MULTIBODY DYNAMICS 2003
Jorge A.C. Ambrósio(Ed.)
IDMEC/IST, Lisbon, Portugal, July 1-4 2003

REAL-TIME DETERMINATION OF THE POSITION OF A CAR WITH TRIAXIAL ACCELEROMETERS

Miguel A. Naya, Javier Cuadrado

Escuela Politécnica Superior
Universidad de La Coruña
Mendizábal s/n, 15403 Ferrol, Spain
e-mail: minaya@cdf.udc.es, web page: http://lim.ii.udc.es

Keywords: Vehicle Dynamics, Control Algorithms, Virtual Prototyping, Simulation, Data-Acquisition, Sensors

Abstract. *During the last years, our group has worked on several formulations for the dynamics of multi-body systems. Now, in order to find out if such methods are suitable to address real industrial problems, we intend to develop control algorithms for a car on its computer model (virtual prototyping), and evaluate the performance of such controllers when implemented on the corresponding physical prototype. A fundamental issue to carry out the control of a car is the accurate real-time determination of its motion at a reasonable cost. To this end, four tri-axial accelerometers have been installed in the chassis of the vehicle. Due to the traditional drawbacks coming from commercial accelerometers (inaccurate orientation, cross-sensitivity, signal drift, etc.), some additional sensors have been used: a Hall-effect sensor on a wheel, and a yaw rate sensor on the chassis. To integrate the signals, an augmented Lagrangian formulation has been applied, which ensures that positions, velocities and accelerations of the points in which the accelerometers have been placed fulfill the rigid body conditions. The resulting algorithm has been tested with sensor information coming either from simulation and experimental measurements, showing that a good level of accuracy can be achieved in the real-time determination of the vehicle motion.*

1 INTRODUCTION

During the last years, our group has worked on real-time formulations for the dynamics of multi-body systems[1,2]. As a result, a robust and efficient method has been developed[2]: an index-3 augmented Lagrangian formulation with projections of velocities and accelerations, which features natural coordinates for the modeling and the trapezoidal rule as numerical integrator. The method has shown an excellent behavior when facing singular configurations, changing topologies and stiff systems. Now, in order to find out if such formulation is suitable to address real industrial problems, we intend to develop control algorithms for a car on its computer model (virtual prototyping), and evaluate the performance of such controllers when implemented on the corresponding physical prototype. Fig. 1 shows the general diagram describing the mentioned objective.

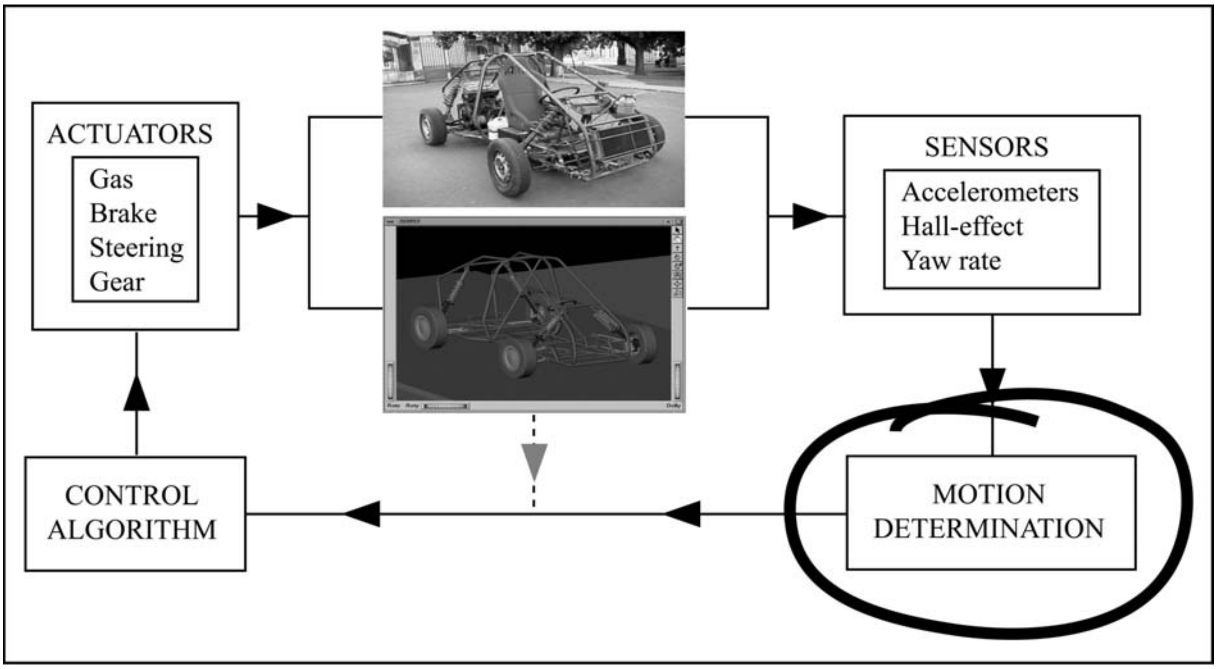

Figure 1: General context of the work.

