WORKSHOP ON VIRTUAL REALITY IN PRODUCT ENGINEERING AND ROBOTICS: TECHNOLOGY AND APPLICATIONS
Special issue of: BULLETIN OF THE TRANSILVANIA UNIVERSITY OF BRAŞOV ISSN 1221-5872

APPLICATION OF REAL-TIME MULTIBODY DYNAMICS TO VEHICLE CONTROL

J. CUADRADO[*], M.A. NAYA, D. DOPICO, M. GONZALEZ

Abstract: *This paper reports on the development of a low-cost and efficient computational framework for the design of automobile motion controllers. The core elements of the tool are a real-time formalism for the dynamics of multibody systems, a virtual reality (VR) interface for human-in-the-loop simulation, and the fuzzy logic toolbox of Matlab. A detailed model of an existing prototype car has been implemented in Fortran language, and controllers have been designed for several maneuvers of the vehicle, in order to test the developed framework. It has been demonstrated that the dynamic formalism is fast enough for both the design and simulation stages, as well as robust, that the virtual reality interface is of great help during the control design process, and that the Matlab fuzzy logic algorithms can be efficiently connected to the Fortran-based computational model of the car.*

Keywords: *multibody dynamics, real-time simulation, vehicle dynamics, control design, human-in-the-loop simulation, virtual reality, driving simulator.*

1. Introduction

Our group has experience on rigid and flexible multibody dynamics and, particularly, has focused on the efficiency of the methods. Efficiency issues appear at different stages: modeling, formulation of the equations of motion, numerical integration, and implementation. All of them should be addressed in order to achieve really efficient formulations, able to provide real-time performance on customary personal computers for the increasingly complex and detailed multibody models required for industrial applications, like those with human- or hardware-in-the-loop.

As an industrially meaningful application of real-time multibody dynamics, we have addressed the design of vehicle controllers. Nowadays, modeling and simulation of vehicle dynamics plays a great role in the design and evaluation of vehicle control systems, since they reduce costly and time-consuming construction of prototypes and experimentation. Multibody simulation can be used for controller design, tuning and testing of electronic control units, optimum control, or onboard devices providing driver advice and/or actuation.

The multibody system model of the vehicle may play different relevant roles in the consideration of vehicle control [1, 2]. Many controllers employ, inside the control loop, a model of the vehicle which,

[*] Escuela Politecnica Superior, University of La Coruña.

in most cases, has been simplified to avoid nonlinearities, like the bicycle model [3]. A model is also used for controller tuning, either for partial control of the vehicle (braking, steering assistance, etc.) or for automatic driving [4, 5]. Of course, differences always exist between the vehicle model and its real counterpart. However, an accurate vehicle model will enable that, despite such differences, the additional controller adjustments required at the time of installing it onboard the vehicle are minimized or even null. This work is oriented in such direction, since it presents a tool for controller design based on a detailed vehicle model.

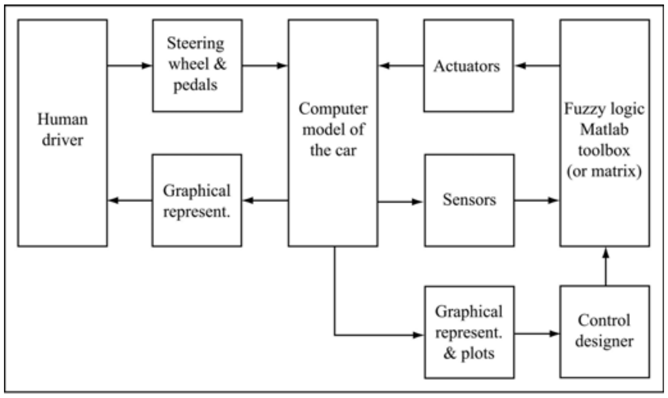
Human driver
Steering wheel & pedals
Graphical represent.
Computer model of the car
Actuators
Sensors
Fuzzy logic Matlab toolbox (or matrix)
Graphical represent. & plots
Control designer

Fig. 1. *Computational framework.*

This paper reports on the development of a low-cost and efficient computational framework for the design and evaluation of automobile motion controllers. The core elements of the tool are a real-time formalism for the dynamics of multibody systems, a virtual reality (VR) interface for human-in-the-loop simulation, and Matlab for the controllers.

