Input Mapping

Introduction

All controllers that are generating inputs not compatible with the simulation model inputs need a mapping function in order to emulate their control actions. This document aims at clarifying how this mapping is performed and used through all the comparison.

Acceleration Mapping

All controllers taken in consideration for the comparison, except for Flatness-based control, generate the required steering angle δ and the longitudinal acceleration $a_{x,des}$ for tracking the reference trajectory. Since the model used in simulation has steering angle δ and front wheel speed ω_f as inputs, a mapping function is required in order to achieve $a_{x,des}$ with the available inputs.

The front wheel speed command is therefore calculated as

$$\omega_f = f_{ID}(\boldsymbol{\chi}, \mu, \delta, a_{x,des}), \tag{1}$$

where $\boldsymbol{\chi} = [x, y, \psi, v_x, v_y, \omega]^T$ contains the states of the vehicle and μ is the road friction coefficient. The vehicle state contains the position x, y expressed in world coordinates and the derivatives v_x, v_y expressed in local vehicle coordinates. The orientation ψ and its derivative ω are both expressed in world coordinates. In our implementation, the MATLAB function `AXController_pacejka` performs this operation.

The controller A presented in [1] calculates the required front wheel forces F_{xf} (longitudinal) and F_{yf} (lateral) such that the vehicle correctly tracks the trajectory. These forces are expressed in vehicle-fixed coordinates. The tire model is then inverted to obtain the inputs δ and ω_f.

By using a similar approach, the mapping f_{ID} calculates front wheel forces by means of δ and $a_{x,des}$. These forces are subsequently used to determine ω_f by inverting the tire-model, which was choosen to be explicitly invertible [1] .

Consider the vehicle longitudinal dynamics:

$$\dot{v}_x = \frac{F_{xf} + F_{xr}}{m} + v_y\omega \tag{2}$$

$$\dot{v}_y = \frac{F_{yf} + F_{yr}}{m} - v_x\omega \tag{3}$$

where $\dot{v}_x$ and $\dot{v}_y$ are the vehicle longitudinal and lateral acceleration respectively and r stands for rear. Since the controller request the longitudinal acceleration to be $a_{x,des}$, equation (2) can be rewritten as

$$F_{xf} = m(a_{x,des} - v_y\omega) - F_{xr}. \tag{4}$$

where F_{xr} is always zero because the rear tire is not actuated and pure rolling motion is assumed.

A specific value for the lateral acceleration $\dot{v}_y$ is not explicitly required by the controllers, therefore F_{yf} cannot be calculated in the same way as F_{xf}. The longitudinal acceleration is generated entirely by the longitudinal force F_{xf} on the front tire. Since a value for F_{yf} is required by the tire inversion algorithm, F_{yf} is set to zero such that $a_{x,des}$ is approximately generated by the simulated vehicle without taking into consideration tire saturation due to lateral forces. Therefore, the mapping function is only taking care of scaling the wheel speed whenever the required longitudinal force F_{xf} saturates the tire.

By inverting the tire model the required ω_f can be calculated. Consider the tire slip equations:

$$\boldsymbol{s}_{xy,f} = \left(\boldsymbol{v}_{xy} - \boldsymbol{R}(\delta) \begin{bmatrix} r\omega_f \\ 0 \end{bmatrix} \right) / \|\boldsymbol{v}_{xy}\| \tag{5}$$

where $s_{xy,f}$ is the front tire slip, v_{xy} is the velocity at the front axle in local vehicle coordinates, $R(\cdot)$ is a rotation matrix and r is the tire radius.

Since $R(\delta)$ is fixed (δ has been already calculated from the controller), only ω_f is left to satisfy equation (5). It is clear that, if the direction of the slip $s_{xy,f}$ is not perfectly aligned with the tire direction δ, the equation is over-constrained. However, this is not a problem if we want to control only the longitudinal tire slip (and consequently the acceleration). In fact, we can use ω_f to satisfy only the longitudinal part:

$$s_{x,f} = \left(v_x - cos(\delta) r\omega_f \right) / \|v_{xy}\|. \tag{6}$$

Now ω_f is found as

$$\omega_f = -\frac{s_{x,f}\|v_{xy}\| - v_x}{rcos(\delta)}. \tag{7}$$

References

[1] Heß, Daniel and Althoff, Matthias and Sattel, Thomas, *Comparison of trajectory tracking controllers for emergency situations*, Intelligent Vehicles Symposium (IV), 2013 IEEE.