

## Implementation Issues of an on Board Real-Time Multibody Model.

Emilio Sanjurjo<sup>\*</sup>, Roland Pastorino<sup>#</sup>, Pasquale Gallo<sup>†</sup>, Miguel A. Naya<sup>\*</sup>

<sup>\*</sup> Mechanical Engineering Laboratory      <sup>#</sup> PMA division, Department of Mechanical Engineering  
University of La Coruña      KU Leuven  
C/ Mendizábal s/n, 15403, Ferrol, Spain      Celestijnenlaan 300b, 3001 Leuven, Belgium  
[emilio.sanjurjo, minaya]@udc.es      roland.pastorino@mech.kuleuven.be

<sup>†</sup> Department of Management and Engineering  
Università degli Studi di Padova  
Stradella S. Nicola 3, 36100, Vicenza, Italy  
pgallo@gest.unipd.it

### ABSTRACT

Improvements in multibody formulations and the increase of computational power open new fields of application for multibody simulations. In this work, a real-time multibody model of a prototype was developed and installed on board the vehicle, so both model and vehicle have the same inputs and run in parallel. The multibody model is fed with the signals coming from the vehicle's sensors. This work demonstrates that is possible to run a complex multibody model in-vehicle, so it becomes clear that controllers based on complex multibody models could be available in a near future.

### 1 INTRODUCTION

Modern automobiles are safer than those of few years ago. Many factors contribute to this safety improvement, such as, for example, enhancements in chassis design, suspensions, and tyres. But nowadays an automobile is not only a mechanical machine. We are assisting to the proliferation of electronic aids (ABS, electronic stability control, rollover prevention, lane change prevention, adaptive speed controller, etc.) [8], which make the vehicles safer and more comfortable. Current and future development of more advanced and sophisticated control systems requires accurate and instantaneous vehicle dynamic information, such as acceleration, body angular rates or wheel slips. Some of this information could be easily achieved with sensors, but some magnitudes are unmeasurable due to technical or economical reasons. A more practical approach is to estimate that dynamical magnitudes of interest using low-cost sensors and a state observer, which can be based either on a kinematic or a dynamical model [10]. The more detailed the dynamical model is, the more reliable information can be extracted from it, at the cost of having to identify all the parameters needed to characterize the model. For this reason, state observers based on multibody models became a field of interest over the last years [2, 4, 11, 13].

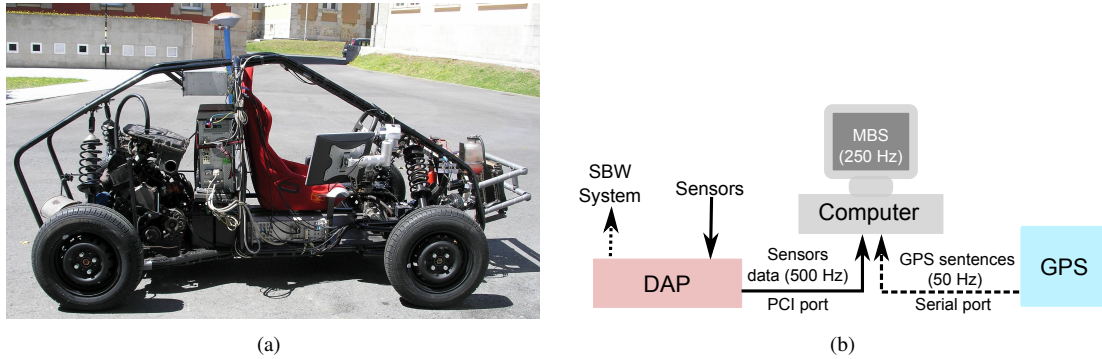
With the advent of electric and hybrid cars, production vehicles are becoming smarter: the need of monitoring the batteries and managing the electric engines leads to an increment of the computational power available on board. In order to maintain the hardware and the software structure as simple as possible, the trend will be to have only one system, but more powerful than the many subsystem available today [14]. For these reasons, it is expected that the next generations of production vehicles will have a great increase of computational power inside, allowing complex multibody-based state observers to be the fundamental part of their control algorithms. The advantage of this approach is that only one observer is needed, as the multibody model contains all the dynamical information that could be required.

The long term goal of the authors is to implement a state observer based on a multibody simulation on board a vehicle. As a previous step, in this work is presented a real-time multibody model of a vehicle running on board. This model has been yet validated [12], although in the current version have been improved some force models and a new method of modelization was developed for the wheels. This work addresses the implementation issues of this multibody simulation, and is organized as follows: section 2 describes the prototype and its hardware; in section 3 the multibody formulation is described; the multibody model is

depicted in section 4; after that, the force models are detailed in section 5; then, in section 6 is depicted the software architecture; finally, some conclusions are remarked.

## 2 THE PROTOTYPE

The prototype was built at the Mechanical Engineering Laboratory of the University of La Coruña. It is a full-size steer-by-wire (SBW) vehicle (Figure 1(a)).



