Determining Tire characteristics through data driven modeling

Eugene Figueiras
BSC student
TU Delft
SN:0000000

Maarten Kleijwegt
BSC student
TU Delft
SN:4113810

Nol Römer
BSC student
TU Delft
SN: 4391888

Akash Soerdjbalie
BSC student
TU Delft
SN: 4227174

dr.ir. Tamás Keviczky
Associate Professor
Delf Center for Systems and Control

Wil van Geest
???
Delft Center for Systems and Control

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Maecenas porttitor congue massa. Fusce posuere, magna sed pulvinar ultricies, purus lectus malesuada libero, sit amet commodo magna eros quis urna. Nunc viverra imperdiet enim. Fusce est. Vivamus a tellus. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Proin pharetra nonummy pede. Mauris et orci.

1 Introduction

If you talk to racing enthusiasts you quickly find out one of the key components in any competition car are the tires. They provide the interface between all the components a race car consists of and the track it's driving on. Therefore, tires dictate any behaviour a car is capable of. Those 4 foot sized patches of rubber make the difference between being able to make those quick sharp corners and spinning out of the ring. In daily traffic making those tires stick to the road can sometimes make the difference between life and death. In the United States alone, the National Highway Traffic Safety Administration (NHTSA) reports that 5 million crashes occurred in 2009 [Reference], causing over 30,000 deaths and 2 million injuries.

Until now, driver safety systems in modern-cars have focussed on keeping the car in the linear regime while driving. For example, ABS prevents the wheels from slipping and traction control prevents the car from 'breaking out'of its path. However, when we look towards professional rally drivers we see that operating in the linear regime isn't specifically necessary for driving safe. If we would be able to give the skills of a rally driver to driver assistance systems, or in the future autonomous cars, more accidents might be prevented. It would give forementioned systems more capabilities and options. Which might result in better accident prevention solutions.

This paper focuses on establishing a measurement setup and determining the tire characteristics of a small-scale RC car with it. It is a step towards achieving a good Nonlinear Model Predictive Control (NMPC) system. NMPC is a new control method which uses a dynamical model that can predict the motion of the car. Therefore, this system is able to let the car make evasive maneuvers up to the nonlinear regime of driving. In order to obtain a good dynamical model for the NMPC, tire characteristics are required. There are a few ways to obtain these characteristics. Normally, these characteristics are determined using special test rigs where only the wheels are placed in. However, these characteristics are only valid for that given tire, in only that given condition(think of temperature, wornness and wetness of the tire) on only a single given surface. A lot of testing can be done to determine the characteristics of all tires in all conditions. Yet it would be far better if a car would be able to determine it's characteristics using on board sensors such as GPS and Inertial Measurment Units or IMU's. Testing with real cars and on-board sensors in a special test environment is a possibility. But developing and refining this technique on small scale RC cars first requires less resources such as large test tracks and expensive cars and tires. Let alone other expenses like experienced mechanics and drivers.Therefore, using a small-scale RC car equipped with sensors is by far the least expen-

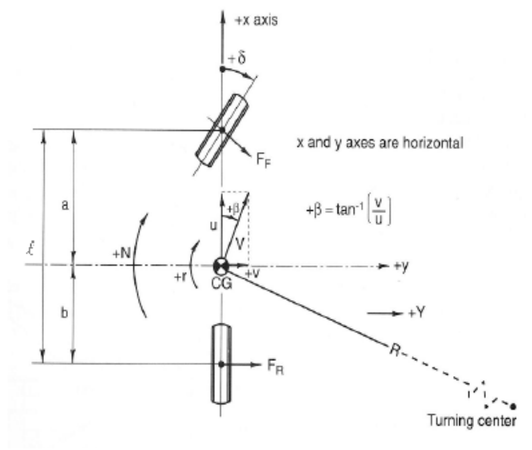


Figure 1. Bicycle model

sive way to obtain these tire characteristics. These methods can then be scaled for actual cars.

This all concludes to the aim of this paper, which is determining the tire characteristics of a small scale RC car using only on board sensors. At first, some scientific background is given on the subject. After that, the experimental setup and data processing will be explained. Next, results will be discussed. Finally, the conclusion will be given. The paper also leaves some room for recommendations and acknowledgements.

2 Background

In order to achieve Model Predictive Control, it's important to build a valid model of the dynamics of the RC car. In order to do so, it's important to understand the tire characteristics of this vehicle. This means there has to be a model for the tire characteristics as well. To build such a model, Data-Driven Model Design is used. To analyse the data of the test rig, a dynamical model is necessary. Accuracy of a dynamical model is closely related to the complexity of the model. However, due to the computational burden of these systems, a relative simple yet accurate model is necessary. Therefore, the Bicycle Model is chosen (Bron: 2014-01-0841 google drive pdf FIXME). With this model, it is possible to generate real time data that is still accurate.

