9th International Workshop on Research and Education in Mechatronics
September 18th-19th 2008, Bergamo, Italy

REAL-TIME SIMULATION OF A RACING CAR

Alessandro Tasora[1]

[1]*Università degli Studi di Parma, Dipartimento di Ingegneria Industriale, Parma, Italy*

Abstract: This work describes the development and the application of a computer-based vehicle simulator. Such real-time simulation software is currently used in our Department as a tool for optimizing the design of a racing car, namely a single-seat vehicle for the SAE-ATA formula. Being based on our custom multi-body library Chrono::Engine, the vehicle simulator exploits a recent formulation founded on vector-measure differential inclusions and cone-complementarity problems (CCP).

Key words: Vehicle dynamics, multibody, tire model, visualization.

I INTRODUCTION

During the last years many car manufacturers looked at multibody computer simulation as a viable tool to test design solutions and parameters, before expensive prototypes are built [7]. The possibility of testing various car settings in a virtual environment is highly welcome when developing racing cars [3], where deadlines and cost constraints may be so tight that it would be unpraticable to test all parameters by building preliminary prototypes.

This is especially the case of the racing car which is currently under development at our University, namely the PR02 vehicle of the PR4300 student team. The PR02 vehicle will compete to the student Formula SAE-ATA this year, for the first time [1] with a working vehicle. The Formula SAE is a competition that challenges teams of university students to conceive, design, fabricate and compete with small, formula style, autocross racing cars [4].

A significant application has been presented by AUDI, which performed race optimisation of its car for the Le Mans 24 hours competition [5]. In that case, commercial multibody software (ADAMS) has been used to simulate entire laps, by introducing a synthetic model of driver wihch acts on throttle and steering. Usually, computational resources for this kind of simulations are quite high because of the many issues to be dealt: friction models, nonlinear motions, aerodynamical effects and so on; most of these factors are major sources of nonlinearities, thus complicating the numerical integration of the system.

If the simulation could happen in real-time, there is no need to introduce heuristic models of drivers because the user can interact with a three-dimensional visualization of the road using computer interfaces (pedals, hand wheel, or mouse and joystick in simpler cases), thus obtaining a man-in-the-loop simulation.

However the simulation of vehicles in real-time has always been one of the most challenging issues in multibody system dynamics: difficulties come both from the problem of providing a fast three-dimensional visualization of the environment, and from the fact that the multibody simulation can be highly CPU intensive. Thank to recent graphics hardware the former problem is not a substantial issue anymore, however the time-integration of the vehicle model can still demand high computational resources.

For the previous reasons we developed a in-house vehicle simulator based on our custom multibody library Chrono::Engine [8]. Such library features a recent formulation based on vector-measure differential inclusions and cone-complementarity problems (CCP) [2]: among the many advantages, there is the fact that the method is completely matrix-less and it runs faster than other commercial multibody packages [1].

Since the solver is very efficient, we can perform the time integration of a car model with 78 degrees of freedom whereas other approaches usually need to introduce simplifications or workarounds in order to cope with the real-time constraints (on average, our simulations run within a 1ms timestep period on a simple laptop).

The car model features four spring-damper articulated systems, 20 rods for the suspensions, a steer mechanism, plus other constraints for wheel spindles and for the differential. The algorithm for the tire-ground contact supports also the case of uneven roads, and can support different friction models [6]. Thank to the CCP formulation, the nonlinear effects caused by stick-slip phenomena does not require the adoption of small timesteps or stiff integrators.

For the visualization we used the Irrlicht library, which is a wrapper for the OpenGL and DirectX rendering systems. The user can see the road from the car

[1]The team already partecipated to a SAE-ATA competition for Design Review Category, last year, obtaining a succesfull 2nd place with its car PR01.

or from other point of views in a full-screen visualization (see Fig.4).

When requested, the simulator can record variables such as accelerations, speeds, reaction forces in struts, wheel slip angle, instant camber and caster, motor torque and many other useful graphs. After the real-time simulation, such recorded data can help in choosing the best settings for the car, for example in terms of gear ratios, suspension geometry and stiffness.

A racing car based on the results computed by this simulator is under construction at our University and will soon partecipate to its first Formula SAE-ATA challenge.