The actual prototype has been built, and its virtual counterpart has been implemented on a computer through the mentioned dynamic formulation. The programming has been done in such a way that the inputs (actuators) and outputs (sensors) of the model and the physical prototype are exactly the same. Then, control algorithms can be designed and tested on the computer model of the car, until satisfying behavior of the controller is achieved. If the simulator is accurate enough, the resulting control algorithm should work properly also when implemented on the actual car.

The main difference between the real and virtual prototypes when addressing the design of motion control algorithms is that, in the real case, only information from the sensors is directly available, while, in the virtual case, information about position, velocity and acceleration of any point of the car can be obtained. Therefore, in order to attain the full analogy of the physical and computational systems, a motion determination module will be needed, which provides position, velocity and acceleration of any point of the chassis as functions of the sensor signals. This paper is focused on the development of such motion determination module. Other works on this topic can be found which show the state-of-the-art of the available systems for electronic control operation of commercial vehicles[3], or the suitability of GPS devices for an accurate detection of the vehicle motion[4].

2 THE PROPOSED ALGORITHM

In order to completely determine the chassis motion, when considered as a rigid body, information about the motion of three non-aligned points is needed. However, as data coming from sensors is never free from inaccuracies, tri-axial accelerometers have been installed in a tetrahedron configuration at four points of the chassis, as illustrated in Fig. 2. Care has been taken so that the dimensions of the tetrahedron are quite similar, so as to avoid ill-conditioning problems when rigid body conditions among the four points are imposed.

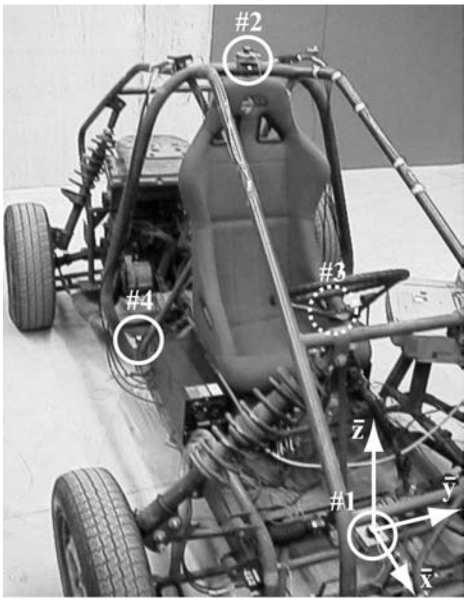

Figure 2: Local reference frame of the chassis and accelerometers situation.

Fig. 2 also shows the local reference frame which has been defined for the chassis. Local coordinates of the accelerometers are: (0, 0, 0), (-1.075, 0, 1.115), (-1.125, 0.3475, 0), (-1.125, -0.3475, 0). The problem variables are to be the global Cartesian coordinates of the four points where the accelerometers have been installed:

$$\mathbf{q}^{\mathrm{t}} = \left\{ \mathbf{r}_1^{\mathrm{t}} \quad \mathbf{r}_2^{\mathrm{t}} \quad \mathbf{r}_3^{\mathrm{t}} \quad \mathbf{r}_4^{\mathrm{t}} \right\} = \left\{ x_1 \quad y_1 \quad z_1 \quad x_2 \quad y_2 \quad z_2 \quad x_3 \quad y_3 \quad z_3 \quad x_4 \quad y_4 \quad z_4 \right\} \tag{1}$$

Therefore, the rotation matrix, which relates the local reference frame of the chassis and the inertial reference frame, can be expressed as function of the problem variables in the form,

$$\mathbf{R} = \mathbf{X}\overline{\mathbf{X}}^{-1} \text{ (as } \mathbf{X} = \mathbf{R}\overline{\mathbf{X}}\text{)} \tag{2}$$

where the bar above a variable indicates that it is written in the local reference frame, and

$$\mathbf{X} = \left[\mathbf{r}_2 - \mathbf{r}_1 \quad \mathbf{r}_3 - \mathbf{r}_1 \quad \mathbf{r}_4 - \mathbf{r}_1 \right] \quad ; \quad \overline{\mathbf{X}} = \left[\overline{\mathbf{r}}_2 - \overline{\mathbf{r}}_1 \quad \overline{\mathbf{r}}_3 - \overline{\mathbf{r}}_1 \quad \overline{\mathbf{r}}_4 - \overline{\mathbf{r}}_1 \right] \tag{3}$$

Since four points have been considered, it is convenient to define matrix **A** as,

$$\mathbf{A}=\begin{bmatrix}\mathbf{R} & \mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{R} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{R} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{R}\end{bmatrix} \tag{4}$$

which will serve to simultaneously rotate vectors associated to the four points.

