ECCOMAS Thematic Conference on Multibody Dynamics
June 29 - July 2, 2015, Barcelona, Catalonia, Spain

# Development, Implementation and Validation of a Hard Real-Time Multibody Simulation For High-Fidelity Steering Wheel Force Feedback

Roland Pastorino*#, Michiel Desloovere#, Fredriek Vanneste#, Pascal Degezelle†, Wim Desmet#

* Laboratorio de Ingeniería Mecánica
Universidad de La Coruña
c/ Mendizábal s/n, 15403 Ferrol, Spain
rpastorino@udc.es

# PMA division
Department of Mechanical Engineering, KU Leuven
Celestijnenlaan 300b, 3001 Leuven, Belgium
(roland.pastorino, wim.desmet)@mech.kuleuven.be

† Optidrive N.V.
Wijgmaalsesteenweg 108, 3012 Leuven (Wilsele), Belgium
pascal.degezelle@optidrive.be

## Introduction

High-fidelity driving simulators are commonly based on complex real-time multibody (MB) models [1]. They are particularly well suited to simulate accurately the vehicle dynamics. Additionally they can also be used to simulate the steering mechanism of the vehicle. Including both the former and the latter in the MB simulation allows for the development of a steering wheel force feedback system. Few such systems based on MB analysis have been developed [2]. This paper presents the development, the implementation and the validation of a hard real-time forward-dynamics multibody simulation for high-fidelity steering wheel force feedback. To develop such an application, the MB model has to be simulated in real-time. This means that several requirements have to be fulfilled. First the algorithms used in the simulation as well as its numerical implementation must be computationally efficient. To achieve this, it is crucial to *a*) choose an efficient MB formulation, *b*) select a stable and accurate integrator, *c*) implement efficient matrix computations, *d*) take advantage of the hardware and compiler features that can speed-up the computations, *e*) select the electronic hardware (CPU, RAM memory, etc) and configure it in order to avoid bottlenecks, *f*) select an efficient operating systems (if any). Then the simulation as well as the electronic hardware used in the simulator must have a deterministic behavior. Driving simulators are Hardware-in-The-Loop (HIL) and Human-In-The-Loop (HITL) systems in which a digital simulation interacts continuously with the analog environment through sensors and actuators. This interaction must be made at regular time intervals with minimal time delays. A loss of time synchronism causes delays in the sensor data sampling and the actuator control, resulting in simulation precision losses as well as control instabilities. For these reasons it is also crucial to *a*) choose a fixed time step integrator, *b*) limit the maximum number of iterations if an implicit integrator is used, *c*) avoid any non-deterministic programming techniques, *d*) use a Real-Time Operating System (RTOS) if any, *e*) sample sensor data and control actuators deterministically and *f*) use deterministic real-time network communication protocols.

## Driving Simulator and Hard Real-Time Multibody Simulation

A driving simulator with steering wheel force feedback has been built (see Figure 1). It consists of a chassis, a seat, a virtual-reality headset and a steering wheel equipped with an AC synchronous direct-drive motor for force feedback. A preliminary study has been carried out to evaluate the performances of geared and direct-drive motors in the frequency range 0-30Hz in order to assess whether a gearbox could be used. It was concluded that a direct-drive motor is the preferred solution. The simulation is executed on a real–time target which is a PC running `Xenomai 3` (a Linux-based RTOS). Both Xenomai cores (the one based on the `PREEMPT-RT` patch and the co-kernel approach) have been used and compared. Two different AC synchronous motors with their respective electric drives can be used on the simulator. Their maximum continuous and peak torques are 8.5 Nm and 15 Nm respectively. The real-time target can control the first one via `UDP` or `PROFINET` and the second one via `EtherCAT`. Both motors are controlled in torque. Multiple MB models have been developed ranging from a standing-still 2D model including the rack and pinion system, the tie rods, the knuckles and the tires up to a 3D model of the full-vehicle including the complete steering mechanism and the suspension. Figure 2 shows a graphical representation of a 3D MB model of the complete steering mechanism. As mentioned earlier, the computational efficiency of the simulation is crucial. For this reason, natural coordinates have

Figure 1: Driving simulator.

