematic vehicle steering model and animation*. https://it.mathworks.com/matlabcentral/fileexchange/54852-simple-2d-kinematic-vehicle-steering-model-and-animation. 2018 (accessed July 17, 2018).

[3] Lucio Demeio. *Elementi di meccanica classica per ingegneria*. CittàStudi, 2016. ISBN: 9788825174120.

[4] Marco Furlan. *LoadTIR.m source file*. https://www.mathworks.com/examples/simulink/community/32257-loadtir. 2017 (accessed July 11, 2018).

[5] T.D. Gillespie. *Fundamentals of Vehicle Dynamics*. Premiere Series Bks. Society of Automotive Engineers, 1992. ISBN: 9781560911999.

[6] U. Kiencke and L. Nielsen. *Automotive Control Systems: For Engine, Driveline, and Vehicle*. Springer Berlin Heidelberg, 2005. ISBN: 9783540264842.

[7] Vittorio Lorenzi. *Adams/Tire – Using the PAC2002Tire Model*. http://mech.unibg.it/~lorenzi/VD&S/Matlab/Tire/tire_models_pac2002.pdf. 2012 (accessed July 11, 2018).

[8] W.F. Milliken, D.L. Milliken, and Society of Automotive Engineers. *Race Car Vehicle Dynamics*. Premiere Series. SAE International, 1995. ISBN: 9781560915263. URL: https://books.google.it/books?id=opgHfQzlnLEC.

[9] Hans Pacejka. *Tire and Vehicle Dynamics*. Butterworth-Heinemann, 2012. ISBN: 9780080970165.

[10] H.B. Pacejka. "The tyre as a vehicle component". In: *26th FISITA Congress, 16-13 June 1996, Prague, Czechoslovakia* (1996).

[11] *SAE J670 – Vehicle Dynamics Terminology*. Standard. Society of Automotive Engineers, Jan. 2008.

[12] ETH Zurich. *Robot Dynamics Lecture Notes*. `https://www.ethz.ch/content/dam/ethz/special-interest/mavt/robotics-n-intelligent-systems/rsl-dam/documents/RobotDynamics2016/RD2016script.pdf`. 2017 (accessed July 16, 2018).