II Vehicle design

Although most vehicles challenging in the SAE formula are based on conventional frame design based on joined steel tubes, the frame of the PR02 car is built with state-of-the-art composite technology (see Figure 1). The honeycomb / carbon fiber frame offers superior stiffness at a lower weight; however it is of vital importance to know in advance the forces acting on the suspensions because, for economical reasons, even the smallest damages cannot be tolerated. Also, late modifications to the composite frame are less likely to be possible, when compared to a traditional steel frame. For these reasons, numerical simulations by means of multibody have been used to know in advance the type of stresses acting on the frame and the optimal geometry of the suspensions. Of course we still provided a set of adjustable struts to allow the fine tuning of the suspensions after the building of the truss, but the composite structure would not allow major rethinking. Hence the need of multibody simulations.

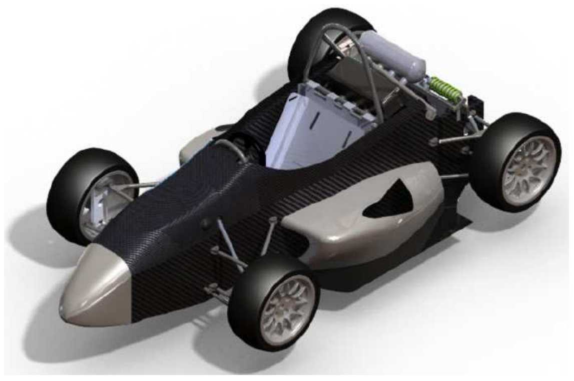
Figure 1: The PR02 racing car of the PR43100 team

The suspensions of the car are based on in-body coilovers featuring coaxial spring and dampers, which

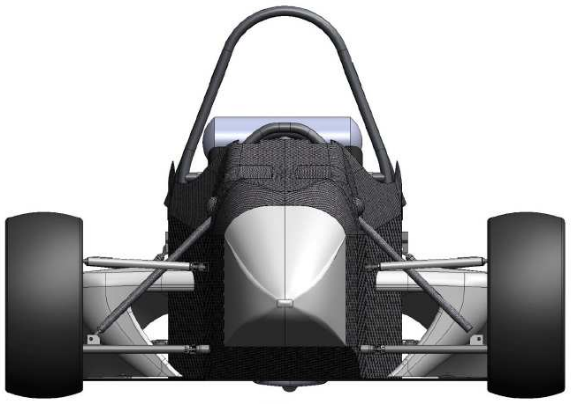
Figure 2: Front view of the PR02 veicle, showing the geometry of the suspensions

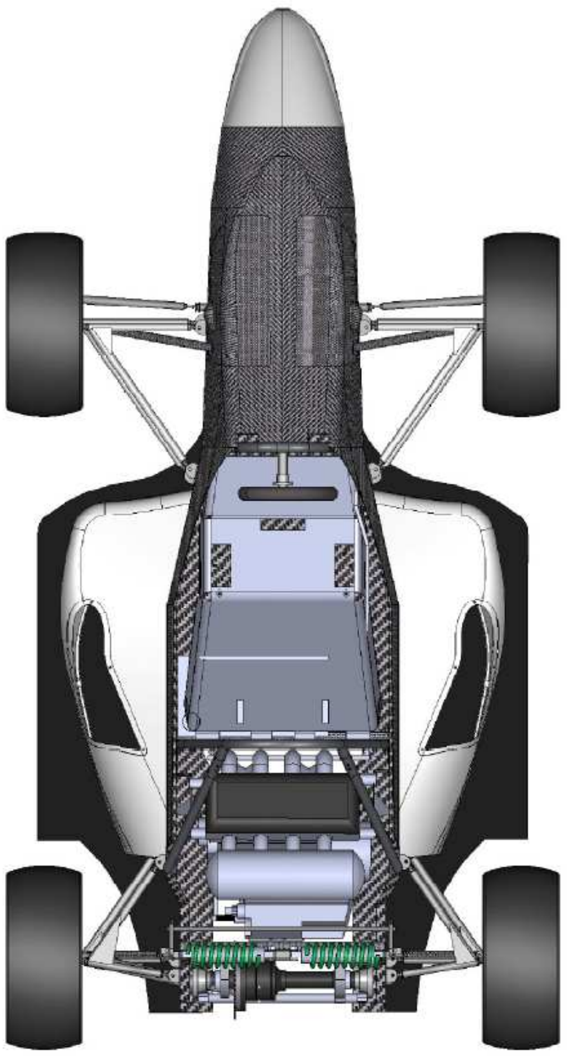
Figure 3: Top view of the veicle. Coilovers and rockers of the anterior suspensions are hidden

