UNIVERSITÀ DI BOLOGNA

School of Engineering

Master Degree in Automation Engineering

Modeling and Simulation of Mechatronic Systems

MODELING AND SIMULATION OF A RIGID CAR WITH LONGITUDINAL LOAD TRANSFER

Professor: **Alessandro Macchelli**

Tutor: **Marco Borghesi**

Students: Cenerini Simone
Cutini Giulia
Musolesi Nicoló
Simonazzi Daniele

Academic year 2022/2023

Abstract

The project is organized as follows:

- in Chapter 1 we outline how we have derived the Bi-cycle model
- in Chapter 2 we describe the development of the tire model
- In Chapter 3 we explain the expansion of the model found previously from the bicycle to the rigid car model, so the 4-wheels model
- In Chapter 4 we present a proper controller for our model

Contents

Chapter 1

Bicycle Model

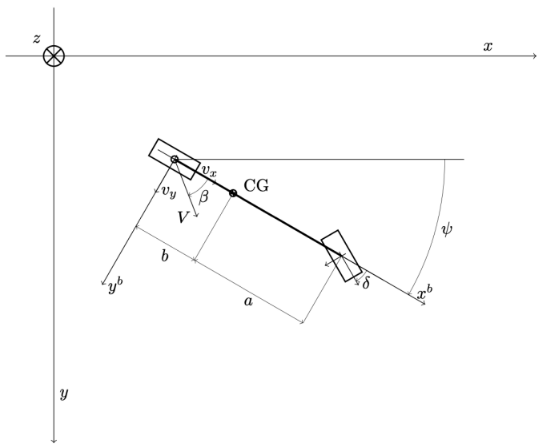


Figure 1.1: Bicycle scheme

The first step of our project was to find the model describing the dynamic behaviour of the bicycle vehicle represented in 1.1. The *bicycle model* is a planar model that approximates the vehicle as a rigid body with two wheels.

Load transfer is introduced even without adding suspensions model. This is achieved by modelling the tire normal loads by means of reaction forces generated at the vehicle contact points with the ground.

As we can see in 1.1 the body frame of the vehicle is attached at the rear contact point with $x - y - z$ axes oriented in a forward-right-down fashion.

1.1 Derivation of model via Lagrangian approach

To find the model we have followed the Langrangian approach, a commonly used mathematical method to describe the dynamic behaviour of mechanical systems. The system is modelled as a single planar rigid body with five degrees of freedom (three displacements (x, y, z) and two rotations (ψ, θ)). It's constrained to move in a plane (three degrees of freedom to describe such motion) interacting with the road at two body-fixed contact points. The set

of *generalized coordinates* used to develop the model is the following:

$$\begin{aligned} q_r &= [x, y, \psi]^T \\ q_c &= [z, \theta]^T \end{aligned} \tag{1.1}$$

where q_r are the reduced unconstrained coordinates, while q_c are the constrained one.

In order to apply the Lagrangian approach, we start by defining the Lagrangian function, which is the difference between the system's kinetic and potential energies:

$$\mathcal{L}(q, \dot{q}) = \mathcal{T}(q, \dot{q}) - \mathcal{U}(q) \tag{1.2}$$

where $\mathcal{T}$ represents the total kinetic energy of the system, and $\mathcal{U}$ represents the potential energy. For what regards the *Kynetic Energy* it has two terms respectively arising from the translational motion and the rotational motion of the bicycle (angular velocity).

$$\mathcal{T} = \frac{1}{2} \cdot m \cdot {v_{lin}}^2 + \frac{1}{2} \cdot {w_{rot}}^T \cdot I \cdot w_{rot} \tag{1.3}$$

The *Potential Energy* takes into account the gravitational potential energy associated with the height of the bicycle's centre of mass.

$$\mathcal{U} = -m \cdot g \cdot h_{CG} \tag{1.4}$$

These equations are written in the absolute reference frame, thus we will need rotation matrix to pass from the absolute reference frame to the body-fixed one.