Another suitable tire model would be the Dugoff tire model. This model provides a less accurate representation of the tire characteristics, but requires less variables. Even though fewer variables are needed though, more of the variables are unknown. Therefore, the Magic Formula model is easier to apply and gives a more accurate result, so to simplify calculations without a reduced quality of the final results, the Magic Formula model is chosen.

2.1 The Bicycle Model

In the Bicycle Model, the car is represented as a rigid, two-dimensional, two-wheel vehicle with a xyz-coordinate system is fixed to the car frame. The front wheels are represented as one wheel and so are the rear wheels [FIXME: verwijzing naar het onderstaande plaatje plus bijschrift bij plaatje] . This model comes with some important assumptions so there are some restrictions to test settings as well. The car should not move in the vertical (z-) position, nor rotate in the roll and pitch directions (around the x- and y- axes respectively). Therefore, movements are only possible in the x- and y-direction and as a rotation around the z-axis(yaw). Furthermore, the bicycle model is divided in two parts: the linear and the nonlinear bicycle model. In general, a linear model is used for simplicity of the controller. Even though the performance of this model degrades at the limits of friction, it is demonstrated that using a linear tire model is still a good way of modelling (BRON: 2012-thesis-Kritayakirana GOOGLE DRIVE page 9). Even though, the non-linear bicycle model gives an even more accurate approximation of vehicle dynamics. With only In the linear bicycle model, constant longitudinal velocity is assumed and lateral acceleration should not exceed 0.3 g . In order to meet these assumptions, the vehicle is not allowed to accelerate in the longitudinal direction while cornering. Therefore, data is not valid in the first seconds of each cornering test where the car has to accelerate. For the longitudinal motion tests however, it is necessary to accelerate and decelerate on the straight. If we still want to use the linear Bicycle Model for these experiments, the accelerations and decelerations cannot be very high. Otherwise this will result in inaccurate values given by this model. However, in some cases it will be helpful to use the nonlinear Bicycle Model. The limitations stated in the previous section will not occur. The nonlinear Bicycle Model will be used for tests where the longitudinal and lateral accelerations are high. This model comes with a larger set of equations and variables, and is solvable if the distribution of forces on the front and rear tires is known. In order to solve the equations, we assume a 50-50 distribution on these wheels, which makes sense if both the front and rear wheels are spinning at the same time. Because the wheels are connected to the same motor ,through differentials, and share the same properties, the forces acting on the wheels are approximately the same. When the forces acting on the wheels of the car are known, thanks to the bicycle model values can be found to determine the slip angle alpha and the slip ratio kappa. [FIXME: kappa en alfa griekse letters in latex] With these known, the tire characteristics can be determined using the Magic Formula tire model 1.

$$[F_{xwi}, F_{ywi}] = MF(F_{zi}, \alpha_i, \kappa_i, \mu)(i = f, r) \tag{1}$$

With equation 1 with every variable known except the friction coefficient μ, the variable μ can be determined. Then an overview is created of all the tire characteristics of the RC vehicle, which is represented in a 3D-plot.

3 Experimental setup

In order to determine the tire characteristics of the scaled vehicle, a good testbed is necessary. In this section the experimental setup used to gather the data needed for determining

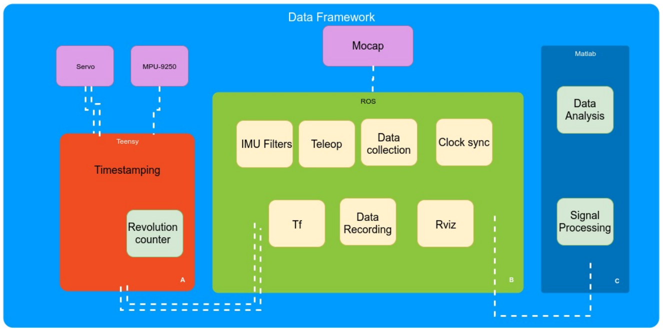


Figure 2. Data acquisition framework

the tire characteristic is discussed.
The experimental setup consists of a modified scaled RC car with an on board IMU, and a tachometer on each wheel. Moreover, a motion capture system or MoCap was used to provide millimeter precision locating.

