Rapid prototyping of electric vehicle controllers combining Dymola and Simulink

Guillermo A. Magallán*, Luis I. Silva,
Cristian H. De Angelo and
Guillermo O. García

Grupo de Electrónica Aplicada,
Facultad de Ingeniería,
Universidad Nacional de Río Cuarto,
Río Cuarto, Ruta Nac. #36 – Km. 601,
ZIP: XC5804BYA, Rio Cuarto – Córdoba, Argentina
E-mail: gmagallan@ing.unrc.edu.ar
E-mail: lsilva@ieee.org
E-mail: cdeangelo@ieee.org
E-mail: g.garcia@ieee.org
*Corresponding author

Abstract: A methodology that allows the rapid prototyping of electric vehicle controllers is presented. A hybrid simulation scheme is implemented. It combines `Dymola` for the dynamic modelling of the prototype and `Simulink` for the design of the vehicle controller. This combination allows a complete simulation of the prototype obtaining, as a final product, the source code for the digital controller. The proposed methodology offers flexibility to evaluate different control strategies, modify the physical model, and verify results quickly. Consequently, time and costs of the development process are reduced.

Keywords: rapid prototyping; Dymola; Simulink; digital signal processors; DSP; simulation; electric vehicles; EVs.

Reference to this paper should be made as follows: Magallán, G.A., Silva, L.I., De Angelo, C.H. and García, G.O. (2012) 'Rapid prototyping of electric vehicle controllers combining Dymola and Simulink', *Int. J. Electric and Hybrid Vehicles*, Vol. 4, No. 2, pp.197–215.

Biographical notes: Guillermo A. Magallán received his Electronic Engineer degree from the Universidad Tecnológica Nacional, Paraná, Argentina, in 2002, and MSc and PhD in Engineering Sciences degree from the Universidad Nacional de Rio Cuarto, Argentina, in 2009 and 2010, respectively. Currently, he is an Assistant Professor at the Faculty of Engineering at the Universidad Nacional de Río Cuarto, Argentina. In 2005, he joined the Grupo de Electrónica Aplicada, Universidad Nacional de Río Cuarto. His research interests include traction control of electric vehicles, electric motors control, and DSP-based implementation.

Luis I. Silva received his Electronic Engineer degree from the Universidad Nacional de Rosario, Argentina, in 2005, Master in Space Sciences and Technologies from the Lulea Tekniska Universitet, Sweeden in 2007, and PhD in Engineering Sciences degree from the Universidad Nacional de Rio Cuarto, Argentina, in 2012. In 2007, he joined the Grupo de Electrónica Aplicada, Universidad Nacional de Río Cuarto, Argentina. His research interests are modelling and simulation applied to hybrid electric vehicles.

Cristian H. De Angelo received his Electrical Engineer degree from the Universidad Nacional de Río Cuarto, Argentina, in 1999, and PhD of Engineering degree from the Universidad Nacional de La Plata, Argentina, in 2004. In 1994, he joined the Grupo de Electrónica Aplicada, Universidad Nacional de Río Cuarto. He is also currently with CONICET. His research interests are in fault diagnosis on electric machines, sensorless motor control, electric vehicles, and renewable energy generation.

Guillermo O. García received his Electrical and Electronics Engineering degree from the Universidad Nacional de Córdoba, Argentina, in 1981, and MSc and PhD degrees in Electrical Engineering from COPPE, Universidade Federal do Rio de Janeiro, Brazil, in 1990 and 1994, respectively. In 1994, he joined the Universidad Nacional de Río Cuarto, Argentina, where he is currently the Director of the Grupo de Electrónica Aplicada. He is also with CONICET. His research interests are in power electronics, electric vehicles and renewable energy conversion.

1 Introduction

The development of control strategies for complex systems, such as an electric vehicle (EV) or a hybrid electric vehicle (HEV), satisfying all the required specifications, involves the use of different tools and procedures to validate theoretical proposals. The development procedure comprises always: modelling the complex system, proposing the control strategy, and implementing the proposed controller in a specific hardware. Different methodologies are used to carry out these stages. They include complex laboratory setups and prototypes making the entire procedure expensive and time consuming.

A first step in modelling is to analyse in detail the physical system in order to determine the simplifying hypotheses. Based on these hypotheses, a second step consists in proposing an ideal system that should derive in a simple mathematical model but with enough precision so as to represent the real system with sufficient fidelity, according to the stated objectives. This implies a trade-off between accuracy and system simplicity. Once the ideal system is obtained, a third step consists in translating the ideal model into a computational model capable to perform simulations.