Lagrange equations for constrained systems are the following:

$$\begin{cases} \frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}}\right) - \frac{\partial \mathcal{L}}{\partial q} = \tau - A^T(q)\lambda \\ A(q)\dot{q} = 0 \end{cases} \tag{1.5}$$

The generalized forces τ have been found by projecting ("into the joint space") the unconstrained forces $f_r = [f_{fx}, f_{fy}, f_{rx}, f_{ry}]$, using the transpose of the Jacobian.

$$J(q) = \begin{bmatrix} \cos(\psi)\cos(\theta) & \sin(\psi)\cos(\theta) & 0 & -\sin(\theta) & 0 \\ -\sin(\psi) & \cos(\psi) & (a+b)\cos(\theta)) & 0 & 0 \\ \cos(\psi)\cos(\theta) & \sin(\psi)\cos(\theta) & 0 & -\sin(\theta) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 & 0 & 0 \end{bmatrix} \tag{1.6}$$

The above is the Jacobian matrix, which maps the $\dot{q}$ into the front and rear contact point velocities expressed in body frame.

$\lambda = f_c = [-f_{fz}, -f_{rz}]$ represents the stack of the constrained forces. These forces comes from the fact that both the front and the rear vehicle's contact points have to keep the contact with the road ($\dot{z}_f = \dot{z}_r = 0$). Such kinematic constraint have to be written in the *Pfaffian form* $A(q)\dot{q} = 0$ where

$$A(q) = \begin{bmatrix} \cos(\psi)\sin(\theta) & \sin(\psi)\sin(\theta) & 0 & \cos(\theta) & -(a+b) \\ \cos(\psi)\sin(\theta) & \sin(\psi)\sin(\theta) & 0 & 0 & \end{bmatrix} \tag{1.7}$$

The matrix A is then used in the computation of the "constrained lagrangian forces" which influence the car dynamics.

In conclusion:

$$\begin{aligned} F_{gen} &= J^T \cdot f_r \\ F_{constr} &= A^T \cdot f_c \end{aligned} \tag{1.8}$$

At this point, by solving the Lagrange equations we have obtained a set of differential equations that describe the dynamic behaviour of the bicycle model. These equations, once casted in the body fixed reference frame, involve the generalized coordinates and their time derivatives, as well as the forces acting on the system. We have made all the calculations in a python file named `Lagrange.py` using the sympy package to address all the derivatives.

$$\begin{bmatrix} m & 0 & 0 & \mu_{fx} & \mu_{rx} \\ 0 & m & mb & \mu_{fy} & \mu_{ry} \\ 0 & mb & (I_{zz}+mb^2) & (a+b)\mu_{fy} & 0 \\ 0 & 0 & 0 & -1 & -1 \\ -mh & 0 & 0 & a+b & 0 \end{bmatrix} \begin{bmatrix} \dot{v}_x \\ \dot{v}_y \\ \ddot{\psi} \\ f_{fz} \\ f_{rz} \end{bmatrix} + \begin{bmatrix} -mb\dot{\psi}^2 - mv_y\dot{\psi} \\ mv_x\dot{\psi} \\ mbv_x\dot{\psi} \\ -mg \\ (I_{zz}+mhb)\dot{\psi}^2 + mhv_y\dot{\psi} + mgb \end{bmatrix} = 0 \tag{1.9}$$

The first three equations are the equations of motion for the unconstrained coordinates v_x ,v_y and $\dot{\psi}$ while the last two equations describe the constrained forces acting on the vehicle, given by the reaction forces of the ground.

The vector $\mu = [\mu_{fx}, \mu_{fy}, \mu_{rx}, \mu_{ry}]$ are the inputs of the system.

1.2 Considerations about the model

The validity and applicability of the derived equations depend on the specific assumptions and simplifications made in the model.

The phenomena that this model can take into account are:

- Translation and yaw rotation motion: the model considers both the linear motion of the vehicle along the x and y directions, and the yaw $\dot{\psi}$ rotation
- Tire forces: the model incorporates the tire forces acting on the bicycle. The variables f_{fz} and f_{rz} represent the vertical forces exerted on the front and rear tires, respectively
- Mass and inertia: the model accounts for the mass and inertia properties of the bicycle.
- Gravity effects
- Centripetal and centrifugal forces: the terms involving $\dot{\psi}$ and v_y capture the centripetal and centrifugal forces acting on the bicycle as it undergoes yaw rotation and lateral motion

For what regards the **limits of validity** of the model we can highlight:

- Speed limitations: bicycle model often assume certain speed ranges in which the model accurately represents the dynamics behaviour. At extremely high speeds or very low speeds, additional factors such as tire slip, aerodynamic effects and non linear dynamics may be considered.
- The interaction forces between ground and tiresare modelled by a specific formula, the Pacejka Magic formula, which has its own limitations: it does not take into account dissipative effect, for example
- Suspension and tire characteristics: the bicycle model takes not into account the effects of suspension dynamics or specific tire characteristics. These factors can significantly influence the behaviour of the bicycle and may require additional modeling considerations.
- Lateral load transfer is not modelled at all

1.3 Simulink implementation

We have implemented the bicycle model in Matlab-Symulink environment considering the $\mu = [\mu_{fx}, \mu_{fy}, \mu_{rx}, \mu_{ry}]$ as the input of the `Car dynamics` function block.