are connected to the push rods by means of triangular rockers. The vertical forces acting on the wheel will mostly flow through the push rod, triangle rocker, coilover and finally into the structure of the frame (see Figure 2). This is a part of the car design that required a significant amout of multibody simulations, to find the optimal alignment of the rods and the rocker so that a nonlinear stiffening behaviour of the system could be achieved.

Steering is actuated by means of a classical rack-pinion mechanism; two cardan joints are needed to raise the shaft of the steering wheel up to an ergonomical position. Also the geometry of the steering articulated mechanism has been optimized to have a precise control over toe-in in all circumstances.

Unsprung masses are based on commercial aftermarket components, brakes are 220 mm discs from Brembo, and wheels are OZ light alloy 13”x6,5”’.

The engine is a modified version of a Suzuki GSX 600 motor, with EFI electronics. Transmissions flows through a chain and a Torsen differential (see Figure 3).

III REALTIME SIMULATOR

In order to obtain a real-time simulation of the vehicle, with man-in-the-loop control, the time integration must be computed really quickly: each time step should not require more than 0.001 s of computational wall-clock time (in some cases, even smaller time steps may be required). For this reason of efficiency, we developed the simulation software as a C++ application.

The executable is dynamically linked to our multibody simulation library, Chrono::Engine [8], which offers more than one thousand of ready-to-use C++ functions and data structures for the creation and simulation of mechanical systems. Basically, this library allows the creation of unlimited rigid bodies in 3D space: those parts can be constrained by joints (selectable among a vaste set of holonomic, rheonomic, scleronomic constraints) or spring-dampers. Hence complex mechanical systems with whatever topology can be described by C++ language statements. For instance, the creation of a rigid body that represent the car truss can be obtained with statements similar to the following:

```
ChSystem physical_system;
ChSharedBodyPtr  truss(new ChBody);
 truss->SetMass(69.1);
 truss->SetPos( ChVector<>(0, 0.016 ,1.99) );
 truss->SetRot( ChQuaternion<>(QUNIT) );
 truss->SetInertiaXX(ChVector<>(4.8,4.5,1));
 truss->SetBodyFixed(false);
physical_system.AddBody(truss);
```

Constraints between the parts can be created with specific statements which define the type, the position, the alignment and the pair of connected parts, as in the following example that creates a revolute joint:

```
ChSharedPtr<ChLinkLockRevolute>
            my_link(new ChLinkLockRevolute);
my_link->Initialize(truss,
                     rocker,
                     ChCoordsys<>(ChVector<>(1,2,0)),
                     ChCoordsys<>(ChVector<>(0,1,0)));
physical_system.AddBody(my_link);
```

All created items are stored in form of transient database into a container object of class `ChSystem`, which is also responsible of assemblying the systems and performing the simulation.

Different integration schemes have been tested, the one which proved to be faster and more robust is based on measure differential inclusions and is described in [2].

For the visualization we used the Irrlicht library, which is a wrapper for the OpenGL and DirectX rendering systems. Since it would be prohibitive to refresh the screen at each integration step, we implemented the main simulation loop in a way that the 3D framebuffer refresh happens only each 1/60th of second, while the integration steps are performed with higher frequency. Note that OpenGL and DirectX offer a feature to force the refresh at each 1/60th of second (the so called *vertical blanking signal*): this helps achieving a good determinism even on Windows platforms, even if not exactly a true hard-real-time with guaranteed periodicity as it could be obtained on special operating systems.

The user can see the road from the car or from other point of views in a full-screen visualization (see Fig.4). The car can be driven with controls over steering, throttle, braking and gear. A synthetic sound is generated too, so that the user can feel also the noise of the engine.