Different models of EV and HEV have been used depending on the desired objectives, on the level of details to be achieved and on the availability of computational capacity. Vehicle models used in simulators can be classified into steady state, quasi steady state [e.g., ADVISOR (Hou and Guo, 2008), PSTAT (Alam and Gao, 2006) and QSS-TB (Rizzoni et al., 1999)] and dynamic models [e.g., PSIM (Onoda and Emadi, 2004), and VTB (Gökdere et al., 2002)]. Specific models working as parts of different types of simulators and emulators, such as hardware in the loop techniques (Huang and Chihsiuh, 2003), are very useful for quick evaluations and modifications of component behaviour in complex systems. A complete revision of vehicle models and their specific features can be found in Gao et al. (2007).

The modelling of EV and HEV presents some difficulties due to the fact that they are multi-physics systems, which means they involve multiple physical sub-systems and different physical phenomena such as mechanical, electrical, and chemical simultaneously. This implies the participation of specialists from different areas of

knowledge, used to work with their specific modelling approaches. This fact demands the existence of specialised tools to facilitate the interconnection of the different sub-models to simulate the whole system.

When it is necessary to obtain an accurate model of a complex physical system, an object-oriented modelling approach facilitate the task. Moreover, as it was stated before, where sub-systems of different physical nature coexist (e.g., electrical and mechanical as in the present case) appropriate interfaces between them are needed. Bond-graphs (BG) are an object-oriented modelling formalism that meets these needs (Gawthrop and Bevan, 2007). A BG model is a graphical representation of a system, independent of its physical nature, including a power-exchange mechanism that allows acting as a non-causal interface between sub-systems.

On the other hand, the most popular approach to model a control strategy is the block diagrams (BDs) approach due to its flexibility and simplicity. It gives a clear overall description of the input/output relationships.

Different alternatives to rapidly develop a control strategy for a complex system (rapid prototyping) were studied in recent years. Examples to be cited include the methodology proposed in Monti et al. (2003) to develop a power electronics controller combining VTB, `Simulink` and a dSPACE platform; or the use of `Dymola` to model a combustion engine and `Simulink` to design its controller (Hoffmann et al., 2006). On EV and HEV applications, some proposals reproduce the vehicle behaviour with an electronic hardware (Ramaswamy et al., 2004; Cheng et al., 2008), which demands very special and costly hardware setups.

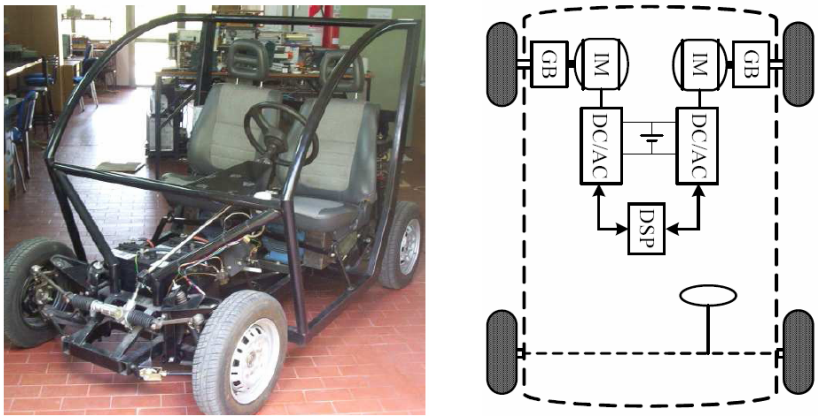
A new methodology for rapid prototyping of an EV traction controller is presented in this paper. The dynamic model of the physical parts of the prototype was carried out using the BG approach and constructed with `Dymola` (Dempsey, 2006), whereas the traction control algorithm was developed using BD and built in `Simulink`. The `Dymola` model was imported to the `Simulink` environment and interconnected with the traction control thus creating a combined model of the whole system. Several simulations, evaluation of results, and modifications of the models were done until the simulation results met the desired specifications. After that, the control algorithm was translated to a specific digital signal processors (DSP) code and downloaded to the physical (real) hardware to carry out tests with the prototype in order to validate experimentally the simulation results.

The rest of this paper is organised as follows: in Section 2, the EV under development is described; in Section 3, the proposed rapid prototyping methodology is explained; in Section 4 simulation and experimental results for different vehicle manoeuvres are shown, and finally, the conclusions are drawn in Section 5

2 Description and modelling of the EV prototype

This work is a result of the effort to develop an experimental EV prototype at GEA-UNRC. Figure 1(a) shows a picture of the prototype and Figure 1(b) shows a simplified scheme of its traction subsystem. It has a DSP to run the traction control algorithm. This algorithm includes two field-oriented controls (FOC), one for each induction machine (IM) fed by inverters (DC/AC) that allow controlling independently the torque of the rear wheels through gear boxes (GB). The traction control algorithm also includes an electronic differential.

