Model-Based Design of a Winning Competition Car at the Freescale Cup

Richard Balogh and Marek Lászlo
Slovak University of Technology
Faculty of Electrical Engineering and Information Technology / URPI, D-511 Ilkovičova 3, Bratislava, SK-812 19, Slovak Republic
balogh @ elf.stuba.sk

Abstract— The paper shows how students used the modeling and simulation capabilities of the Matlab/Simulink to improve the control design of their winning FEIminetors car for the worldwide known Freescale Cup competition. Creating and simulating the model gives a) better understanding of the processes and b) almost bug-less transfer of the code to the embedded processor and c) first estimation of controller parameters.

I. INTRODUCTION

The Freescale Cup, is a global competition where student teams build, program and race an intelligent autonomous model car around a track. The fastest car to complete the track without going off the track wins the race. Total number of 75 students, from 25 teams representing their respective universities from 11 European countries raced their cars on the 2014 Freescale Cup track at Fraunhofer IIS in Erlangen, Germany. The 180 sq/m racetrack consisted of speed bumps, intersections, hills and chicane curves.

Fig. 1. FEI-minetors team at the Freescale Cup Competition with their car.

Winners of the Freescale Cup EMEA competition, the FEI-minetors team from the Slovak University of Technology in Bratislava were not a newcomers in this competition. During the initial testing and programming of the car, they recognized the need for better understanding of various parameters and car properties. To be faster than any other car, they require to understand not only the basic physics beyond the car and the properties of the proposed controller, but they also need to know how certain parameter influences another, which were the most important factors etc.

As the students of the STU in Bratislava are using the Matlab/Simulink during the academic courses, it was a natural choice to use this tool also for modelling the car for the Freescale Cup. As an illustration, we will show two important models we created for the competition.

II. ELECTRONIC DIFFERENTIAL

The car contains two DC motors in the Ackermann steering geometry chassis, so it was necessary to implement so called electronic differential to safely drive all the curves in high speeds. Its function is to modify the speed of inner and outer wheels according the steering angle. For better understanding of its function and for easier implementation of the function in the embedded microcontroller, model of the steering geometry was created. It includes lot of parameters, starting from geometry (dimensions of the wheels, radius, length of the axis), including motor properties (speed, torque) and also some others.

```matlab
% wheel driving radius
R = wheel_base/tand(angle);
% peripheral speed
omegav = v(11)/R;
% wheel speed
vR = omegav*(R-(track/2));
vL = omegav*(R+(track/2));
t_ratio_r = 1-(v(11)/vR);
t_ratio_l = 1-(v(11)/vL);
% calculate differential values
diff_r = t/2 * t_ratio_r *
(exp(c*v(11)));
diff_l = t/2 * t_ratio_l *
(exp(c*v(11)));
% calculate each wheel torque
tR = t/2 + diff_r * k;
tL = t/2 + diff_l * k;
```

Part of the code for calculation of the torques.
Resulted differential was simulated and later implemented in the C programming language. See the Fig 2. for the 3D chart of the Speed/Torque and Steering Angle for both left and right wheels. Later the model was modified based on real tests and empiric knowledge.

Fig. 2. Ackerman drive model and 3D chart of the electronic differential. Two planes represent the respective speeds of the rear left and right wheels.

III. CONTROLLER DESIGN

For modelling, simulations and testing much more complicated model was created (see Fig. 5). In Simulink, it was quite easy to start with modelling subsystems (car geometry, DC motor, controller, etc.) and then integrate them into the one complex layered model. We started with the standard text-book model of the DC motor with some measured and some more empiric parameters (see Fig. 3).

Fig. 3. Model of one of the DC motors.

Later we added also model of the car chassis including its weight, dimensions, etc. – see Fig. 4. From the beginning, it shows that considering model non-linearities is crucial for good correspondence of the model with reality. In each step, the parameters were adjusted to obtain real results.

Fig. 4. Model of the car for the controller design purpose.

After the subsystems testing and comparing with real system, all was combined into the one complex model (see Fig. 5). The main controller of the system was tested and parameters were modified many times based on results of the simulations. Especially the current spikes were observed and the final design of the power stage electronics takes this into the account. Later, based on the real measurements and observations, further modifications of the model was included. Finally, the controller code was almost without changes transferred to the microcontroller. In the future, we plan to use the Matlab embedded coder for the Freescale Freedom Board platform, which we are just testing.

Later we studied the influence of non-linearities in the system and controllers behaviours without and with non-linear parts in the model were compared.

Fig. 5. Study of the noise influence on the proper controller operation.

Fig. 6. The car during the final race. (Photo: Andrej Lenčucha)
IV. CONCLUSION

The paper shows how students used the modeling and simulation capabilities of the Matlab/Simulink to improve the control design of their winning FEIminetors car for the worldwide known Freescale Cup competition. Creating and simulating the model gives a) better understanding of the processes and b) almost bug-less transfer of the code to the embedded processor and c) first estimation of controller parameters.