**Figure 1.** (a) The prototype. (b) Hardware communication scheme.

This vehicle is powered by a four cylinder petrol engine. The power is transmitted to the rear wheels through an automatic gearbox, while the brake system is composed of four discs. Simulation hardware consists of a conventional desktop computer (Intel Core 2 Duo processor, 2 Gb of RAM) connected to a PCI data acquisition processor (DAP), model DAP 4200a made by Microstar Laboratories. The only modifications made to the computer are the power supply and the hard drive. The power supply works with 12 V of direct current, provided by the battery of the prototype. The hard disk drive was changed by a solid state drive (SSD), because it is more resistant to vibrations. The DAP controls the SBW system and samples all the sensors installed on the vehicle at 500 Hz. Then, it sends the data through a PCI port to the computer. This rate provides good performance to do the necessary real time control of the SBW system. The full list of the installed sensors is given in Table 1. A high precision GPS receiver was added to the vehicle instrumentation. It is not connected to the DAP, but it is connected to the computer via a serial port. The GPS works at 50 Hz. The hardware communication schema can be seen in Figure 1(b). Most of the sensors installed on the prototype were selected because they are similar to the ones that can be found in a production vehicle, while others, such as the wheel torque sensor and the high precision GPS, are necessary in order to develop force models and to validate future state observers.

**Table 1.** List of installed sensors.

Measured magnitudes	Sensor
Vehicle accelerations (X,Y,Z)	Accelerometers
Vehicle angular rates (X,Y,Z)	Gyroscopes
Vehicle tilt angles (X,Y)	Inclinometers
Wheel rotation angles	Hall-effect sensors
Brake line pressure	Pressure sensor
Steering wheel and steer angles	Encoders
Engine speed	Hall-effect sensor
Steering torque	Inline torque sensor
Throttle pedal angle	Encoder
Rear wheel torque	Wheel torque sensor
Position, speed and course	GPS receiver

## 2.1 Some GPS considerations

A GPS receiver calculates the position of its antenna by measuring the distance from this antenna to a constellation of satellites. The distance is determined measuring the time since a satellite signal is sent until it arrives to the receiver's antenna. Positioning errors can be produced due to the position errors of the satellites, atmospheric effects, clock drifts, etc. However, if two GPS receivers are available, and one of them is at a known location, this receiver can measure the positioning errors, and the other one can use this information to improve its precision. This technique is called differential GPS. The GPS used in this work employs this technique to achieve 1 cm of precision under optimal conditions, and can reach an output rate of 50 Hz. Moreover, a GPS receiver can give speed measurements based on the Doppler effect.

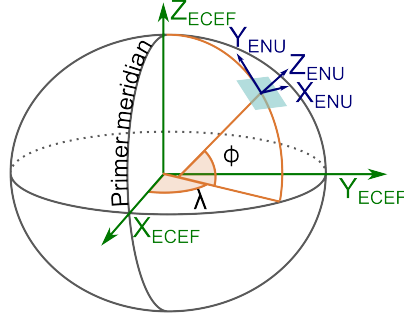


Figure 2. ECEF and ENU reference systems

The GPS positioning output consist of geodetic coordinates: longitude ( $\lambda$ ), latitude ( $\phi$ ), and ellipsoidal height ( $h$ ). By default, these coordinates are referred to the WGS84 reference system (the system in which the GPS satellites are located), but if differential corrections are applied, they can include a reference transformation. In this work, corrections provided by the IGN (the Spanish National Geographic Institute) were used. These corrections change the reference system from WGS84 to ETRS89, the official geodetic reference system in Spain, which moves with the Eurasian plate. The geodetic parameters of latitude and height are defined in terms of the ellipsoid normal at the user's position. Notice that unless the user is on the poles or the equator, the ellipsoid normal does not point exactly towards the center of the earth (see Figure 2). In order to be used along with the multibody model, the geodetic coordinates are not very useful, so they must be transformed to the same reference axis of the multibody simulation: the ENU (east, north, up) reference system. This transformation is carried out in two steps. Firstly, a transformation from geodetic to ECEF (Earth-centered, Earth-fixed) coordinate system following the next equations [9]:

$$\mathbf{r}^{ECEF} = \begin{pmatrix} \frac{a \cos \lambda}{\sqrt{1 + (1 - e^2) \tan^2 \phi}} + h \cos \lambda \cos \phi \\ \frac{a \sin \lambda}{\sqrt{1 + (1 - e^2) \tan^2 \phi}} + h \sin \lambda \cos \phi \\ \frac{a(1 - e^2) \sin \phi}{\sqrt{1 - e^2 \sin^2 \phi}} + h \sin \phi \end{pmatrix} \quad (1)$$

being  $a = 6378137.000$  m the semi-major axis of the ETRS89 ellipsoid,  $b = 6356752.314140$  m its semi-minor axis, and  $e = \sqrt{1 - b^2/a^2}$  the ellipsoid eccentricity. Then, the transformation from ECEF to ENU must be accomplished as follows:

$$\mathbf{r}^{ENU} = \mathbf{R} (\mathbf{r}_P^{ECEF} - \mathbf{r}_0^{ECEF}) \quad (2)$$

where  $\mathbf{r}_P^{ECEF}$  is the position of the GPS receiver,  $\mathbf{r}_0^{ECEF}$  is the position of the origin of the ENU system, and  $\mathbf{R}$  is a rotation matrix, following the next equation:

$$\mathbf{R} = \begin{pmatrix} -\sin \lambda & \cos \lambda & 0 \\ \cos \lambda \sin(-\phi) & \sin \lambda \sin(-\phi) & \cos(-\phi) \\ \cos \lambda \cos(-\phi) & \sin \lambda \cos(-\phi) & -\sin(-\phi) \end{pmatrix} \quad (3)$$