The starting point of the work has been an existing prototype car, whose virtual counterpart has been implemented in Fortran language through a dynamic formulation developed by the authors [6]. This will make possible to experimentally validate, in a future work, the controllers designed through the use of the proposed environment. Virtual reality capabilities have been given to the program by means of a realistic graphical output (either on screen or HMD) and game-type driving peripherals (steering wheel and pedals), in order to enable participation of human drivers in the controller design and evaluation processes. Although the framework is independent on the control method selected, in this work fuzzy logic has been chosen for the design of the control algorithms, which have been implemented on the Matlab environment. Figure 1 illustrates the general structure of the computational tool developed.

The paper is organized as follows: Section 2 describes the computational model of the car; Section 3 focuses on the virtual reality features of the tool, and their role in the controller design and evaluation processes; Section 4 explains the structure of the control module; Section 5 shows two examples to illustrate the advantages of human-in-the-loop simulation for the design and evaluation of the controllers; Section 6 makes some efficiency remarks;

finally, Section 7 outlines the conclusions of the work.

2. Car model

The mathematical description of the car, illustrated in Fig. 2 along with the physical prototype, has been carried out in natural coordinates [7]: 44 points, 7 unit vectors, 5 distances and 5 angles have been used as problem variables, leading to a total problem size of 163. As it can be seen in Fig. 2, the coordinates have been used to model the chassis, the steering and suspension systems, and the wheels.

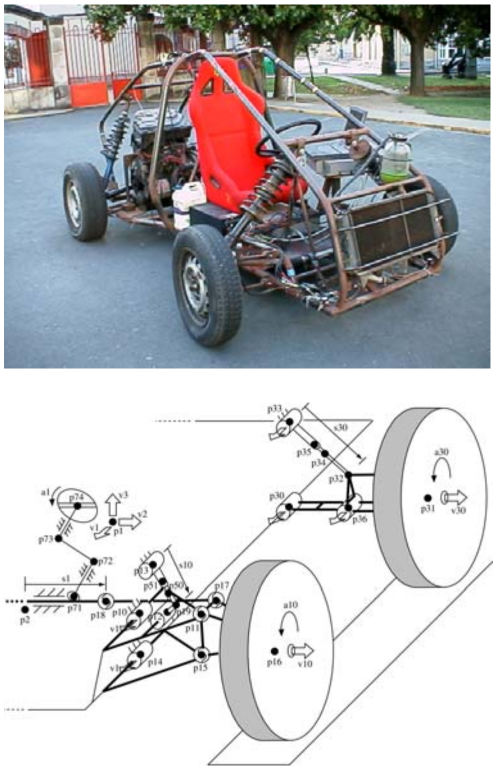

Fig. 2. *Prototype and model.*

The steering wheel is kinematically guided. Forces which deserve to be described are the following:

– The chassis inertial parameters have been obtained from a CAD model, in which the actual structural elements have been replicated. The engine inertial parameters have been experimentally estimated.
– Suspension forces: linear models of springs and dampers have been considered.
– Tire forces: lateral and longitudinal forces, as well as self-aligning torques, have been introduced through the *Magic Formula* [8] with coefficients provided by the tire maker.
– Power transmission forces: from the torque-speed engine relationships and the gear ratios, both provided by the engine maker, the automatic gearing has been modeled. The torque is applied on the rear wheels.
– Brake forces: the braking torque has been estimated from disk geometry, and applied to the four wheels.