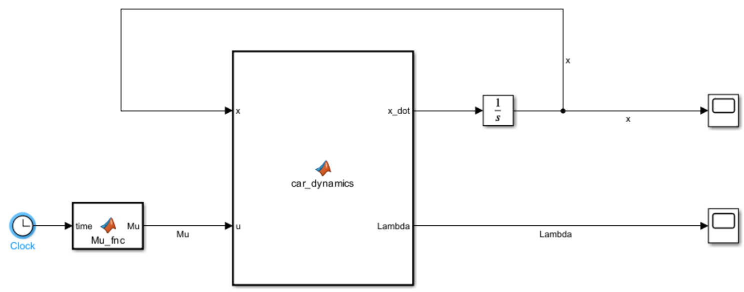


Figure 1.2: Simulink implementation of the bicycle

Inside the function block the 1.9 are solved by means of a matrix inversion and, given meaningful values for the input variables, corresponding output values are found. These output are then divided into $\dot{x} = [\dot{v}_x, \dot{v}_y, \ddot{\psi}]$ that is integreted into $x = [v_x, v_y, \dot{\psi}]$ and gave back as input, and the $\lambda = [f_{fz}, f_{rz}]$.

Chapter 2

Tire Model

The bicycle model developed in chapter 1, even if it's of great interest since it's able to describe many important phenomena with equations of reasonable complexity, it does not take into account many important dynamic effects such as the **lateral load transfer** and the tire model. In order to introduce an additional level of complexity in our previous model, we have added a proper model of the vehicle tires.

2.1 Pacejka's magic formula

We have modelled the tire forces by using a suitable version of the **Pacejka's Magic Formula** able to describe the nonlinear behaviour of tires. It provides a way to predict the forces and moments that a tire generates when subjected to various operating conditions, such as different slip angles, slip ratios, and vertical loads.

In order to write the Magic Formula we have to introduce two quantities: the *slip ratio* and the *slip angle*.

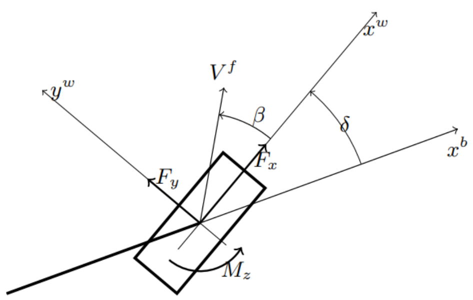


Figure 2.1: Scheme of the quantities used in the tire model

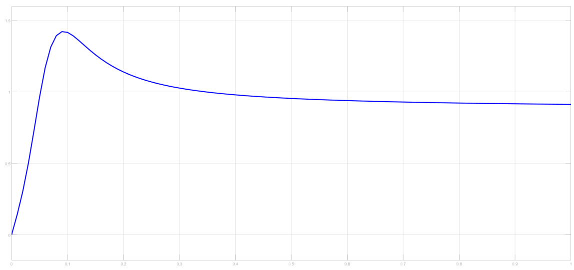


Figure 2.2: μ evolution w.r.t. σ (we considered only positive values of σ since we won't encounter negative values)

The quantity σ_x is know as ***slip ratio*** (or longitudinal slip) and quantifies the relative difference between the rotational speed of a wheel when slipping and rotational speed of a wheel in conditions of pure rolling.

$$\sigma_x = \frac{w_w - w0}{w0} = -\frac{v_{cx} - r_w w_w}{v_{cx}} \tag{2.1}$$

It's an essential factor in understanding and modeling tire behaviour in the longitudinal direction.

A positive slip ratio ($\sigma_x > 0$) is associated with acceleration or driving forces, conversely, a negative slip ratio is associated with braking or deceleration forces.

In the tire models like Pacejka's Magic Formula, the slip ratio is one of the input parameters used to determine the longitudinal tire force. In our model we distinguish σ_r and σ_f respectively for the rear and front wheel.