### 3 THE MULTIBODY FORMULATION

The multibody formulation used in this work is an index-3 augmented Lagrangian formulation with mass-orthogonal projections [3, 5]. The equations of motion have the following form:

$$\mathbf{M}\ddot{\mathbf{q}} + \Phi_{\mathbf{q}}^T \alpha \Phi + \Phi_{\mathbf{q}}^T \boldsymbol{\lambda}^* = \mathbf{Q} \quad (4)$$

where  $\mathbf{M}$  is the mass matrix,  $\ddot{\mathbf{q}}$  are the accelerations,  $\Phi_{\mathbf{q}}$  the Jacobian matrix of the constraint equations,  $\alpha$  the penalty factor,  $\Phi$  the constraints vector,  $\boldsymbol{\lambda}^*$  the Lagrange multipliers and  $\mathbf{Q}$  the vector of applied forces. The Lagrange multipliers are obtained from the following iterative process:

$$\boldsymbol{\lambda}_{i+1}^* = \boldsymbol{\lambda}_i^* + \alpha \Phi_{i+1} \quad i = 0, 1, 2, \dots \quad (5)$$

The sub-index  $n$  stands for the time step, while the sub-index  $i$  refers to the iteration step within a time step. The employed integration scheme is the trapezoidal rule. The corresponding difference equations in velocities and accelerations are:

$$\dot{\mathbf{q}}_{n+1} = \frac{2}{\Delta t} \mathbf{q}_{n+1} + \hat{\mathbf{q}}_n \quad \text{with} \quad \hat{\mathbf{q}}_n = - \left( \frac{2}{\Delta t} \mathbf{q}_n + \dot{\mathbf{q}}_n \right) \quad (6)$$

$$\ddot{\mathbf{q}}_{n+1} = \frac{4}{\Delta t^2} \mathbf{q}_{n+1} + \hat{\hat{\mathbf{q}}}_n \quad \text{with} \quad \hat{\hat{\mathbf{q}}}_n = - \left( \frac{4}{\Delta t^2} \mathbf{q}_n + \frac{4}{\Delta t} \dot{\mathbf{q}}_n + \ddot{\mathbf{q}}_n \right) \quad (7)$$

Then, the equations of motion are discretized by introducing the difference equations 6 and 7 into equation 4. The solution of this nonlinear discrete system is obtained using an iterative Newton-Raphson method. Finally, mass-damping-stiffness-orthogonal projections in velocities and accelerations are performed to guarantee the compliance of the velocity and accelerations constraints,  $\dot{\Phi} = 0$  and  $\ddot{\Phi} = 0$ , which are not imposed in the solution process.

### 4 THE MULTIBODY MODEL

The real-time multibody model was defined with natural coordinates [1] although relative coordinates were used for modelling the wheels. The details of the method used for the wheels will be given later. The model consists of 18 rigid bodies (see Figure 3): the chassis, the four wheels, four knuckles, three bars for the steering mechanism, four arms for the front suspensions (double wishbone), and two more arms for the rear suspensions (MacPherson strut). It has 14 degrees of freedom: 6 for the rigid body motion of the chassis, 4 for the suspension and 4 more for the wheels' rotation. The steering is kinetically guided, so it is not a degree of freedom.

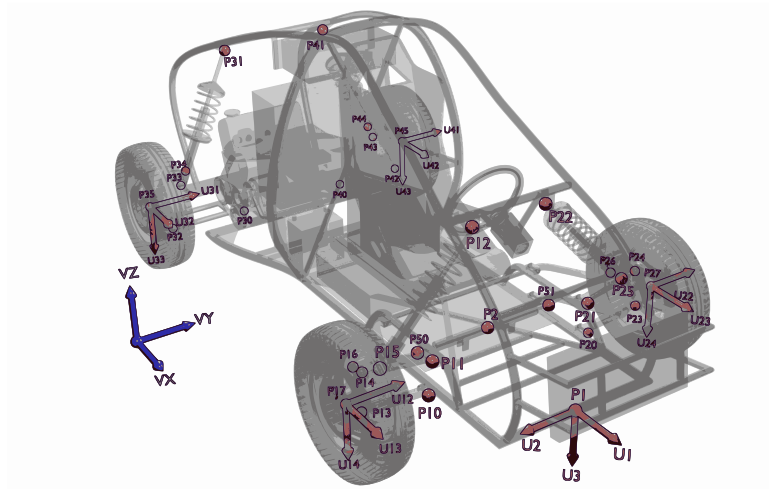
The inputs for the multibody model are given by sensors installed in the prototype: a hydraulic pressure sensor in the brake line, an in-wheel torque sensor mounted in one of the rear wheels, and an encoder measuring the position of the steering rack and pinion mechanism. The hydraulic pressure sensor is used as an input for a brake model, which will be explained in section 5. The output of this brake model is the torque applied at the front wheels. The torque measured with the in-wheel torque sensor is applied to the rear wheels. The output of this sensor is the sum of the drive and brake torque applied to the wheel. As both rear wheels have the same brake system, and the engine power is delivered through a differential, it is assumed that the torque is the same in both rear wheels.