Figure 1 (a) EV prototype and (b) EV prototype traction scheme (see online version for colours)



(a) (b)

Notes: GB: gear box; IM: induction motor; DC/AC: inverters; DSP: digital processor

All the mechanical components (chassis, suspensions and wheels) including the actuators (both IMs on the rear wheels) are defined as 'physical parts of the prototype', whereas 'traction controller' refers to the controller itself including all sensors.

2.1 Modelling of the physical parts of the prototype (Silva et al., 2008)

These parts were modelled in `Dymola` where models are described by schematics that consist of connected components. These components have 'connectors' that describe the interaction possibilities, e.g., an electrical pin, a mechanical flange, or an input/output signal. By drawing connection lines between connectors a physical system model can be constructed. Internally, a component is defined by an equation-based description of the model in the open Modelica (Fritzson and Bunus, 2002) syntax. This language is a textual description to define all parts of a model and to structure model components in libraries, called packages.

The modelling of the physical parts of the prototype was performed with multi-bond graphs (MBG) (Sanchez and Garcia, 1993) that is a vectorial extension of the regular BG. They are well suited for modelling multi-dimensional processes. Naturally, Modelica provides a library with elements to generate in `Dymola` a BG model (BondLib) and a library for MBG models (MultiBondLib).

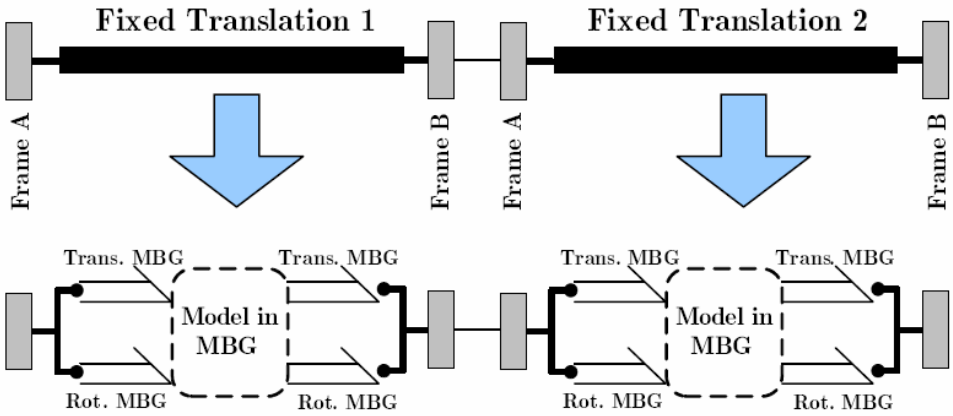
In 3D mechanical applications, three single bonds can be grouped into a multi-bond representing compactly the dynamics along the x, y, and z axes. To completely represent a 3D mechanical element two coupled MBGs models are needed, one for the rotational dynamics and one for translational dynamics.

Along this section, the model of the chassis, suspensions, and wheels are presented. They were carried out with components of the 'Modelica library for MBG' (Zimmer, 2006). These components are internally represented with elements of the MBG library and are provided with external frames where the rotational and translational multi-bonds are connected.

The mechanical connection between two 3D components is made via their frames that act as interfaces for exchanging power. The upper part of Figure 2 depicts a graphical

representation of the connection of two 3D components called 'fixed translation' while the lower part shows that their internal model uses the MBG approach. The connection implies that both components have the same angular/linear velocities and positions at the connection point.

Figure 2 Connection of 3D mechanical elements (see online version for colours)



2.1.1 Chassis

The model of the chassis, as represented in Figure 3(a), is composed by the sprung mass and a set of fixed translations. For the translational dynamics, the sprung mass behaves like a lumped mass but for the rotational dynamics it contains the inertia tensor thus acting like a rigid body.

A fixed translation acts like a massless rod that projects the forces and velocities from one extreme to the other. The topological disposition of the fixed translations produces an undeformable structure that links the sprung mass, located in the centre of mass to the frames 1, 2, 3 and 4 where the suspensions will be connected.