The β factor known as ***slip angle*** (or lateral slip) is a parameter that describes the angle between the direction of motion of the vehicle and the direction in which the tires are pointing.

$$\beta = \arctan \frac{v_{cy}}{v_{cx}} \tag{2.2}$$

It plays a crucial role in understanding and modeling tire behaviour in the lateral or sideways direction. The slip angle is influenced by the vehicle's steering input and its interaction with the road surface.

When the tire is aligned with the direction of travel, the slip angle is zero. A positive slip angle ($\beta > 0$) indicates that the tire is deviating to the right (for a left turn) or to the left (for a right turn) relative to the direction of travel. Conversely, a negative slip angle ($\beta < 0$) indicates that the tire is deviating in the opposite direction. When a tire experienced a slip angle, it generates lateral forces that contribute to the vehicle's ability to corner and maintain a desired trajectory. The slip angle influences the magnitude and direction of these lateral forces, which play a significant role in vehicle handling, steering response and stability. The slip angle can vary depending on factors such as vehicle speed, tire properties, road conditions, and steering inputs.

Pacejka's formula incorporates three terms: the longitudinal force f_x due to the slip ratio, the lateral force f_y due to the slip angle and the combined longitudinal and lateral forces $g_{x\beta}$ and $g_{y\sigma}$.

$$\begin{aligned} f_{x0}(\sigma) &= d_x \sin\{c_x \arctan[b_x\sigma - e_x(b_x\sigma - \arctan b_x\sigma)]\} \\ f_{y0}(\beta) &= d_y \sin\{c_y \arctan[b_y\beta - e_y(b_y\beta - \arctan b_y\beta)]\} \end{aligned} \tag{2.3}$$

$$\begin{aligned} g_{x\beta}(\sigma, \beta) &= \cos\left[c_{x\beta} \arctan\left(\beta \frac{r_{bx1}}{1 + {r_{bx2}}^2\sigma^2}\right)\right] \\ g_{y\sigma}(\sigma, \beta) &= \cos\left[c_{x\sigma} \arctan\left(\sigma \frac{r_{by1}}{1 + {r_{bx2}}^2\beta^2}\right)\right] \end{aligned} \tag{2.4}$$

where the Pacejka's coefficients where given.

The most important thing to say about the Pacejka's equations is that the longitudinal and lateral forces developed by the tires in their reference frame (f_x and f_y) are supposed to depend linearly on the vertical force of the wheel on the ground f_z .

$$\begin{aligned} f_x &= -f_z f_{x0}(\sigma) g_{x\beta}(\sigma, \beta) = -f_z \mu_x(\sigma, \beta) \\ f_y &= -f_z f_{y0}(\beta) g_{y\sigma}(\sigma, \beta) = -f_z \mu_y(\sigma, \beta) \end{aligned} \tag{2.5}$$

The μ are the friction coefficients that depend on σ and β.

2.2 Tire dynamics

In order to give proper σ_x and β to Pacejka, we have implemented a simple dynamics for both the front and rear wheels, with a proper inertia term. Thanks to this block we are able to use front and rear toques as new input for our total dynamics.

In order to write the wheel dynamics we have choose the following parameters:

r_w	wheel radius	$0.25m$
m_w	wheel mass	$15kg$
I_w	wheel inertia	$\frac{1}{2}m_w(r_w)^2$

and we have computed the angular accelerations of the wheels with the following dynamic equation:

$$\alpha = \frac{\tau - r_w f_x}{I_w} \tag{2.6}$$

Then, given the velocities of the front and rear contact points expressed in the body- fixed frame we have used the following proper rotation matrix to translate them into the wheel-fixed frame.

$$R_z(\delta) = \begin{bmatrix} \cos(\delta) & \sin(\delta) & 0 \\ -\sin(\delta) & \cos(\delta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.7}$$

After that we have done the following steps for both the front and rear tires:

- Calculate slip angles: starting from the velocities of the contact points (in wheel-frame) we have developed the following calculations:

$$\begin{aligned} \beta_f &= -\arctan\left(\frac{v_{cfy}}{v_{cfx}}\right) \\ \beta_r &= -\arctan\left(\frac{v_{cry}}{v_{crx}}\right) \end{aligned} \tag{2.8}$$

  where v_{cf} and v_{cr} are the contact points velocities respectively of front and rear wheels.

- Calculate slip ratios:

$$\begin{aligned} \sigma_f &= \frac{w_r w_{nf} - v_{cfx}}{v_{cfx}} \\ \sigma_r &= \frac{w_r w_{nr} - v_{crx}}{v_{crx}} \end{aligned} \tag{2.9}$$

  In order to compute the angular velocities of the wheels (w_{nf} and w_{nr}) we have integrated the angular accelerations (α_f and α_r) previously computed.