While developing the model, special care was put on its geometric characterization. For example, the prototype has some construction errors which lead to wheel misalignments. This fact produces a rise of the rolling resistance and has effects on the steering behaviour. The rolling resistance is treated deeply in section 5. The exact orientation of the wheels was measured with a wheel aligner at a local garage, and then that orientation was introduced into the multibody model.

#### 4.1 Wheel modelling

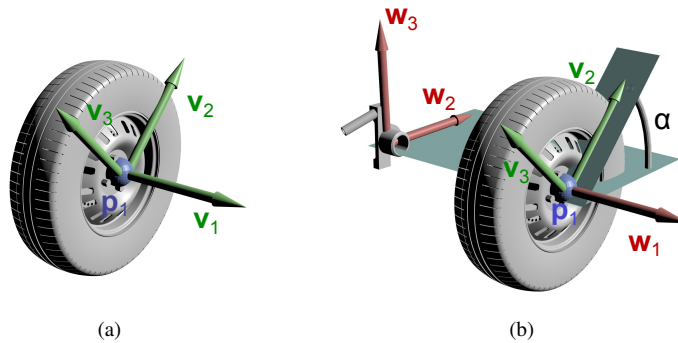
The easiest way to define a wheel with natural coordinates is to use one point and three vectors attached to the wheel (see point  $\mathbf{p}_1$ , and vectors  $\mathbf{v}_1$ ,  $\mathbf{v}_2$  and  $\mathbf{v}_3$  in Figure 4(a)). However this modelling technique is





**Figure 3.** Points and main vectors of the multibody model

not well suited to be applied to fast rotating elements: it leads to great energy dissipation, the convergence is slow, so many iterations are needed to solve the dynamic system, and the program crashes with velocities greater than 50 km/h. For these reasons, a new approach was developed in this work: the wheel is modelled using point  $\mathbf{p}_1$ , vectors  $\mathbf{w}_1$ ,  $\mathbf{w}_2$ ,  $\mathbf{w}_3$ , and angle  $\alpha$  (see Figure 4(b)). Vectors  $\mathbf{w}_2$  and  $\mathbf{w}_3$  are equal to vectors  $\mathbf{v}_2$  and  $\mathbf{v}_3$  at the beginning of the simulation, but they are now attached to the knuckle. The angle  $\alpha$  is the turn of the wheel. In order to develop this model, some conditions were assumed:



**Figure 4.** (a) Old wheel model. (b) Current wheel model.

- Coordinate vector is  $\mathbf{q} = (\mathbf{p}_1 \ \mathbf{w}_1 \ \mathbf{w}_2 \ \mathbf{w}_3 \ \alpha)^T$ .
- Both the center of mass of the wheel and the point  $\mathbf{p}_1$  are aligned with the axle of the wheel. The distance between them is  $r_g$ .
- Moments of inertia of the wheel  $I_1$ ,  $I_2$ , and  $I_3$  are calculated with respect to point  $\mathbf{p}_1$  and following the orientation of vectors  $\mathbf{w}_1$ ,  $\mathbf{w}_2$  and  $\mathbf{w}_3$ . The wheel has rotational symmetry, so  $I_2 = I_3$ .
- Vectors  $\mathbf{w}_1$ ,  $\mathbf{w}_2$  and  $\mathbf{w}_3$  are orthonormal.

These assumptions take to the next mass matrix.

$$\mathbf{M} = \begin{pmatrix} m\mathbf{I}_3 & mr_g\mathbf{I}_3 & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,1} \\ mr_g\mathbf{I}_3 & (I_2 - \frac{1}{2}I_1)\mathbf{I}_3 & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,1} \\ \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \frac{1}{2}I_1\mathbf{I}_3 & \mathbf{0}_{3,3} & \frac{1}{2}I_1\mathbf{w}_3 \\ \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \frac{1}{2}I_1\mathbf{I}_3 & -\frac{1}{2}I_1\mathbf{w}_2 \\ \mathbf{0}_{1,3} & \mathbf{0}_{1,3} & \frac{1}{2}I_1\mathbf{w}_3^T & -\frac{1}{2}I_1\mathbf{w}_2^T & I_1 \end{pmatrix} \quad (8)$$

where  $\mathbf{I}_3$  is the identity matrix of size 3, and  $\mathbf{0}_{n,m}$  is a matrix of zeros with size  $n \times m$ . The kinetic energy can be calculated as follows.