For details on the formulation of the equations of motion of the multibody system, and the numerical procedure to integrate them along the time, the reader is referred to [6]. A code that calculates the dynamics of the described model according to the indicated formalism has been implemented in Fortran language, due to its high efficiency.

3. The VR interface

The computational model of the car has been used to create a driving simulator, shown in Fig. 3, by combining the already mentioned vehicle dynamics code, along with a realistic graphical output (received by the user either through the computer screen or through a HMD) and game-type driving peripherals (steering wheel and pedals). Both the gearing and the driver's point of view are operated from buttons on the wheel.

For the graphical output, the selected application programming interface (API) has been OpenGL, due to its portability and intuitive programming structure. Communication between the program and the driving peripherals has been managed with the library DirectInput from the API DirectX.

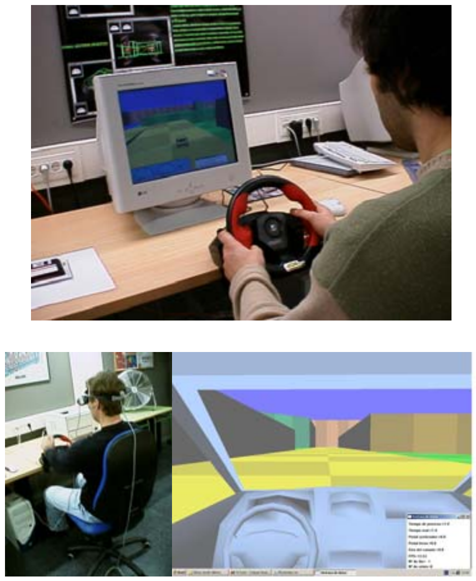

Fig. 3. *Driving simulator (screen, HMD).*

Such a virtual reality interface serves to several purposes. Prior to the control design stage, a human driver is expected to carry out the targeted maneuver on the simulator, in order to provide an initial assessment on maneuver feasibility and realistic values of the involved parameters: time required to perform the maneuver, maximum speed and acceleration for stability, more challenging parts of the maneuver, etc. Moreover, the performance achieved by the human driver may be used as a reference to measure the quality of the successive controllers obtained during the design stage, until the desired behavior is attained. Additionally, when the aim is to partially control the motion of the car, like in driver assistance applications (as opposed to cases in which the total automatic control of the car is pursued), both the human driver and the controller must simultaneously operate (see Fig. 1): this is also enabled by the virtual reality interface.

On the other hand, it must be pointed out that the graphical representation of the car motion, along with the plots provided by the program, are essential elements which allow the designer to tune the controller parameters until a satisfying result is achieved, as illustrated in Fig. 1.

4. The control module

As indicated in the Introduction, the controllers have been based on fuzzy logic, implemented by using the corresponding toolbox of Matlab. The use of fuzzy logic is justified as the system is highly nonlinear, a simplified model is not required, the driver's behavior can be reproduced, and a reliable Matlab toolbox is available.

The controllers receive data from virtual sensors defined in the computational model of the car, and act upon actuators, also defined in the virtual model, as illustrated in Fig. 1. For sensor and actuator definition, the objective has been to replicate as much as possible their counterparts in the actual prototype, so that the designed controllers may be tested in the future onboard the real car for their experimental validation. In the case of the actuators, this goal has been completely achieved: the controller acts upon the steering wheel and pedals (throttle and brake); the engine possesses automatic gearing, so that no actuation is needed upon the gear-lever. In what respects to the sensors, all the magnitudes required as inputs by the controllers can be derived from the information about position, velocity and acceleration of the car; while such information is directly provided by the computational model of the vehicle, it is not so straightforwardly obtained in the real case: an intermediate module for motion recovery from sensor data will be required for the experimental task.

The controller design process, included in Fig. 1, is iteratively carried out by the designer, based on the results achieved by each new design, which may be appraised by means of the graphical representation and plots provided by the tool.