IV THE MULTIBODY MODEL OF THE VEHICLE

To avoid wasting computational resources with unnecessary details, not all moving parts of the vehicle have been modeled as separate bodies. For example, we adopted a simplified model of the gear train because we were less interested in studying subtle details of the transmission. On the other hand, we did not want to use over-simplified models (such as those used in many real-time car simulators). We experienced that, with our multibody technology, a good tradeoff between level of detail and computational throughput is

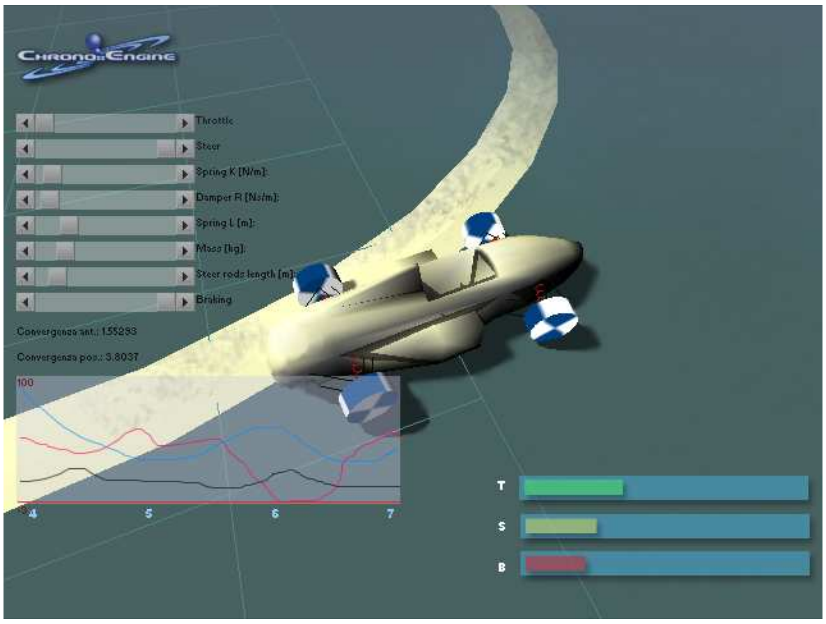

Figure 4: A snapshot from the main screen of the real-time car simulator

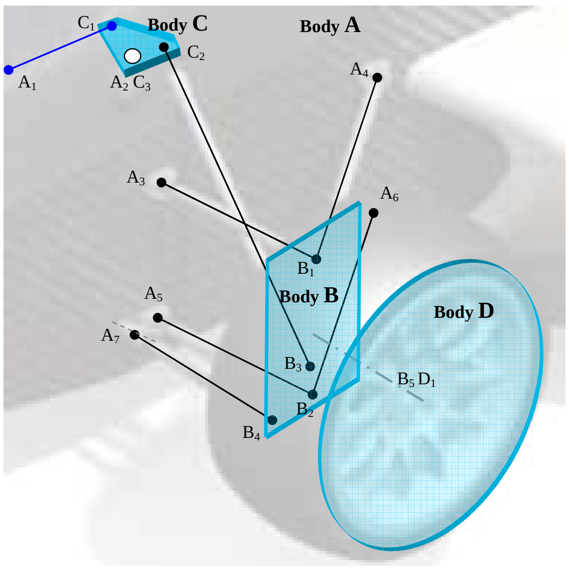

Figure 5: A simplified representation showing parts and constraints introduced in the multibody dynamical model

a model which considers 13 rigid bodies per vehicle. The following is a list of the 13 parts, for a total of 78 degrees of freedom which are then reduced to 14 after the application of various kinematic pairs:

- car truss,
- front left wheel
- front left hub
- front left rocker
- front right wheel
- front right hub
- front right rocker
- rear left wheel
- rear left hub
- rear left rocker
- rear right wheel
- rear right hub
- rear right rocker

In Figure 5 one can see a schematic representation of the front left suspension.

Note that the previous list do not include the A-arms for the suspensions, neither the rods of the steering. This is a possible because in Chrono::Engine there is a special constraints of type `ChLinkDistance` which can be used to simulate a massless rod (see the black lines between points A and B in Figure 5).

Since we were not interested in the inertial effects of the A-arms (which are very lightweight) this proved to be an efficient way to model the 20 rods, 16 for the A-arms of the 4 suspensions and 4 for the steering rods. Two of the steering rods are fixed, for the rear axle, and the two anterior rods can move their endpoints along the line of the rack, depending on how the user acts on the steer wheel.