$$T = \frac{1}{2}\dot{\mathbf{q}}^T\mathbf{M}\dot{\mathbf{q}} \quad (9)$$

where  $\dot{\mathbf{q}}$  is the vector of velocities. Applying the Lagrange's equation for dependent coordinates (10) and developing the equations of motion for  $\mathbf{w}_2$  (11) and  $\mathbf{w}_3$  (12) leads to:

$$\frac{d}{dt} \left( \frac{\partial T}{\partial \dot{\mathbf{q}}} \right) - \frac{\partial T}{\partial \mathbf{q}} + \Phi_{\mathbf{q}}^T \boldsymbol{\lambda} = \mathbf{Q} \quad (10)$$

$$\frac{1}{2}\ddot{\mathbf{w}}_2 I_1 + \frac{1}{2}I_1\mathbf{w}_3\ddot{\alpha} + \Phi_{\mathbf{w}_2}^T \boldsymbol{\lambda} = \mathbf{Q}^{\mathbf{w}_2} - \underline{I_1\dot{\mathbf{w}}_3\dot{\alpha}} \quad (11)$$

$$\frac{1}{2}\ddot{\mathbf{w}}_3 I_1 - \frac{1}{2}I_1\mathbf{w}_2\ddot{\alpha} + \Phi_{\mathbf{w}_3}^T \boldsymbol{\lambda} = \mathbf{Q}^{\mathbf{w}_3} + \underline{I_1\dot{\mathbf{w}}_2\dot{\alpha}} \quad (12)$$

where  $\Phi_{\mathbf{q}}$  is the Jacobian of the constraints with respect to  $\mathbf{q}$ ,  $\boldsymbol{\lambda}$  is the Lagrange multipliers vector,  $\Phi_{\mathbf{w}_2}^T$  and  $\Phi_{\mathbf{w}_3}^T$  are the partial derivatives of the constraints with respect to  $\mathbf{w}_2$  and  $\mathbf{w}_3$  respectively,  $\mathbf{Q}$  is the generalized forces vector, and  $\mathbf{Q}^{\mathbf{w}_2}$  and  $\mathbf{Q}^{\mathbf{w}_3}$  are the elements of the generalized forces vector correspondent to  $\mathbf{w}_2$  and  $\mathbf{w}_3$  respectively. The underlined terms in equations (11) and (12) are velocity-dependent terms representing gyroscopic and Coriolis forces. These terms must be added to the vector of generalized forces.

#### 4.1.1 Generalized forces vector assembly

In order to fit the multibody equations, the forces must be projected over the multibody variables, in this case, point  $\mathbf{p}_1$ , vectors  $\mathbf{w}_1$ ,  $\mathbf{w}_2$  and  $\mathbf{w}_3$ , and angle  $\alpha$ . In order to ease the projection process, an auxiliary system attached to the wheel is considered. It is composed of point  $\mathbf{p}_1$  and vectors  $\mathbf{w}_1$ ,  $\mathbf{v}_2$  and  $\mathbf{v}_3$  (see figure 4(b)), although vectors  $\mathbf{v}_2$  and  $\mathbf{v}_3$  are no part of the wheel model now. Lets consider a force applied to one point of the wheel, and calculate its power:

$$\dot{W} = \mathbf{F}^T \dot{\mathbf{r}} = \mathbf{F}^T \mathbf{C}\dot{\mathbf{q}}_{\mathbf{v}} = \mathbf{Q}^T \dot{\mathbf{q}}_{\mathbf{v}} \quad (13)$$

being  $\mathbf{q}_{\mathbf{v}} = (\mathbf{p}_1 \ \mathbf{w}_1 \ \mathbf{v}_2 \ \mathbf{v}_3)^T$  and  $\mathbf{r}$  the position vector of the force application point, which can be expressed as

$$\mathbf{r} = \mathbf{p}_1 + c_1\mathbf{w}_1 + c_2\mathbf{v}_2 + c_3\mathbf{v}_3 = \mathbf{C}\mathbf{q}_{\mathbf{v}} \quad (14)$$

where  $\mathbf{C}$  is as follows

$$\mathbf{C} = ( \ \mathbf{I}_3 \quad c_1\mathbf{I}_3 \quad c_2\mathbf{I}_3 \quad c_3\mathbf{I}_3 \ ) \quad (15)$$

According to equation 13, the generalized forces vector would be  $\mathbf{Q} = \mathbf{C}^T\mathbf{F}$ , but as vectors  $\mathbf{v}_2$  and  $\mathbf{v}_3$  are not part of the multibody model, additional transformations must be done in order to assemble the generalized forces vector. The coordinates  $\mathbf{q}_{\mathbf{v}}$  are related with the multibody coordinates  $\mathbf{q}$  with a transformation matrix  $\mathbf{T}$ .

$$\mathbf{q}_{\mathbf{v}} = \mathbf{T}\mathbf{q} \quad (16)$$

$$\mathbf{T} = \begin{pmatrix} \mathbf{I}_3 & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,1} \\ \mathbf{0}_{3,3} & \mathbf{I}_3 & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,1} \\ \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{I}_3 \cos(\alpha) & \mathbf{I}_3 \sin(\alpha) & \mathbf{0}_{3,1} \\ \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & -\mathbf{I}_3 \sin(\alpha) & \mathbf{I}_3 \cos(\alpha) & \mathbf{0}_{3,1} \end{pmatrix} \quad (17)$$

According to equation 16, the time derivative of  $\mathbf{q}_{\mathbf{v}}$  can be expressed as follows.