It must be emphasized that the objective of this work is to show the applicability of a robust and easy-to-implement real-time formulation, to the development of a low-cost and efficient VR based computational framework for the design and evaluation of automobile controllers. Hence, the control method employed is not relevant.

In order to introduce the control in the prototype computer model, the Fortran code containing the dynamics of the car, and the Matlab functions implementing the fuzzy logic control algorithms, must be combined. To connect Fortran and Matlab, two alternatives have been investigated: Matlab Engine and MEX files. Both of them require compatibility between Matlab and the Fortran compiler. The investigation led to the conclusion that Matlab Engine is preferable, since it yields lower execution times.

5. Examples

To illustrate the use of the proposed tool, controllers have been designed for two particular maneuvers of the car. Fuzzy logic controllers have been developed employing Mamdani-type inference and its typical defuzzification process [9], finding the centroid of a two-dimensional function to determine the value of the output variable from its membership to the output fuzzy set.

In the first maneuver, shown in Fig. 4, the car starts from rest, covers a distance of 20 m following a straight line, and stops. The objective is to stop the car as close to the target point as possible. First, the maneuver was carried out by a human driver, through the already described VR interface. Based on the behavior of the car under human guidance, a maximum speed value of 5 m/s has been allowed, and an acceleration limit of 3 m/s^2 imposed. The time spent in the maneuver has not been limited.

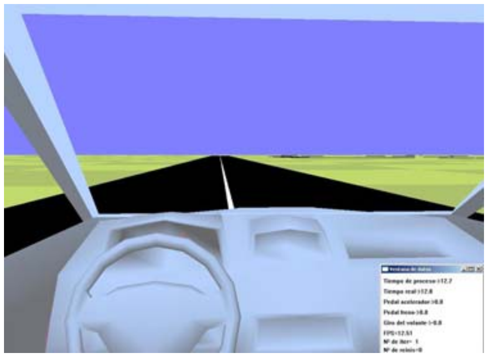

Fig. 4. *Straight path.*

It was observed that less time is employed by the controller, although velocities are kept smaller than in the human driven case, and acceleration and brake are not so severe. This result is due to a more constant actuation that avoids abruptness. The error attained by the controller is one order of magnitude lower than that achieved by the human driver.

The second example is an obstacle avoidance maneuver, illustrated in Fig. 5. Starting from rest, the car covers an initial straight path of 20 m, then follows a full period (from peak to peak) of a sinoidal path of amplitude 1.75 m and, finally, must return to the straight line. First, a human driver carried out the maneuver so as to determine realistic performance values for the maneuver. In such a way, a region of feasible velocities was established, which served to define the fuzzy sets for such magnitude. As a result, the speed has been kept below 8 m/s, and the acceleration has been restricted to the interval ±3 m/s^2, in order to avoid severe actions.

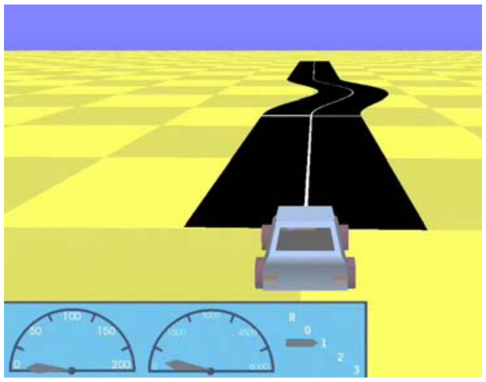

Fig. 5. *Obstacle avoidance.*

It was observed that, although the human driver reaches higher velocities, the time spent for the maneuver is similar to that required by the controller. Furthermore, the human driver commits higher errors too. Therefore, it can be affirmed that the vehicle follows the track more accurately when guided by the controller.

6. Efficiency remarks

The CPU-times obtained for the two maneuvers reported using Matlab Engine on a Pentium IV @ 2 GHz are listed in Table 1. CPU-times* include the time spent in opening a Matlab session, while CPU-times** are obtained when the running program is attached to an already opened Matlab session. It must be indicated that, provided no control is considered, the Fortran program comfortably achieves real-time performance when solving for the dynamics of the vehicle.