2.3 Simulink implementation

In order to properly model tires we have introduced two separate blocks, upstream of the `Dynamic` block. The first one is the `Tire dynamics` block, which takes the torques on front and rear wheels as input. It takes into account the dyamics for each wheel and produces the proper coefficients, namely σ_x and β, as input for the `Pacejka` block. The `Pacejka tire model` block, on the other hand, takes the σ_x and β coefficients as input, and generates the μ coefficients as output using the calculations described in section 1.

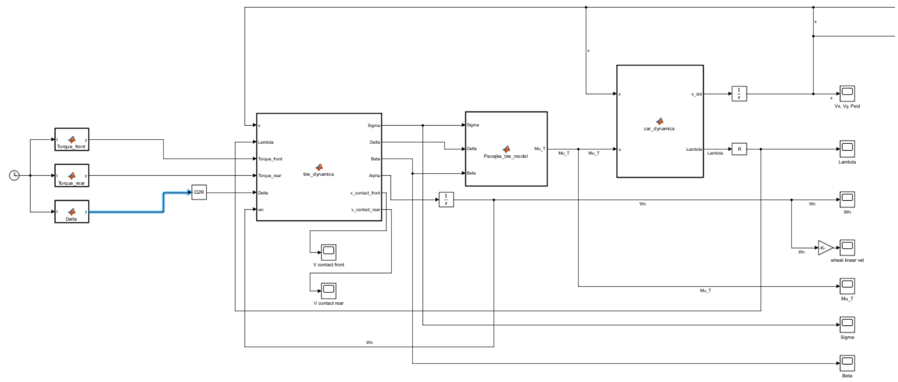


Figure 2.3: Symulink implementation of bicycle with tires dynamic

2.4 Simulations

Our simulations have been developed with an Euler integrator (ode1) and an integration step equal to $1e^{-5}$. This small integration step is required since the wheel's dynamics changes very fast and so a small integrator step is required in order to avoid oscillations.

For what regards the initial conditions of the integrators used in the Simulink model: the one used to integrate the state variables from $[\dot{v}_x, \dot{v}_y, \ddot{\psi}]$ into $[v_x, v_y, \dot{\psi}]$ has been initialized with a constant value for v_x and zero elsewhere. While the one used to integrate from the anglular acceleration α into the angular velocity w has been initialized with a value equal to the velocity along x direction over the wheel radius $(\frac{v_x}{w_r})$ for both the front and rear wheels.

Simulations can be found at the following link: One-Drive

Chapter 3

Rigid Car Model

In this section we want to develop a reduced-order model for a two-track car vehicle, called *Rigid Car*. Differently from the bicycle model considered up to now, the rigid car captures the main dynamic features of the full-vehicle and, in particular **longitudinal and lateral load transfer**.

3.1 Derivation of model via Lagrangian approach

The model has been found by using the Lagrangian approach (as before), and all the calculations have been developed in the `Lagrangian equations.py`.

The first step was, as in the bicycle case, to find a proper set of generalized coordinates to describe the dynamics of the rigid car:

$$\begin{aligned} q_r &= [x, y, \psi]^T \\ q_c &= [z, \phi, \theta]^T \end{aligned} \tag{3.1}$$

where q_r are the reduced unconstrained coordinates, while q_c are che constrained ones. The ϕ, θ and ψ angles are respectively the roll, pitch and yaw angles used to describe the orientation of the car model (body-fixed frame) with respect to the absolute reference frame. This transformation is properly described by the following rotation matrix:

$$R = R(\phi, \theta, \psi) = R_z(\psi)R_y(\theta)R_x(\phi) = \begin{bmatrix} c_\psi c_\theta & c_\psi s_\theta s_\phi - s_\psi c_\phi & c_\psi s_\theta c_\phi + s_\psi c_\phi \\ s_\psi c_\theta & s_\psi s_\theta s_\phi - c_\psi c_\phi & s_\psi s_\theta c_\phi + c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix} \tag{3.2}$$

Once we have all our quantities expressed in the absolute reference frame we can write the Lagrangian function and follow the same steps already described in Chapter 1.

The only difference to take into account is that the rigid car, differently from the bicycle, interacts with the ground at four body-fixed contact points.