$$\dot{\mathbf{q}}_{\mathbf{v}} = \mathbf{T}\dot{\mathbf{q}} + \dot{\mathbf{T}}\mathbf{q} \quad (18)$$

where  $\dot{\mathbf{T}}$  is

$$\dot{\mathbf{T}} = \dot{\alpha} \begin{pmatrix} \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,1} \\ \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,1} \\ \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & -\mathbf{I}_3 \sin(\alpha) & \mathbf{I}_3 \cos(\alpha) & \mathbf{0}_{3,1} \\ \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & -\mathbf{I}_3 \cos(\alpha) & -\mathbf{I}_3 \sin(\alpha) & \mathbf{0}_{3,1} \end{pmatrix} \quad (19)$$

The term  $\dot{\mathbf{T}}\mathbf{q}$  can be rearranged as follows.

$$\dot{\mathbf{T}}\mathbf{q} = \mathbf{B}\dot{\mathbf{q}} = \begin{pmatrix} \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,1} \\ \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,1} \\ \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & -\sin(\alpha)\mathbf{w}_2 + \cos(\alpha)\mathbf{w}_3 \\ \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & -\cos(\alpha)\mathbf{w}_2 - \sin(\alpha)\mathbf{w}_3 \end{pmatrix} \dot{\mathbf{q}} \quad (20)$$

where  $\mathbf{B}$  is an auxiliary matrix employed to ease the notation. Combining this result with equation 18:

$$\dot{\mathbf{q}}_{\mathbf{v}} = (\mathbf{T} + \mathbf{B}) \dot{\mathbf{q}} = \mathbf{D}\dot{\mathbf{q}} \quad (21)$$

Introducing this result in the equation 13 gives the result below.

$$\dot{W} = \mathbf{F}^T \dot{\mathbf{r}} = \mathbf{F}^T (\mathbf{C}\dot{\mathbf{q}}_{\mathbf{v}}) = \mathbf{F}^T (\mathbf{C}\mathbf{D}\dot{\mathbf{q}}) \quad (22)$$

Finally, identifying terms, the generalized forces vector results as follows.

$$\mathbf{Q} = (\mathbf{C}\mathbf{D})^T \mathbf{F} \quad (23)$$

## 5 FORCE MODELS

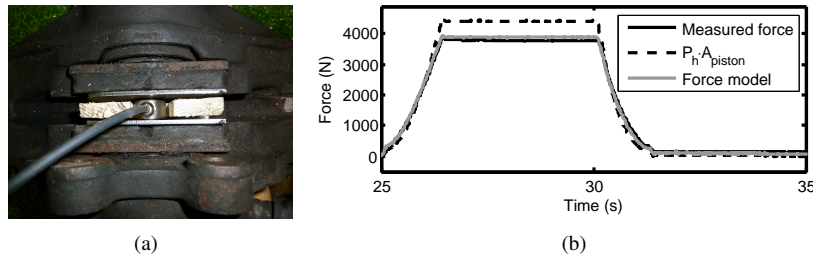
Force models are very important in order to get a good reliability in a multibody simulation. At low speed manoeuvres, the most important forces are those coming from the engine and brakes, the tyre forces, such as normal, longitudinal, and lateral forces, and the rolling resistance. The aerodynamic forces were not considered in this work, because they become important only at speeds higher than the considered in our tests.

The drive torque goes to the rear wheels through a conventional differential, and the brake systems are identical on both rear wheels, so it is assumed that the torque applied to both rear wheels is the same. Inside one of the rear wheels was installed a torque sensor, so brake, engine and gearbox models are not needed for the rear wheels.

On the front wheels the only torque applied is the one coming from the disc brake system, but the only available sensor is the brake pressure sensor, so a brake model has to be applied. This model has to be able to represent both stiction and sliding situations. The brake model used is based on the tangential force model described in [6], but neglecting its viscous component. This brake model is based on a combination of two effects: when the car is stand still, or when a wheel is blocked, the brake model behaves most like a spring-damper force, but when the wheel is spinning, the model behaves like dry friction. In order to compute the maximum available brake torque, the next equation is employed:

$$T_b = 2\mu F_n R_{eq} \quad (24)$$

where  $T_b$  is the dry friction component of the brake torque,  $\mu$  is the friction coefficient,  $F_n$  is the normal force between the disk and the pad, and  $R_{eq}$  is a geometric parameter representing the distance of the point of application of brake force and the axle of the wheel. The normal force between the disc and the pad cannot be measured while driving the prototype in an obvious way. Instead, a pressure sensor was installed in the brake line. In order to know how the hydraulic pressure is transmitted to the brake pads, the caliper was removed and a load cell was put between the pads (see Figure 5(a)). With this experimental setup the brake pedal was stepped, while logging data from the load cell and the hydraulic pressure. The force data was approximated with a cubic polynomial (see Figure 5(b)), so the force made by the piston to the pad can be calculated knowing the hydraulic pressure  $P_h$ , given by the pressure sensor installed on the prototype, and the brake piston area.



**Figure 5.** (a) Load cell between the brake pads. (b) Force model for the brake pads. This model approximates the real force (measured with the load cell) from the theoretical force, calculated as  $P_h A_{piston}$ , where  $P_h$  is the hydraulic pressure and  $A_{piston}$  is the brake piston area.