3.1 Testbed

The scaled RC car is a Losi TEN Rally-X. It is a 1:10 scale car with 4WD. Each wheel was fitted with it's own hall effect sensor which generates a pulse every time one of the magnets inside the rim passes it. Many of the first tests were done using only 2 magnets giving only 1 pulse per pi radial turned. However after most of the testing was done we figured out a rim with 24 magnets would be much more suited. To fit both the 2 magnets and the 24 magnet versions to the car a set of custom 3d printed rims were created. Using the 24 magnet version a pulse every $pi/12$ radial was generated giving a higher resolution. Finally, the car's suspension is replaced with stiff turnbuckle rods to eliminate the degrees of freedom of roll and pitch, to meet the assumptions of the Bicycle model.

3.1.1 MoCap

Usually a MoCap system can't be regarded as an 'onboard sensor',seeing as it requires an external area with set up with camera's. However in our case it was used as a replacement for a gps sensor. A gps sensor would be less desirable in our situation as the accuracy doesn't scale down with the car. Moreover a MoCap system enables us to do testing inside. Something which is troubling at best when tried with a GPS system. Moreover, most of the MoCap system was delivered as is. Which meant all of the data would be presented in ROS. And the sampling rate of the mocap would be set in stone at 120HZ.

3.2 Data acquisition

Data acquisition was done using a ROS (Robot Operating System) based system. The sensors were read using a Teensy 3.6 protyping board running a custom ROSserial node, connected to a Raspberry pi 3b running ROS. The custom rosserial node uses an interrupt controlled programming to gather sensor data in a fast and accurate way. The other option would be polling based programming which always lacks accuracy as the data that gets timestamped is always older than that point in time. Moreover, on board hardware timers were employed as pulse counters for the tacho signal. Handling these pulses as interrupts would create the possibility of overflowing the microcontroller with interrupts, as signals of more than 1500hz are within the realm of possibilites. Using dedicated hardware for this would provide a neater solution. The MoCap connected to the system utilizig the labs Wi-Fi network. Further communication was done using a self written ROS package. After data collection further processing was done using Matlab.Figure 2 further eleborates the Data acquisition framework.

3.2.1 ROS

ROS was chosen as basis for the system for 4 reasons. First, there was already a lot of experience in the research team using ROS which made it an easy choice to use as a basis for this testbed. One key component is timestamping. To provide usable data the testbed needs a reliable timestamping method. ROS automatically syncs all the clocks on every device providing an easy 'out-of-the-box'solution to this problem. Third the MoCap generated it's data already in ROS. Which meant combining everything in ROS would be a simplification. Last ROS offers really easy Data recording options called ROSbags. These can be easily imported in to MATLAB to do further analysis.

3.2.2 Sampling rate

Seeing as the MoCap already delivered her data at a sampling rate of 120hz. It was easy to pick this as the target sampling rate for the whole system. Seeing as initial tests showed no accelerations beyond 1.5g, this would mean the maximum increase of velocity per timestep would be 0.122[m/s]. This was deemed accurate enough for our tests. Moreover testing will also be done at speeds as slow as 1.5[m/s] including accelerating from standstill. With 24 magnets this would result in signals of 127hz. If our sampling frequency would be higher than the frequency of wheel pulses this would generate multiple points with the same distance traveled (as no new pulses from the wheel were encountered inbetween) increasing signal analysis dificulty requiring further interpolating before analysis can be done.

3.3 Experiments

The experiments that we conducted can be classified into three groups. The first set of experiments are the ones on the straight (longitudinal motion). During these we accelerated and braked while driving straight ahead. The second set are the steady state cornering experiments (lateral motion). Steady state cornering means cornering at a constant longitudinal velocity and constant steering angle. The first group of experiments focuses on longitudinal forces and slip ratios, while the second group focuses on lateral forces and slip angles. The tests were separated in order to distinguish longitudinal and lateral motion (Recommendation Barys Shyrokau). Finally, the third set of experiments are of combined motion.

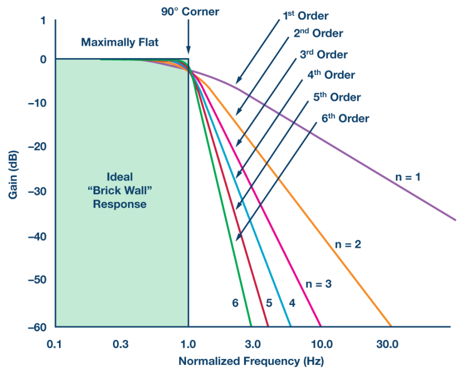


Figure 3. Brick wall response

$$
\begin{cases}
s_{wi} = 1 - \dfrac{u_{wi}}{R_w \omega_{wi}}, & u_{wi} \le R_w \omega_{wi}, \\
s_{wi} = \dfrac{R_w \omega_{wi}}{u_{wi}} - 1, & u_{wi} > R_w \omega_{wi},
\end{cases}
$$

Figure 4. Longitudinal slip

These tests take both lateral and longitudinal motion into account. Variables of the tests are acceleration/deceleration for longitudinal motion, longitudinal velocity and steering angle for lateral motion, and these three combined for combined motion. The variables were slightly increased each experiment in order to determine the 'borderline'between linear and nonlinear behaviour is.