Table 1

Maneuver	1	2
Time (s)	12	24
CPU-time* (s)	18.00	36.14
CPU-time** (s)	11.99	30.45

Therefore, Matlab Engine represents a good solution when, as in the present case, only some parts of the program need to be executed in Matlab. Furthermore, it is adequate for the stage of controller tuning, since real-time is not required.

However, if the controllers are to be employed in stand alone applications that require real-time, the previous procedure is not a good option. Time delay is due to Fortran-Matlab connection, and the heavy evaluation of the controllers in Matlab. In order to find a solution, it must be taken into account that the fuzzy logic generates, by means of rules of membership and actuation, a hyper-surface which relates the input (error, velocity, etc.) and output variables (throttle-brake, steering). An evaluation of a matrix containing all the possible combinations of inputs of the controller can be carried out. Matlab provides function *evalfis* to this end. A matrix may be used as first input of the function, in such a way that each row of the matrix contains one of the possible inputs required by the controller. *Evalfis* function will return another matrix, whose rows are the corresponding outputs. If the controller is stored in this way, its later evaluation just consists of an interpolation of the output matrix.

Table 2 shows the CPU-times obtained with controllers stored in matrices, and loaded in the program as it starts. CPU-time* is now the time obtained with Matlab Engine attached to a Matlab previous session (the faster case presented above), and CPU-time[**] is the time consumed when the controllers are stored into matrices. As it can be seen, now real-time is comfortably achieved.

Table 2

Maneuver	1	2
Time (s)	12	24
CPU-time* (s)	11.99	30.45
CPU-time** (s)	4.87	11.75

In order to give an idea about the practical capabilities of the described framework, the two following applications have been developed:

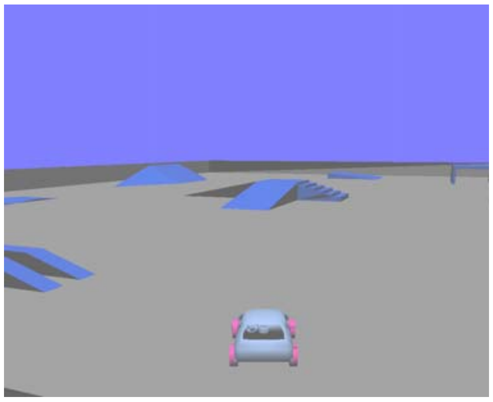
Fig. 6. *Demo application: robustness.*

a) The environment full of obstacles illustrated in Fig. 6. The aim of this application was to highlight the robustness of the dynamic formulation when dealing with a full model of a car (compared to that provided by the usual real-time explicit integrators). The result was that the model can be driven in real-time by a human driver through the different obstacles, i.e. undergoing highly violent motion, during very long simulation times. This means that the proposed framework can be used not only for smooth and easy maneuvers, but also for demanding and critical situations.

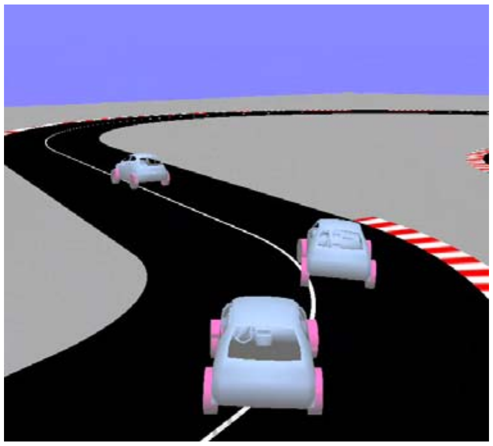
Fig. 7. *Demo application: efficiency.*

b) The simultaneous simulation of three cars in real-time, one driven by a human driver and the two others automatically driven by controllers developed within the proposed framework for the replica of Cataluña circuit shown in Fig. 7. A module for collisions was included, so that the cars could impact each other, thus forcing the controllers to react in order to keep the vehicles inside the road and to follow the correct trajectories. This application served to demonstrate the efficiency of the whole approach, since it is capable of dealing with three detailed car models (163 variables each) at the same time, two of them operated by controllers, and to show that robust controllers can be designed with the proposed tool.