After the normal force is determined, the only unknown is the friction coefficient. To find this coefficient, the wheel torque sensor was installed in one of the front wheels and some manoeuvres were performed. Knowing the hydraulic brake pressure and the torque applied to the wheel, the friction coefficient can be straightforwardly determined using equation 24. This friction coefficient is  $\mu = 0.35$  for the front pads.

The tyre forces represent the interaction between the ground and the tyres, so a good characterization of the ground geometry is as important as the model itself. The prototype is intended to run over paved surfaces, so no deformation is considered on the ground. It is modelled as a triangle mesh, with data taken from a topographical survey of the test track. Moreover, the mesh is georeferenced in order to be able to put the multibody model in the proper position with the GPS measurements. The contact detection routine is explained in [6]. After a wheel contact is detected, the normal force is applied as a spring-damper force. The longitudinal and lateral forces are based on the TMeasy tyre model [7], but using linearised characteristic curves.

The rolling resistance was added as a brake torque, following the next equation,

$$T_{rr} = C_{rr} N R_w \quad (25)$$

where  $T_{rr}$  is the rolling resistance torque,  $C_{rr}$  is the rolling resistance coefficient,  $N$  is the normal force between the ground and the wheel, and  $R_w$  is the radius of the wheel. The rolling resistance coefficient was experimentally determined. In order to determine it, the prototype was let go down a slope-known ramp with the gearbox in neutral position and without braking. The position of the vehicle was measured, and then the acceleration was computed, so the rolling resistance coefficient can be calculated as follows

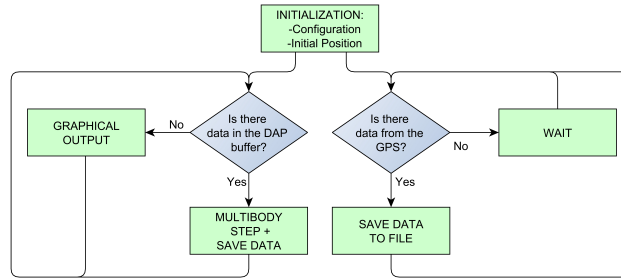
$$C_{rr} = \frac{a_r - a_t}{g \cos(\gamma)} \quad (26)$$

Being  $a_r$  the acceleration of the prototype,  $a_t$  the theoretical acceleration,  $g$  the gravity acceleration, and  $\gamma$  the angle of the ramp with the horizontal plane. The wheel misalignments lead to an increment of the rolling resistance, and the brakes also produces some friction, yet considered in the brake model. In order to not consider these phenomena twice, the theoretical acceleration was determined performing with the complete multibody model the same manoeuvre that was done with the prototype, but without taking into account the rolling resistance forces. The rolling resistance coefficient is  $C_{rr} = 0.017$ .

## 6 THE SIMULATION SOFTWARE ARCHITECTURE

This simulation program has some different components, most of them programmed in C++, although the multibody simulation was programmed in Fortran. The main components of the software are the multibody simulation, the DAP interface, the GPS interface and the graphical output. Currently, the software runs on Windows, so hard real time cannot be guaranteed. The only piece of software that runs in hard real time is the one running in the DAP. This software controls the steering by wire system, acquires the data from the sensors and sends them to the computer. From the computer side, this fact is used for synchronization

purposes: the DAP is used as the time reference for the entire program. As can be seen in Figure 6, the program has two execution threads. This structure allows to easily deal with the different sampling rates (the DAP works at 500 Hz, while the GPS works at 50 Hz), and takes advantage of the multi-core processors available nowadays.



**Figure 6.** Load cell between the brake pads

The program starts with an initialization sequence. During initialization the program reads a configuration file, where some parameters can be selected: the dynamical tolerance, the scenario, the time step of the multibody simulation, etc. After that, the initial position must be set. In order to do that, when the program is started the vehicle must be stopped. Then, the program requires that the driver of the vehicle drives a short distance (about 3 meters) in a straight line, and then stops the vehicle again. The GPS measures the positions before and after this short driving, so it can be calculated the initial position and yaw angle of the multibody model. After that, the main part of the program starts: the two execution threads are created. One of them samples the GPS and saves its data to a file, while the other executes the multibody simulation and the graphical output. The GPS thread works in an asynchronous way: the serial port buffer is gathered when it is possible. When there are some data in the serial buffer, then these data are parsed. When a block of GPS data is complete, it is saved to a file. The last block of GPS data is made available to the main thread of the program. The multibody thread uses the data coming from the DAP as the timing reference. This thread checks the DAP buffer. If a block of data is available, then it is gathered. As the DAP is set to work at 500 Hz, every time a block of data is gathered means that it is at least 2 ms later than the last time it happened. When a block of data is gathered, if it is time for the multibody to do a time step, it is done immediately. If not, the DAP buffer is checked again. When no data is available it means that the simulation is going too much fast, so it must wait. When this happens, the graphical output is generated. Although it is not an essential part, the graphical output is very useful in order to check the behaviour of the model while driving, so the driver can easily check the model performance. With this synchronization strategy, the multibody simulation is only allowed to run at time steps which are multiple of 2 ms. With a time step of 4 ms, the multibody always runs faster than real time, and allows a fluent graphical output. However, with slow motion manoeuvres in flat scenarios, real-time simulations can be attained with a time step of 2 ms.