Note that a digital filter is applied to the steering control coming from the user in order to avoid unnatural sudden changes of steering (in fact if the mouse or similar cheap devices are used to steer, the control signal may be aliased or unrealistically discontinuous like a stairstep).

The rockers are connected to the truss by means of revolute joints. The same type of joint is used to connect wheels and hubs.

Objects of type `ChLinkSpringDamper`, connecting the rockers and the truss, are used to simulate the coilovers; these also implement upper and lower limits on the displacement.

Other constraints are used to simulate the differential and the brakes. The motor is simulated using the torque-speed curve and the gear ratios of the adopted Suzuki engine.

The contact between the wheel and the surface is implemented with a model which can take into account uneven pavements, thank to a state-of-the-art collision detection engine.

V RESULTS

Many simulations have been simulated with different purposes. From time to time, parameters, sizing and weights of the car have been changed until we converged to a satisfactory design.

Following are the most relevant results which we obtained by using this simulator.

- The alignement and position of rocker, push rod and coilover have been optimized so that an hardening effect is obtained in the suspension, without making the suspension too stiff during non-critical trajectories. This required also quasi-static analysis.

- Simulations helped to choose the stiffness and the damping coefficient of the four coilovers. A virtual track with few bumps has been created to this end. In Fig.11 one can see the outcome of one of these simulations in terms of spring force.

- Kinematic and dynamical simulations showed what happens in terms of suspension geometry when the car turns, jumps, accelerates, brakes, etc. This helped choosing proper default values of toe-in angle, caster angle, camber angle (Fig.6). The user interface features some controls which can be used to adjust in real-time these parameters, so it is immediate to see the effects on the handling of the vehicle.

- Reactions on the A-arms and push rods have been recorded in graphs such as those of Fig.**??** and Fig.9, during various dynamic conditions (brake, acceleration, bump, turn). Results have been used to perform a fatigue analysis of the parts, using the Rainflow method and a FEM software, in sake of the best compromise between structural safety and light weight.

Length	2680 mm
Wheelbase	1600 mm
Track	1220 mm
Ride height	45 mm
Static camber, front	-1.0deg
Static camber, rear	-0.5deg
Front caster	2.57deg
Roll center height	28 mm

Table 1: Main parameters of the PR02 car.

Some of the parameters of the final design are summed up in Table 1; further details cannot be given either for reasons of space and for secretness.

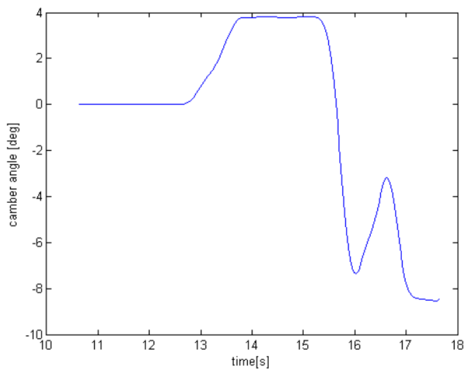

Figure 6: Change in camber of the front wheels, during a test manouver that causes large deflections in suspensions

VI CONCLUSIONS

A custom vehicle simulator, completely developed in-house, has been used to perform kinematic and dynamical analysis of the PR02 racing car. The car is represented by a multibody model composed by 13 parts and 42 constraints.

High performance computing and visualization solutions have been used to allow man-in-the-loop control of the simulated vehicle.

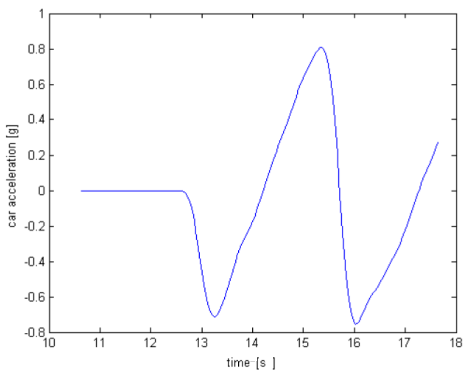

Figure 7: Output graph showing lateral acceleration during a simulated maneuver

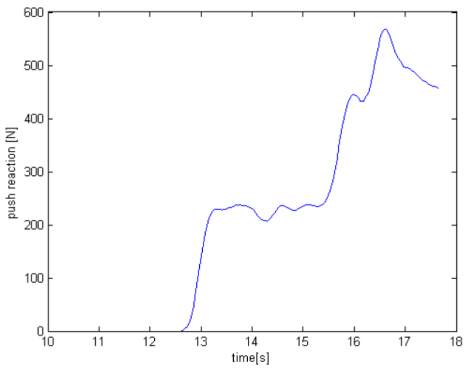

Figure 8: Example of output graph: reaction force in one of the front push rods during a simulated trajectory