If $\bar{\ddot{\mathbf{q}}}$ is the (12x1) vector containing the twelve signals coming from the accelerometers (three from each of them), which are expressed on the local reference frame, its global counterpart is

$$\ddot{\mathbf{q}}=\mathbf{A}\bar{\ddot{\mathbf{q}}} \tag{5}$$

However, as sensor information is never clean, the accelerations of the four points obtained through this method will not fulfill the rigid body conditions. Therefore, some constraint terms must be included. The method of Lagrange multipliers can be chosen for this purpose, so that Eq. (5) turns into,

$$\mathbf{I}\ddot{\mathbf{q}}+\mathbf{\Phi}_{\mathbf{q}}^{\mathrm{t}}\boldsymbol{\lambda}=\mathbf{A}\bar{\ddot{\mathbf{q}}} \tag{6}$$

where $\mathbf{\Phi}$ is the constraints vector, $\mathbf{\Phi}_{\mathbf{q}}$ is the Jacobian matrix of the constraints, and the (12x12) identity matrix $\mathbf{I}$ has been introduced for convenience.

The constraints vector plays the role of assuring that rigid body conditions among the four points are fulfilled. It can be formulated as follows,

$$\mathbf{\Phi}=\begin{Bmatrix}(\mathbf{r}_2-\mathbf{r}_1)^{\mathrm{t}}(\mathbf{r}_2-\mathbf{r}_1)-L_{12}^2\\ (\mathbf{r}_3-\mathbf{r}_1)^{\mathrm{t}}(\mathbf{r}_3-\mathbf{r}_1)-L_{13}^2\\ (\mathbf{r}_4-\mathbf{r}_1)^{\mathrm{t}}(\mathbf{r}_4-\mathbf{r}_1)-L_{14}^2\\ (\mathbf{r}_2-\mathbf{r}_1)^{\mathrm{t}}(\mathbf{r}_3-\mathbf{r}_1)-C_{23}\\ (\mathbf{r}_3-\mathbf{r}_1)^{\mathrm{t}}(\mathbf{r}_4-\mathbf{r}_1)-C_{34}\\ (\mathbf{r}_4-\mathbf{r}_1)^{\mathrm{t}}(\mathbf{r}_2-\mathbf{r}_1)-C_{42}\end{Bmatrix} \tag{7}$$

The three first elements of vector $\mathbf{\Phi}$ represent the constant distance conditions between point 1, origin of the local reference frame, and each of the other three points where accelerometers have been installed, being L_{12}, L_{13}, L_{14} the corresponding lengths. The three second elements of vector $\mathbf{\Phi}$ represent the constant dot products between each couple of vectors whose constant lengths have been imposed in the previous constraint equations. The six constants can be easily obtained from the previously given local coordinates of the points.

Looking at Eq. (6), it is easy to realize that the form of such set of equations is analogous to that of the Lagrange dynamic equations when they are formulated in dependent coordinates.

$$\mathbf{M}\ddot{\mathbf{q}}+\mathbf{\Phi}_{\mathbf{q}}^{\mathrm{t}}\boldsymbol{\lambda}=\mathbf{Q} \tag{8}$$

Therefore, any method applied to the solution of the equations of motion in dependent coordinates can be used to solve our problem, if the mass matrix **M** is substituted by the identity matrix, and the vector of applied forces is substituted by the product $\mathbf{A}\ddot{\bar{\mathbf{q}}}$.

Due to the good experience obtained by our group when using an index-3 augmented Lagrangian formulation with projections in velocities and accelerations to solve the dynamics of multi-body systems[5], such formulation has been selected to address the present problem. Then, Eq. (6) turns into,

$$\mathbf{I}\ddot{\mathbf{q}} + \mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}}\alpha\mathbf{\Phi} + \mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\lambda}^{*} = \mathbf{A}\ddot{\bar{\mathbf{q}}} \tag{9}$$

where the Lagrange multipliers are obtained from the following iteration process (given by sub-index i, while sub-index n stands for the time-step),

$$\boldsymbol{\lambda}_{i+1}^{*} = \boldsymbol{\lambda}_{i}^{*} + \alpha\mathbf{\Phi}_{i+1} \quad , \quad i=0,1,2,... \tag{10}$$

and the value of $\boldsymbol{\lambda}_{0}^{*}$ is taken equal to the $\boldsymbol{\lambda}^{*}$ obtained in the previous time-step. The penalty factor, α, has been set to a value of 10^9.