## 7 CONCLUSIONS

This paper is focused on the development of a real-time multibody model of an automobile, and its implementation on board a vehicle. The hardware employed consists of a conventional computer with a PCI data acquisition processor (DAP) board. This DAP board samples the sensors installed on the vehicle. Then, sensor data feed the multibody model, so the model and the actual vehicle run in parallel and with the same inputs. Additionally, a GPS receiver was added.

The multibody model was modelled with natural coordinates, although a new formulation was developed to model the wheels using a relative angle. This new formulation improves the convergence and the conservation of energy. Moreover, most important force models were included: tyre forces, brake forces and rolling resistance. When possible, the coefficients of the models were experimentally determined.

Regarding the software architecture, the developed program has two execution threads. This architecture allows to easily deal with the different sample rate of the GPS receiver and the other sensors, and takes advantage of the multi-core processors available nowadays. In this program, the DAP is used as the timing

reference. The developed interfaces for the DAP board and the GPS receiver will be useful for future works, focused on the development of automotive state observers based on multibody models.

## 8 ACKNOWLEDGEMENTS

The research of Emilio Sanjurjo is funded by doctoral fellowship BES-2013-063598 of Spanish Government. The research of Roland Pastorino is funded by grant GCP2013/056 of Galician Government.

## REFERENCES

- [1] Bayo, E.; García de Jalón, J.: Kinematic and Dynamic Simulation of Multibody Systems: The Real-Time Challenge. Berlin: Springer, 1994.
- [2] Cuadrado, J.; Dopico, D.; Barreiro, A.; Delgado, E.: Real-Time State Observers Based on Multibody Models and the Extended Kalman Filter. *Journal of Mechanical Science and Technology*, Vol. 23, No. 4, pp. 894–900, 2009.
- [3] Cuadrado, J.; Dopico, D.; Naya, M.: Penalty, Semi-Recursive and Hybrid Methods for MBS Real-Time Dynamics in the Context of Structural Integrators. *Multibody System Dynamics*, Vol. 12, No. 2, pp. 117–132, 2004.
- [4] Cuadrado, J.; Dopico, D.; Perez, J. A.; Pastorino, R.: Automotive Observers Based on Multibody Models and the Extended Kalman Filter. *Multibody System Dynamics*, Vol. 27, No. 1, pp. 3–19, 2011.
- [5] Cuadrado, J.; Gutierrez, R.; Naya, M.: A Comparison in Terms of Accuracy and Efficiency Between a MBS Dynamic Formulation with Stress Analysis and a Non-Linear FEA Code. *International Journal for Numerical Methods in Engineering*, Vol. 51, No. 9, pp. 1033–1052, 2001.
- [6] Dopico, D.; Luaces, A.; Gonzalez, M.; Cuadrado, J.: Dealing with Multiple Contacts in a Human-In-The-Loop Application. *Multibody System Dynamics*, Vol. 25, No. 2, pp. 167–183, 2010.
- [7] Hirschberg, W.; Rill, G.; Weinfurter, H.: Tire Model TMeasy. *Vehicle System Dynamics*, Vol. 45, Supp. 1, pp. 101–119, 2007.
- [8] Høyve, A.: The Effects of Electronic Stability Control (ESC) on Crashes—an update. *Accident Analysis and Prevention*, Vol. 43, No. 3, pp. 1148–1159, 2011.
- [9] Kaplan, E. D.; Hegarty, C. J.: *Understanding GPS: Principles and Applications*. Norwood: Artech House, 2006.
- [10] Leung, K. T.; Whidborne, J. F.; Purdy, D.; Dunoyer, A.: A Review of Ground Vehicle Dynamic State Estimations Utilising GPS/INS. *Vehicle System Dynamics*, Vol. 49, No. 1-2, pp. 29–58, 2011.
- [11] Naets, F.; Pastorino, R.; Cuadrado, J.; Desmet, W.: Online State and Input Force Estimation for Multibody Models Employing Extended Kalman Filtering. *Multibody System Dynamics*, DOI 10.1007/s11044-013-9381-8, pp. 1–20, published online 2013.
- [12] Pastorino, R.; Dopico, D.; Sanjurjo, E.; Naya, M. A.: Validation of a Multibody Model for an X-BY-WIRE Vehicle Prototype Through Field Testing. In *ECCOMAS Thematic Conference. Multibody Dynamics 2011*, pp. 144–145, Brussels, 2011.
- [13] Pastorino, R.; Richiedei, D.; Cuadrado, J.; Trevisani, A.: State Estimation Using Multibody Models and Non-Linear Kalman Filter. *International Journal of Non-Linear Mechanics*, Vol. 53, pp. 83–90, 2013.
- [14] Sommer, S.; Camek, A.; Becker, K.; Buckl, C.; Zirkler, A.; Fiege, L.; Armbruster, M.; Spiegelberg, G.; Knoll, A.: RACE: A Centralized Platform Computer Based Architecture for Automotive Applications. In *2013 IEEE International Electric Vehicle Conference (IEVC)*, pp. 1–6, Santa Clara, CA, 2013.