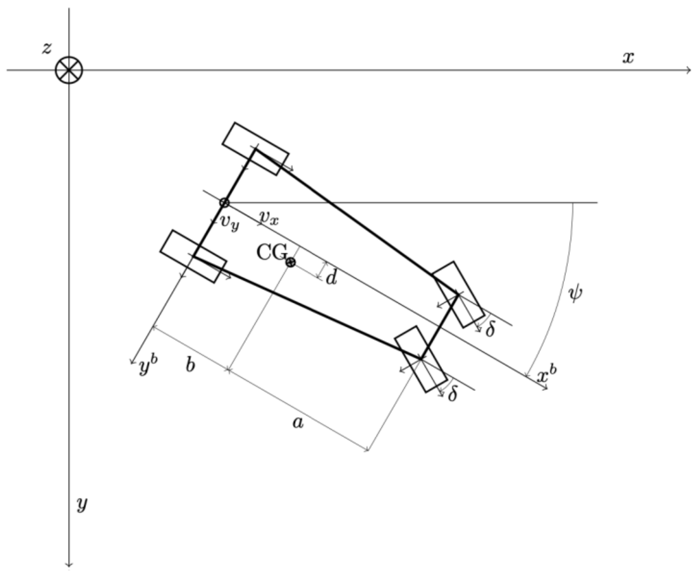


Figure 3.1: Rigid car scheme

Thus it results to be a hyper-static structure (it has more support reactions that necessary to maintain equilibrium) so that the 4 reaction forces are not uniquely determined. In fact we are dealing with a system with 7 unknowns $(\dot{v_x}, \dot{v_y}, \ddot{\psi}, f_{frz}, f_{flz}, f_{rrz}, f_{slz})$ but just 6 equations. We use the principle of least work to get a **compatibility equation** and thus resolve the indeterminateness by adding as 7-th equation the following:

$$-c_{fr}t_r f_{frz} + c_{fl}t_r f_{flz} + c_{rr}t_f f_{rrz} - c_{rl}t_f f_{rlz} = 0 \tag{3.3}$$

where t_f, t_r are the parameters indicating the coordinates of the front and rear tire in the body fixed reference fram, while $c_{fr}, c_{fl}, c_{rr}, c_{rl}$ are mechanical ones that include material properties such as Young's modulus.

The compatibility equation enables the explicit derivation of four reaction forces due to the load distribution of vertical loads among the contact points.

At the end of calculations the dynamics equations obtained are the following:

$$\begin{bmatrix} m & 0 & -md & \mu_{frx} & \mu_{flx} & \mu_{rrx} & \mu_{rlx} \\ 0 & m & mb & \mu_{fry} & \mu_{fly} & \mu_{rry} & \mu_{rly} \\ -md & mb & I_{zz}+m(b^2+d^2) & l\mu_{fry}-t_f\mu_{frx} & l\mu_{fly}-t_f\mu_{flx} & -t_r\mu rrx & t_r\mu rlx \\ 0 & 0 & 0 & -1 & -1 & -1 & -1 \\ 0 & mh & I_{xz}+mbh & -t_f & t_f & -t_r & t_r \\ -mh & 0 & mhd & l & l & 0 & 0 \\ 0 & 0 & 0 & -c_{fr}t_r & c_{fl}t_r & c_{rr}t_f & -crlt_f \end{bmatrix} \cdot \begin{bmatrix} \dot{v}_x \\ \dot{v}_y \\ \ddot{\psi} \\ f_{frz} \\ f_{flz} \\ f_{rrz} \\ f_{rlz} \end{bmatrix} + \begin{bmatrix} -m\dot{\psi}(v_y+b\dot{\psi}) \\ m\dot{\psi}(v_x-d\dot{\psi}) \\ m\dot{\psi}(dv_v-bv_x) \\ -mg \\ mh\dot{\psi}(v_x-d\dot{\psi})-mgd \\ mhv_y\dot{\psi}+(I_{xz}+mhb)\dot{\psi}^2+mgb \\ 0 \end{bmatrix} = 0 \tag{3.4}$$

The only new parameter introduced with respect to the bicycle model is d, the y-position of the centre of mass of the vehicle expressed in body-fixed reference frame (see figure 3.1), and all the part related to the compatibility equation.