As integration scheme, the implicit single-step trapezoidal rule is adopted. The corresponding difference equations in velocities and accelerations are:

$$\dot{\mathbf{q}}_{n+1} = \frac{2}{\Delta t}\mathbf{q}_{n+1} + \hat{\dot{\mathbf{q}}}_{n} \quad \text{with} \quad \hat{\dot{\mathbf{q}}}_{n} = -\left(\frac{2}{\Delta t}\mathbf{q}_{n} + \dot{\mathbf{q}}_{n}\right) \tag{11}$$

$$\ddot{\mathbf{q}}_{n+1} = \frac{4}{\Delta t^{2}}\mathbf{q}_{n+1} + \hat{\ddot{\mathbf{q}}}_{n} \quad \text{with} \quad \hat{\ddot{\mathbf{q}}}_{n} = -\left(\frac{4}{\Delta t^{2}}\mathbf{q}_{n} + \frac{4}{\Delta t}\dot{\mathbf{q}}_{n} + \ddot{\mathbf{q}}_{n}\right) \tag{12}$$

If Eqs. (11-12) due to the integrator are introduced into Eq. (9) at time step n+1, the following non-linear set of equations is obtained, wherein the positions $\mathbf{q}_{n+1}$ are the unknowns,

$$\frac{4}{\Delta t^{2}}\mathbf{I}\mathbf{q}_{n+1} + \mathbf{\Phi}_{\mathbf{q}n+1}^{\mathrm{T}}\left(\alpha\mathbf{\Phi}_{n+1} + \boldsymbol{\lambda}_{n+1}^{*}\right) - \left(\mathbf{A}\ddot{\bar{\mathbf{q}}}\right)_{n+1} + \mathbf{I}\hat{\ddot{\mathbf{q}}}_{n} = 0 \tag{13}$$

For numerical reasons, the scaling of equation (13) by a factor of $\Delta t^2/4$ seems to be convenient, thus yielding

$$\mathbf{I}\mathbf{q}_{n+1} + \frac{\Delta t^{2}}{4}\mathbf{\Phi}_{\mathbf{q}n+1}^{\mathrm{T}}\left(\alpha\mathbf{\Phi}_{n+1} + \boldsymbol{\lambda}_{n+1}^{*}\right) - \frac{\Delta t^{2}}{4}\left(\mathbf{A}\ddot{\bar{\mathbf{q}}}\right)_{n+1} + \frac{\Delta t^{2}}{4}\mathbf{I}\hat{\ddot{\mathbf{q}}}_{n} = 0 \tag{14}$$

or, symbolically $\mathbf{f}(\mathbf{q}_{n+1}) = 0$.

In order to obtain the solution of this non-linear system, the widely used iterative Newton-Raphson method may be applied

$$\left[\mathbf{f}_{\mathbf{q}}(\mathbf{q})\right]_{i}\Delta\mathbf{q}_{i+1} = -\left[\mathbf{f}(\mathbf{q})\right]_{i} \tag{15}$$

where the residual vector is

$$\left[\mathbf{f}\left(\mathbf{q}\right)\right]=\frac{\mathit{\Delta} t^2}{4}\left(\mathbf{I}\ddot{\mathbf{q}}+\mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}}\alpha\mathbf{\Phi}+\mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\lambda}^{*}-\mathbf{A}\bar{\ddot{\mathbf{q}}}\right) \tag{16}$$

and the approximated tangent matrix

$$\left[\mathbf{f}_{\mathbf{q}}\left(\mathbf{q}\right)\right]=\mathbf{I}+\frac{\mathit{\Delta} t^2}{4}\mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}}\alpha\mathbf{\Phi}_{\mathbf{q}} \tag{17}$$

The procedure explained above yields a set of positions $\mathbf{q}_{n+1}$ that not only satisfies Eq. (9), but also the constraint conditions $\mathbf{\Phi}=0$. However, it is not expected that the corresponding sets of velocities and accelerations satisfy $\dot{\mathbf{\Phi}}=0$ and $\ddot{\mathbf{\Phi}}=0$, because these conditions have not been imposed in the solution process. To overcome this difficulty, projections in velocities and accelerations are performed[6]. It can be seen that the projections leading matrix is the same tangent matrix appearing in Eq. (17). Therefore, triangularization is avoided and projections in velocities and accelerations are carried out with just forward reductions and back substitutions.