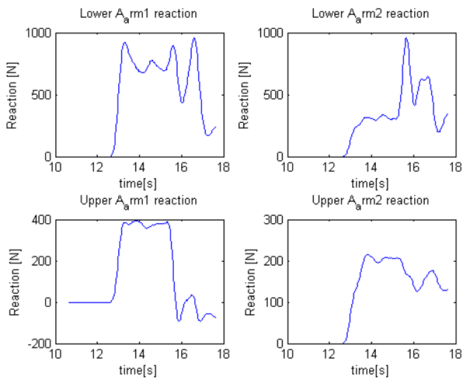

Figure 9: Example of output graph: reactions in the suspension arms on the front-left quater of car

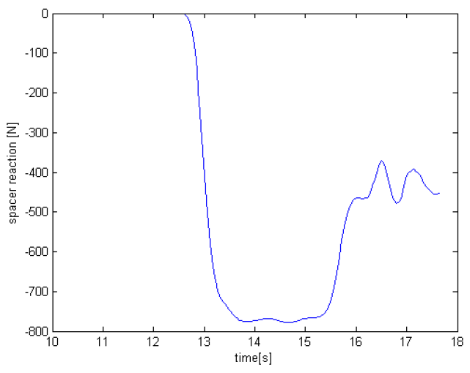

Figure 10: Reaction on the wheel spacer

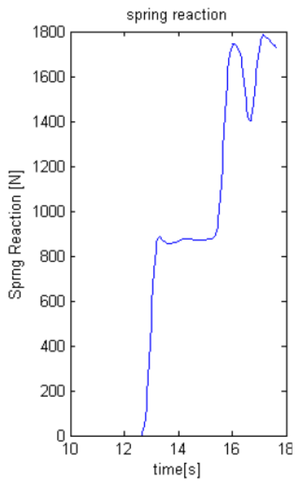

Figure 11: Force from the spring of the coilover, during a simulated trajectory

This tool helped in optimizing the design of the car, which has been recently built and prepared for the SAE-ATA championship.

ACKNOWLEDGEMENTS

The Author thanks the PR43100 team-leader Marcello Alfieri for helping with the development and for providing data. Also, the Author thanks Francesco Bacchieri, Massimo Bercella, Nicola Buttini, Marco Caffarra, Salvatore La Malfa, Alessandro Mistrali, Maurizio Pinzoni, Riccardo Viola and all the students of the PR43100 team for the fruitful collaboration on this and other related topics.

REFERENCES

[1] Paolo Righettini Alessandro Tasora, Marco Silvestri. Architecture of the chrono::engine physics simulation middleware. In *Proceedings of Multibody Dynamics 2007, ECCOMAS thematic conference*, Milano, Italy, June 2007.

[2] Mihai Anitescu and Alessandro Tasora. An iterative approach for cone complementarity problems for nonsmooth dynamics. Technical Report ANL/MCS-P1413-0507, Argonne

National Laboratory, May 2007. Available online from http://www.mcs.anl.gov/ anitescu/PUBLICATIONS.

[3] Mike Blundell and Damian Harty. *The Multibody Systems Approach to Vehicle Dynamics*. Butterworth-Heinemann, Oxford, 2004.

[4] Associazione Tecnica dell'Automobile. Formula ata web page, 2008.

[5] N. Muller M. Muhlmeier. Optimisation of the driving line of a race track. *AutoTechnology*, 2, 2003.

[6] Hans Pacejka. *Tyre and Vehicle Dynamics*. Butterworth-Heinemann, Oxford, 2005.

[7] Martin Muhlmeier Stefano Pieri, Carlo Poloni. Integrating multibody simulation and cfd: toward complex multidisciplinary design optimization. *JSME International Journal Series*, 48(2):224–228, 2005.

[8] A. Tasora. Chrono::engine project, web page, 2006.