7. Conclusions

In order to show the applicability of a real-time formulation for the dynamics of multibody systems to vehicle control, a low-cost and efficient computational framework for the design of automobile motion controllers has been developed. The core elements of the tool are the real-time formalism for the dynamics of multibody systems, a VR interface for human-in-the-loop simulation, and the fuzzy logic toolbox of Matlab. The conclusions of the work are:

a) Although the dynamic formalism is global (easy-to-implement) and features an implicit integrator (stable and robust), real-time performance is comfortably achieved for a detailed model of a car, implemented in Fortran language on a conventional PC.

b) In the controller design process, the VR interface serves to provide an initial assessment on maneuver feasibility and realistic values of the involved parameters, to have a reference to measure the quality of the successive controllers obtained, and to enable simultaneous operation of human driver and controller for partial control applications, like driver assistance.

c) The use of fuzzy logic is justified since the system is highly nonlinear, a simplified model is not required, the driver's behavior is reproduced, and a reliable Matlab toolbox is available.

d) Matlab Engine is a good option to connect Fortran (car model) and Matlab (fuzzy logic controller) codes during the design stage, while the matricial storage of the designed controller is the solution for real-time applications.

e) The proposed framework can deal with cars undergoing highly violent maneuvers (robust), and enables the simultaneous real-time simulation of human-driven and automatically controlled vehicles (efficient).

References

1. Antos, P., Ambrosio, J.: *A control strategy for vehicle trajectory tracking using multibody models*. In: Multibody System Dynamics, vol. 11, 2004, p. 365-394.
2. Gordon, T.J., Best, M.C., Dixon, P.J.: *An automated driver based on convergent vector fields*. In: Proc. Instn. Mech. Engrs. Part D: J. Automobile Engineering, vol. 216, 2002, p. 329-347.
3. Frezza, R., Saccon, A., Minen, D., Ortmann, C.: *Smart driver: a research project for closed loop vehicle simulation in MSC.ADAMS*. In: Multi-body Dynamics. Monitoring and Simulation Techniques III, London and Bury St Edmunds, Prof. Eng. Publishing, 2004, p. 401-413.
4. Hebden, R.G., Edwards, Ch., Spurgeon S.K.: *Automotive steering control in a split-μ manoeuvre using an observer-based sliding mode controller*. In: Vehicle System Dynamics, vol. 41, 2004, p. 181-202.
5. Hayakawa, Y., White, R., Kimura, T., Naito G.: *Driver-compatible steering system for wide speed-range path following*. In: IEEE/ASME Trans. on Mechatronics, vol. 9, 2004, p. 544-552.
6. Cuadrado, J., Cardenal, J., Morer, P., Bayo, E.: *Intelligent simulation of multibody dynamics: space-state and descriptor methods in sequential and parallel computing environments*. In: Multibody System Dynamics, vol. 4, 2000, p. 55-73.
7. Garcia de Jalon, J., Bayo, E., *Kinematic and dynamic simulation of multibody systems –The real-time challenge–*. New York. Springer-Verlag, 1994.
8. Bakker, E., Pacejka, H.B.: *The magic formula tyre model*. In: 1st Int. Colloquium on Tyre Models for Vehicle Dynamics Analysis. Delft, Netherlands, 1991, p. 1-18.
9. Mamdani, E.H., Assilian, S.: An experiment in linguistic synthesis with fuzzy logic controller. In: *Int. J. of Man-Machine Studies*, vol. 7, 1975, p. 1-13.