If $\dot{\mathbf{q}}^{*}$ and $\ddot{\mathbf{q}}^{*}$ are the velocities and accelerations obtained after convergence has been achieved in the Newton-Raphson iteration, their cleaned counterparts $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ are calculated from

$$\left[\mathbf{f}_{\mathbf{q}}(\mathbf{q})\right]\dot{\mathbf{q}}=\mathbf{I}\dot{\mathbf{q}}^{*}-\frac{\mathit{\Delta} t^2}{4}\mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}}\alpha\mathbf{\Phi}_{t} \tag{18}$$

for the velocities, and

$$\left[\mathbf{f}_{\mathbf{q}}(\mathbf{q})\right]\ddot{\mathbf{q}}=\mathbf{I}\ddot{\mathbf{q}}^{*}-\frac{\mathit{\Delta} t^2}{4}\mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}}\alpha\left(\dot{\mathbf{\Phi}}_{\mathbf{q}}\dot{\mathbf{q}}+\dot{\mathbf{\Phi}}_{t}\right) \tag{19}$$

for the accelerations.

It should be noted that, since constraints vector $\mathbf{\Phi}$ does not depends on time (see Eq. (7)), the terms $\mathbf{\Phi}_{t}$ and $\dot{\mathbf{\Phi}}_{t}$ are, by now, identically null.

So far, a coherent formulation has been described to perform the rotation of the local accelerations coming from the sensors and, simultaneously, assure that rigid body conditions among the four points involved are fulfilled. However, when testing the resulting algorithm, it is observed that the global vertical coordinates of the points suffer from a drift during the integration, even when data from the simulation are recorded.

To overcome this hurdle, two slack variables are introduced for each of the four points, so as to avoid that their vertical coordinates drift away from reasonable values (as the car can neither fly, nor sink into the road). Therefore, the vector of problem variables given in Eq. (1) is increased from 12 to 20 variables,

$$\mathbf{q}^{\mathrm{t}}=\left\{\mathbf{r}_1^{\mathrm{t}}\quad\mathbf{r}_2^{\mathrm{t}}\quad\mathbf{r}_3^{\mathrm{t}}\quad\mathbf{r}_4^{\mathrm{t}}\quad s_{1u}\quad s_{1d}\quad s_{2u}\quad s_{2d}\quad s_{3u}\quad s_{3d}\quad s_{4u}\quad s_{4d}\right\} \tag{20}$$

where s_{iu} and s_{id} with i=1,...,4 stand for the “up” and “down” slack variables of point i.

The corresponding constraint equations for each point, which must be added to the vector of constraints previously stated in Eq. (7), are

$$\begin{aligned} \left(z_i^{\max} - z_i\right) - s_{iu}^2 = 0 \\ \left(z_i - z_i^{\min}\right) - s_{id}^2 = 0 \end{aligned} \tag{21}$$

being $z_i^{\max}$ and $z_i^{\min}$ the maximum and minimum value allowed for the global vertical coordinate of point i. Therefore, the size of the constraints vector $\mathbf{\Phi}$ rises until 14. A gap of ± 5 cm with respect to the equilibrium position of the car has been allowed for each point.

The last issue regarding the slack variables concerns the assigned value to the “mass” of these variables. Following with the analogy between the dynamic equations and Eq. (6), the role of the dynamic mass matrix is now played by the identity matrix for the first 12 variables (i.e. those variables corresponding to the coordinates of the four points). It has been observed that a value of 1.3**I** is appropriate for the “mass” associated to the slack variables. Then, the (12x12) identity matrix so far used when describing the formulation, must be substituted by the new (20x20) matrix

$$\mathbf{I}^* = \begin{bmatrix} \mathbf{I}_{\mathbf{12x12}} & \mathbf{0} \\ \mathbf{0} & 1.3\mathbf{I}_{\mathbf{8x8}} \end{bmatrix} \tag{22}$$

which is consistent with the new vector of variables given in Eq. (20). The slack variables have been previously used in multibody dynamics, as shown in the references[7].

When the described method is applied, it is observed that the path followed by the vehicle is quite well determined from the signals coming from the four tri-axial accelerometers. However, the orientation of the car is not altered during the motion (it just translates), which means that wrong results are obtained when a significant change in orientation is undergone.