3.2 Considerations about the model

The main important difference between the bicycle model and the 4-wheels one is that the second, more complex, one takes into account also the lateral and longitudinal load transfer. In order to derive the 4-wheels model we have introduced a **compatibility equation** in order to describe how the vertical load is distributed between the 4 wheels. To derive it we need to consider the equilibrium of forces and moments acting on the vehicle during cornering. Once all the forces and moments acting on the vehicle have been identified, we are required to apply equilibrium conditions:

$$E = \sum_{\substack{i=f,r \\ j=r,l}} \frac{1}{2} c_{ij} f_{z,ij}^2 \tag{3.5}$$

where E is the total amount of energy stored or dissipated in a rigid car model during cornering. In order to get our compatibility equation we can minimize this energy's derivative with respect to the unknown reaction forces.

3.3 Simulink implementation

In this chapter we just focus our attention in the changes with respect to the bicycle version.

The block Car dynamics have been changed and the new dynamic has been implemented. This block take as input

$$\mu = [\mu_r, \mu_l] = [\mu_{frx}, \mu_{rrx}, \mu_{fry}, \mu_{rry}, \mu_{flx}, \mu_{rlx}, \mu_{fly}, \mu_{rly}]. \tag{3.6}$$

Upstream of this block we have inserted two separate combination of elements, one for what regards the left side of the car (that produces $[\mu_{flx}, \mu_{rlx}, \mu_{fly}, \mu_{rly}]$) and another one for the right side ($[\mu_{frx}, \mu_{rrx}, \mu_{fry}, \mu_{rry}]$.

The block elements that implement the Pacejka and the Tire dynamics are the same used in the bicycle version in Chapter 2.

In the final Simulink implementation of the 4-wheels model we can insert from outside proper values of front and rear torque, and a proper steering angle δ.

3.4 Simulations

Simulations can be found at the following link: One-Drive

Chapter 4

Control

Once the bicycle model and the rigid car model were completed, we proceeded in the decision of which one to control and with which type of control. Since the last semester we developed an optimal control scheme for a simplified bicycle model, we thought that it would be interesting to try it on a more complex bicycle model.

4.1 Simplified vs Complex

Let's first introduce a quick comparison between the two models.

- the OPTCON Model was characterized by a static distribution of load, the MSMS Model instead takes into longitudinal load transfer
- the OPTCON Model assumed that μ and F_x were decoupled, meaning that μ was assumed constant and influenced only the values of F_y, while F_x was our chosen control input and we were free to assign it. In contrast, the MSMS Model uses the Pacejka Magic formula, which relates μ to Fx.

4.2 The OPTCON alhorithm

The MSMS model should perform an 8-shaped trajectory, the "skidpad". We approached the task by first solving an OPTCON problem with quadratic cost function on the simplified bicycle model, this was done by means of the Newton's Method with and adaptive stepsize (Armijo's rule). Once we got our optimal states-input trajectory, X_{opt} and U_{opt}, we used a trajectory-tracking algorithm (the controller) on our MSMS Model to track such optimum. Such controller can be found as the solution of a Linear-Quadratic-Program.

Algorithm 2 LQ Optimal Controller

Given an optimal state-input trajectory $(\mathbf{x}_{\text{opt}}, \mathbf{u}_{\text{opt}})$ obtained via Newton's method;
Linearize the system around $(\mathbf{x}_{\text{opt}}, \mathbf{u}_{\text{opt}})$:
for $t = 0, ..., T$ **do**
    $A_t^{\text{opt}} = \nabla_{x_t} f(x^{\text{opt}}, u^{\text{opt}})^T$
    $B_t^{\text{opt}} = \nabla_{u_t} f(x^{\text{opt}}, u^{\text{opt}})^T$
    $\Delta x_{t+1} = A_t^{\text{opt}} \Delta x_t + B_t^{opt} \Delta u_t$
end for
Compute the LQ Optimal Controller
Set $P_T = Q_T^{reg}$ and
for $t = T-1, ..., 0$ **do**
    Compute:
    $P_t = Q_t^{reg} + A_t^{\text{opt}^T} P_{t+1} A_t^{\text{opt}} - (A_t^{\text{opt}^T} P_{t+1} B_t^{\text{opt}})(R_t^{\text{reg}} + B_t^{\text{opt}^T} P_{t+1} B_t^{\text{opt}})^{-1}(B_t^{\text{opt}^T} P_{t+1} A_t^{\text{opt}})$
end for
Compute the feedback gain
for t=0,1,....,T-1 **do**
    Compute:
    $K_t^{\text{reg}} = -(R_t^{\text{reg}} + B_t^{\text{opt}^T} P_{t+1} B_t^{\text{opt}})^{-1}(B_t^{\text{opt}^T} P_{t+1} A_t^{\text{opt}})$
end for
Track the generated Optimal Trajectory
for $t = 0, ..., T-1$ **do**
    $u_t = u_t^{\text{opt}} + K_t^{\text{reg}}(x_t - x_t^{\text{opt}})$
    $x_{t+1} = f(x_t, u_t)$
end for