3.4 Data processing

The experimental data is collected in rosbags and processed afterwards. This makes it possible to filter the noise from the IMU with a Zero-Phase Low Pass Butterworth Filter. This is a non-causal filter with a phase slope of zero. This eliminates any delay, commonly caused by causal filters. The Low Pass filter is of the 20th order, in order to approach an ideal 'Brick wall Response'.

3.5 Longitudinal slip

Differentiating the Mocap Data creates a lot of noise. Integrating the IMU data results in a much more stable velocity signal. This signal combined with the data from the Hall Effect Sensors is used to calculate the Longitudinal slip ratio using the following formulas:

3.6 Forces

The forces, acting along the y-axis of the body fixed frame of the car, are calculated using the equations of motion. As mentioned earlier there are three types of experiments that were conducted. This makes it easier to obtain these forces.

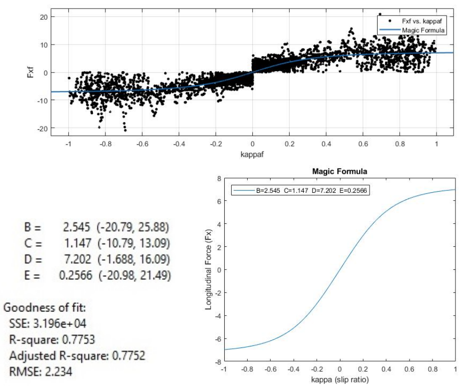
Where the forces on the wheels are obtained by formula

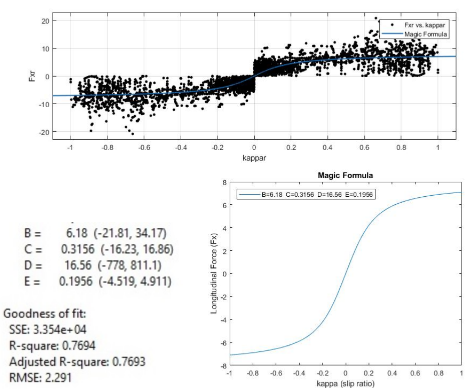
$$m(\dot{u} - vr) = \sum F_{xi} \quad (i = f, r),$$

$$m(\dot{v} + ur) = \sum F_{yi} \quad (i = f, r),$$

$$I_z \dot{r} = F_{yf} \cdot a - F_{yr} \cdot b,$$

Figure 5. equations of motion





3.7 Curve Fitting

After obtaining the forces, slip ratio and slip angles, from multiple bags, the data is curve fitted in order to obtain the 4 parameters from the magic formula.

4 Results and discussion

As noticeable from the pictures 4 and 4, the Magic Formula fits the points pretty descend. The accuracy of the fits is around R-square: 0.77. Furthermore, both the plots of the front and rear tires look similar. This had to be this case because we used the same tires for both front and rear. Even though the R-square of 0.77 is pretty good, there are still some minor errors in the plot. First of all, the Magic Formula has a peak and descend afterwards. This is not the case with our plots. It converges to a asymptote. Secondly, most magic formulas ascend much faster than our Magic Formula. There are a few possibilities why our Magic Formula differs from others. A possibility can be the data given by the revcounters. This data contains a lot of zeros because the sampling time of ROS is much faster than the sampling time of the Hall sensors. Because of this, interpolation between non-zero values is needed. This can result in inaccurate data.

Another possibility is due to noise generated by our sensors: Looking at figure [REFERENCE] we see a lot of noise generated in our plots compared to plots generated from other tests [misschien een Refererence]. There are two possible explanations to this noise. First is the RC car we used. As the aim at first was to improve the data acquisition software, not much was changed about the car itself. It wasnâ€™t after much testing was done it became clear there were faults in the hardware design. One of the big problems discovered during testing was that having only two magnets per wheel generated a lot of zeros of distance traveled at our sampling rate of 120hz. This let to a need of a lot of signal filtering. Therefore wheels which were fitted with 24 magnets were created, however thusfore no testing with those have been done.

5 Conclusion