To overcome this problem, a yaw rate sensor has been also installed on the chassis. To introduce the information coming from such sensor into the formulation so far described, the following non-holonomic constraint must be added to the set of constraints previously stated,

$$\mathbf{\Phi}_{\text{nh}} = \left(\mathbf{r}_4 - \mathbf{r}_1\right)\left(\dot{\mathbf{r}}_3 - \dot{\mathbf{r}}_1\right) + 2\bar{x}_3\bar{y}_3\bar{\omega}_z = 0 \tag{23}$$

where $\bar{\omega}_z$ is the local z-component of the angular velocity of the chassis, measured by the yaw rate sensor, and $\bar{x}_3$, $\bar{y}_3$ are the local coordinates of point 3.

Due to the non-holonomic character of Eq. (23), it cannot be directly included into the formulation as in the case of the other constraints[8], which are holonomic. In fact, Eq. (23) can be written as

$$\mathbf{\Phi}_{\text{nh}} = \mathbf{\Phi}_{\mathbf{q}}\dot{\mathbf{q}} + b(t) \tag{24}$$

Therefore, in the index-3 augmented Lagrangian formulation already described, the Jacobian matrix of the constraints, $\mathbf{\Phi_q}$, should be increased with the row coming from the new non-holonomic constraint. Furthermore, the terms containing time-derivatives of the constraints are no longer null, but

$$\begin{aligned}\Phi_t &= b(t)\\ \dot{\Phi}_t &= \dot{b}(t)\end{aligned} \tag{25}$$

which means that the derivative of the local z-component of the angular velocity of the chassis, $\dot{\bar{\omega}}_z$, will also be needed. It has been obtained by numerical differentiation of $\bar{\omega}_z$.

The information provided by the yaw rate sensor is not only used as explained. Further use of it has been made in order to correct the chassis orientation as the motion takes place. To this end, the global vertical component of the angular velocity of the chassis, ω_z, has been approximated by the $\bar{\omega}_z$ given by the sensor, and independently integrated along the time, by means of the trapezoidal rule.

$$\left(\varphi_z\right)_{n+1} = \left(\varphi\right)_n + \frac{1}{2}\left[\left(\omega_z\right)_n + \left(\omega_z\right)_{n+1}\right] \tag{26}$$

where φ_z is the yaw angle of the chassis.

Then, the yaw angle corresponding to the matrix rotation **R** obtained from the position of the four points, is compared, at each Newton-Raphson iteration (see Eq. (15)), with the yaw angle obtained from integration of the yaw rate sensor signal. In case that a discrepancy is detected between these two values, rotation matrix **R** is updated, so that the correct yaw angle is imposed. In this way, the orientation of the chassis during the motion is reasonably captured.

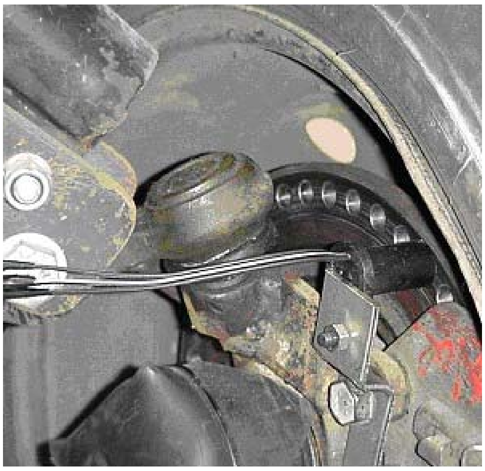

Figure 3: Hall-effect sensor.

The last remark to the proposed method refers to the use of a Hall-effect sensor, which has been installed on the front left wheel, as illustrated in Fig. 3. The purpose of such sensor is to serve as a switch indicating whether the algorithm should go on integrating the signals coming from the other sensors or not. The reason is that, if the car stops, only noise is received from the sensors, thus producing undesirable drifts in the results.

Therefore, when signals from the Hall-effect sensor have not been received for a time interval of 0.3 s, corresponding to a maximum car speed of 0.14 m/s, it is assumed that the car is at rest conditions, and the integration procedure is paused, keeping the problem variables the last value they had before the switch was performed. The pause is also used for re-calibrating the accelerometers and yaw rate sensor. When new pulses from the Hall-effect sensor are received, the switch is triggered again, and the integration restarts.

In a rather violent maneuver, there exists the risk that the wheel, in which the Hall-effect sensor is mounted, locks. This is why, to switch off the integration, the longitudinal velocity of one point is also checked, so as to corroborate that rest conditions have been achieved.