Figure 4.1: The Optcon Algorithm

4.3 Matching of the Inputs

There are only two things left to do:

- Expand the MSMS state vector from 3 to 6, including also x, y, ψ. The state expansion, in addition to the choice of the weight-matrices, allow us to either devlop a position or velocity control
- Convert the two inputs coming from the OPTCON Model into the three inputs needed by the MSMS one.

The expansion of the state vector is trivial, it is just a matter of defining three new dynamics equations by projecting V_x and V_y in the inertial reference frame.

Switching to the second point, the problem is a little more complex and requires some assumptions. The MSMS Model needs as inputs σ_f, σ_r and δ, while the control law coming from OPTCON provides us with F_x and δ. We first made the assumption that only the front wheel is actuated, thus we immediately placed σ_r to 0. Thus, we now need to convert F_x in σ_f.

From Pacejka, we know that:

$$f_x = -f_z f_{x0}(\sigma) g_{x\beta}(\sigma, \beta) = -f_z \mu_x(\sigma, \beta) \tag{4.1}$$

We need two more assumptions:

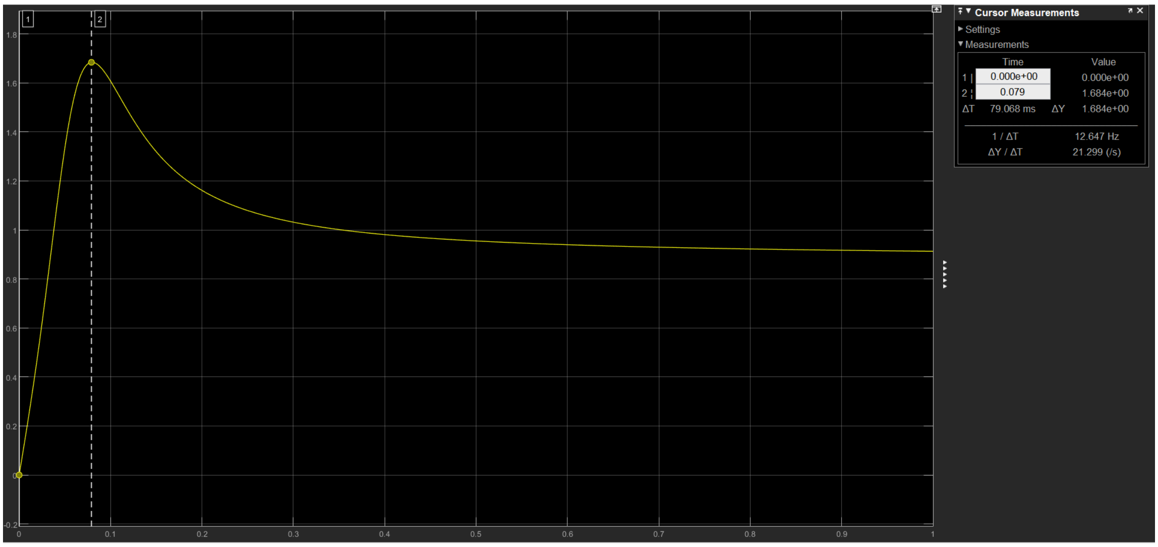
- Static load distribution: $f_z = m * g * b/(a + b)$
- Linearization of Pacejka Magic Formula: $\mu_x(\sigma, \beta) = slope * \sigma$

Let's spend a few words on the linearization of Pacejka. *Beta* is assumed to be constant and equal to the value associated to the steering along the skidpad: 0.008. The *slope*-coeff was computed empirically by evaluating the slope of the following graph.

Now we just have to express σ as a function of F_x:

$$\sigma_f = \frac{slope * (a + b) * F_x}{m * b * g} \tag{4.2}$$

And now we are done.


Cursor Measurements
Settings
Measurements
Time
Value
1 0.000e+00 0.000e+00
2 0.079 1.684e+00
ΔT 79.068 ms ΔY 1.684e+00
1 / ΔT 12.647 Hz
ΔY / ΔT 21.299 (/s)
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9