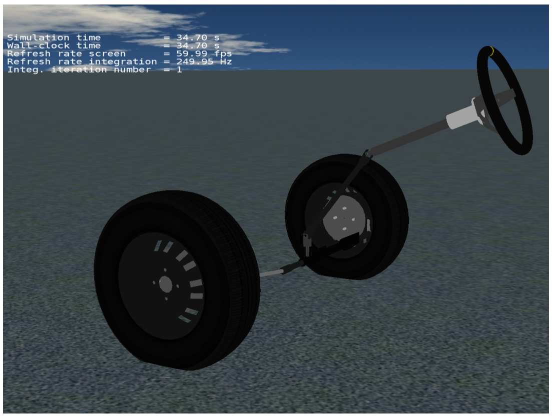


Figure 2: 3D steering mechanism

been employed to build the MB models. Then an index-3 augmented Lagrangian formulation with mass-damping-stiffness-orthogonal projections in velocities and accelerations has been chosen together with the generalized-$\alpha$ integrator and a fixed time step [3]. It is important to be able to add a certain amount of numerical damping as some noise from the steering wheel encoder will inevitably propagate into the MB simulation. For all these models the encoder angle is measured by the corresponding drive and sent to the real-time target via one of the real-time network protocols mentioned earlier. A Kalman filter computes the instantaneous angular velocity and acceleration using the encoder angle as sensor data and passes them all to the MB simulation. The steering wheel of the MB model is kinematically guided using this information. The steering wheel torque is retrieved from the constraint reaction force associated with the Lagrange multiplier of the kinematical guidance constraint [3]. It is then fed back to the drive via network and applied to the motor to close the control loop. The simulation runs at 1 kHz (i.e. $\Delta t = 1$ ms) such that it is not necessary to have a fast haptic loop and a slow simulation loop. The minimal code for each MB model was generated using the software developed in [4]. As this software allows to change both the programming language of the generated code and the libraries used for matrix computations, different implementations are compared: dense vs sparse matrices for example.

**Experimental Validation**

The parameters of the MB models have been taken from an available commercial car. A steering robot has been mounted on it and tests have been conducted in order to compare the simulation torque predictions with its experimental counterpart.

**Acknowledgments**

The authors gratefully acknowledged the support of *a*) the IWT Flanders within the MBSE and MODRIO projects *b*) the Research Fund KU Leuven *c*) ITEA2 through the MODRIO project *d*) Flanders Make strategic research centre and *e*) the Belgian Programme on Interuniversity Attraction Poles, initiated by the Belgian Federal Science Policy Office (DYSCO)

**References**

[1] A. Hoskins, M. El–Gindy. Technical report: Literature survey on driving simulator validation studies. International Journal of Heavy Vehicle Systems. Vol. 13, No. 3, pp. 241–252, 2006.

[2] T. Shiiba, W. Murata. Experimental validation of steering torque feedback simulator through vehicle running test. English. Journal of Mechanical Science and Technology. Vol. 23, No. 4, pp. 954–959, 2009.

[3] D. Dopico, F. González, J. Cuadrado, J. Kövecses. Determination of Holonomic and Non-Holonomic Constraint Reactions in an Index-3 Augmented Lagrangian Formulation with Velocity and Acceleration Projections. Journal of Computational and Nonlinear Dynamics. Vol. 9, No. 4, 2014.

[4] R. Pastorino, F. Cosco, F. Naets, J. Cuadrado, W. Desmet: Towards Real-time Multibody Simulations using ARM-based Embedded Systems. In The 3rd Joint International Conference on Multibody System Dynamics - The 7th Asian Conference on Multibody Dynamics, Busan, Korea, 2014.