3 IMPLEMENTATION ASPECTS

The described algorithm has been programmed in Matlab 6.1 environment, and implemented on a laptop PC with processor Pentium IV @ 2 GHz, which can be mounted onboard the prototype. Data acquisition from the sensors is performed through a processor DAP3200a from Microstar Laboratories.

A sample time of 0.01 s has been adopted. Filters have not been applied to the signals to avoid the intrinsic delay they produce. In fact, the own integration procedure constitutes a filter, as previously explained. Before the integration starts, calibration of the accelerometers and yaw rate sensor is carried out, using the average values obtained from 200 samples.

Real-time, which is essential for the purpose of including this module into a control loop onboard the car, is comfortably reached. To give an idea of the algorithm speed, the CPU-time needed to integrate the signals off-line (provided they have been previously recorded and stored), corresponding to a maneuver of 40 s, is of 16.65 s, which means that the algorithm is 2.4 times faster than real-time.

4 SIGNALS FROM SIMULATION

A first validation of the developed algorithm for the determination of the car motion from the sensors signals has been addressed, by performing a simulation of the computational model of the prototype, and recording the data from the “simulated” sensors. Then, the algorithm has been applied to such data, and the obtained motion has been compared with the motion actually carried out by the virtual vehicle.

Fig. 4 shows the path followed by the car in the simulation (solid line), and the corresponding path obtained through application of the proposed algorithm to the virtual signals coming from the simulated sensors (x-marks). It can be appreciated that an excellent correlation has been achieved between both trajectories. It should be noted that the distance traveled by the car is rather long, close to 100 m. The maneuver was carried out at low speed, always under 40 Km/h, for a total time of 12 s.

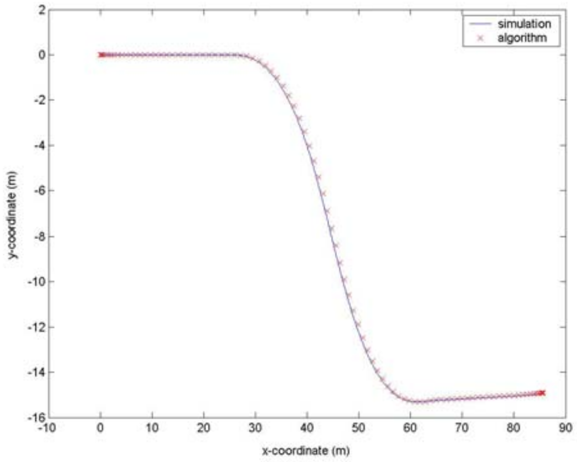

Figure 4: Integrated and actual trajectory of the simulated prototype.

The plot in Fig. 5 illustrates the behavior of the vertical coordinate of point 1 during the simulation, thus showing the stabilizing effect caused by the slack variables.

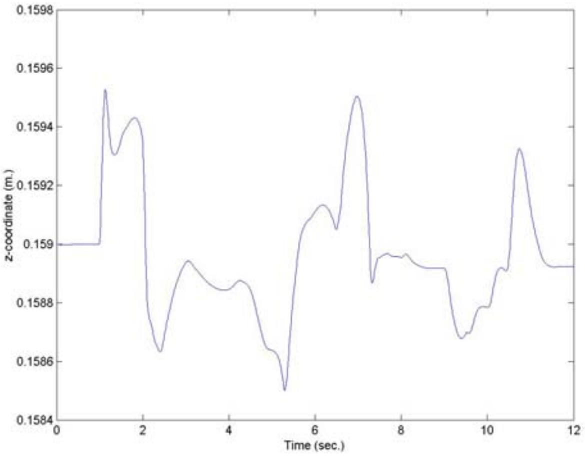

Figure 5: Effect of the slack variables on the vertical coordinate of point 1.

5 SIGNALS FROM EXPERIMENTAL MEASUREMENT

The second validation of the proposed algorithm has been carried out through integration of the signals coming from the actual sensors installed on the physical prototype.

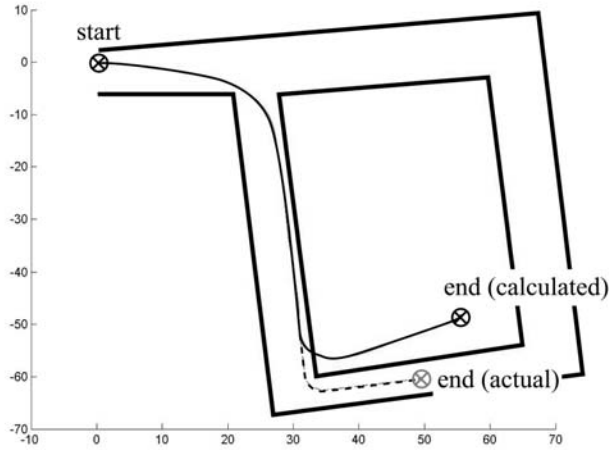

Figure 6: Integrated trajectory of the physical prototype.

Fig. 6 shows the path of the car obtained by application of the proposed algorithm for motion determination (solid line), together with the limits of the road, and the path actually followed by the vehicle (dashed line). The vehicle traveled over a distance of 120 m approximately, always under 40 Km/h. The total time of data recording was 30 s. As it can be seen in Fig. 6, the first part of the maneuver has been very well captured. However, an error of about 8 m can be appreciated at the end of the straight part of the path, as well as an excessive angle in the final turn to the left.

6 CONCLUSIONS

Based on the results previously described, the conclusions can be drawn as follows:

a) An algorithm has been developed which can serve as a motion determination module to obtain the actual motion of a car from the signals coming from the following set of sensors: four tri-axial accelerometers, a yaw rate sensor and a Hall-effect sensor.

b) The algorithm consists of an index-3 augmented Lagrangian formulation, which simultaneously performs the transformation to the inertial frame of the local accelerations coming from the accelerometers, and assures that the rigid body conditions among the four points (where the accelerometers are placed) are fulfilled.

c) The numerical integration is carried out by means of the single-step, implicit trapezoidal rule, which is combined with the equations provided by the augmented Lagrangian formulation, thus leading to a non-linear set of algebraic equations, solved through a Newton-Raphson iteration procedure. In order to enforce constraint fulfillment at first and second derivative level, projections of velocities and accelerations are performed at the end of each time-step, so producing a robust integration scheme.

d) To avoid drift appearance in the vertical motion, slack variables are included, so that only a reasonable vertical gap about the equilibrium position of the car is allowed.
e) To better capture the orientation of the vehicle chassis, information coming from the yaw rate sensor is included in the formulation through a non-holonomic constraint, and also used to correct the rotation matrix obtained from the four points in which the accelerometers are placed.
f) To avoid perturbation from sensors noise when the car is at rest, the pulses coming from the Hall-effect sensor serve to switch off the integration if such situation occurs, and to switch it on back when the car starts moving again.
g) Validation of the proposed algorithm has been carried out on the real car prototype and on its virtual counterpart, through computational and experimental techniques, respectively, having obtained acceptable results.

ACKNOWLEDGMENTS

This research has been sponsored by the Spanish CICYT (Grant No. DPI2000-0379) and the Galician SGID (Grant No. PGIDT01PXI16601PN).

REFERENCES

[1] J. Cuadrado, J. Cardenal and E. Bayo, “Modeling and Solution Methods for Efficient Real-Time Simulation of Multibody Dynamics”, *Multibody System Dynamics*, **1**, 259-280 (1997).

[2] J. Cuadrado, J. Cardenal, P. Morer and E. Bayo, “Intelligent Simulation of Multibody Dynamics: Space-State and Descriptor Methods in Sequential and Parallel Computing Environments*”, Multibody System Dynamics*, **4**, 55-73 (2000).

[3] L. Palkovics and A. Fries, “Intelligent Electronic Systems in Commercial Vehicles for Enhanced Traffic Safety”, *Vehicle System Dynamics*, **35**, 227-289 (2001).

[4] D.M. Bevly, J.C. Gerdes and C. Wilson, “The Use of GPS Based Velocity Measurements for Measurement of Sideslip and Wheel Slip”, *Vehicle System Dynamics*, **38**, 127-147 (2002).

[5] J. Cuadrado, R. Gutiérrez, M.A. Naya and P. Morer, “A Comparison in Terms of Accuracy and Efficiency between a MBS Dynamic Formulation with Stress Analysis and a Non-linear FEA Code*”, Int. J. for Numerical Methods in Engineering*, **51**, 1033-1052 (2001).

[6] E. Bayo and R. Ledesma, “Augmented Lagrangian and Mass-Orthogonal Projection Methods for Constrained Multibody Dynamics”, *Nonlinear Dynamics*, **9**, 113-130 (1996).

[7] J.M. Goicolea and J.C. Garcia, “Quadratic and Higher-Order Constraints in Energy-Conserving Formulations of Flexible Multibody Systems”, *Multibody System Dynamics*, **7**, 3-29 (2002).

[8] J. Garcia de Jalon and E. Bayo, *Kinematic and Dynamic Simulation of Multibody Systems –The Real-Time Challenge–*, Springer-Verlag, New York (1994).