2.1.2 Suspensions

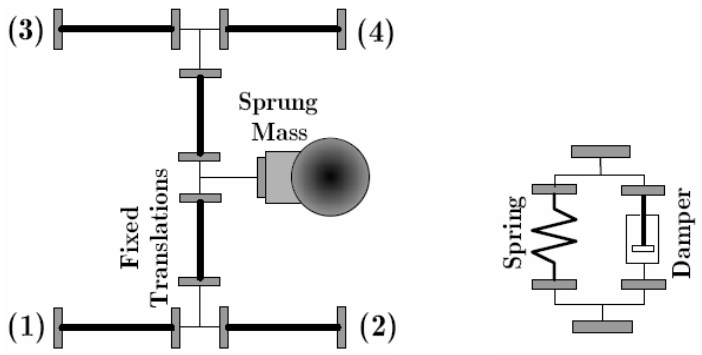
The suspension system is composed by four suspensions that couple each of the wheels to the chassis. Each suspension contains an ideal spring and a damper. Figure 3(b) shows the spring-damper connection that corresponds to the real mechanical configuration.

2.1.3 Wheels

The dynamic behaviour of the traction wheels is represented in two different BG models coupled by signals [see Figure 4(a)]. One part is devoted to the translational dynamics. It includes the calculation of the normal force at the tyre contact patch (F_N) based on the vertical dynamics. The longitudinal/lateral forces (F_x / F_y) are evaluated in the 'Pacejka' block using experimental curves (Pacejka and Sharp, 1991). The rotational dynamics receives the torque generated by the IM (T_e) and calculates its angular velocity (ω_r). The torque balance also includes: rolling resistance, inertia momentum of the wheel (I_W) and the opposition torque due to the longitudinal force on the tyre patch contact at distance

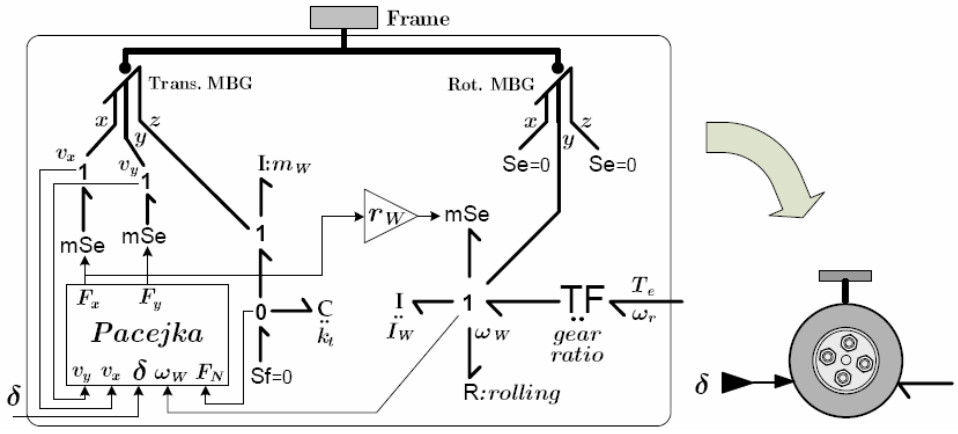
r_W. The wheel model was wrapped [Figure 4(b)] and provided with a frame to connect with the suspension, a single bond to interact with the IM, and an input to receive the steering signal (δ).

Figure 3 Chassis and suspension representation in Dymola



(a) (b)

Figure 4 (a) BG model of the wheel (b) External representation in Dymola

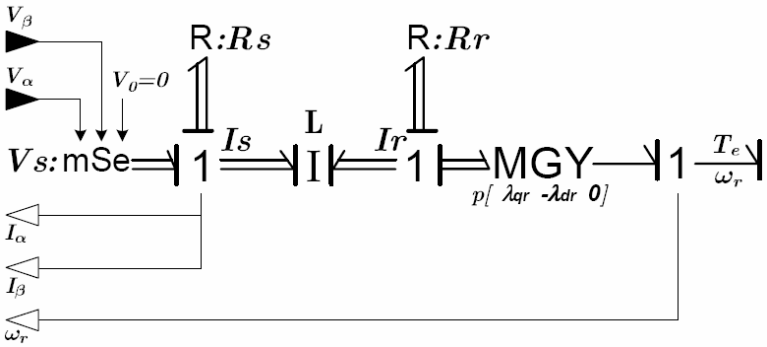


(a) (b)

2.1.4 Induction motors

From the electrical equivalent circuit corresponding to each phase of the IM, the BG model can be represented compactly using a MBG approach (Junco, 1993). The resultant model can be observed in Figure 5. It receives the reference voltages from the FOC and generates them directly with the modulated source of voltage (mSe). That is, the inverter is included in the model and is considered to be ideal. Stator currents I_α and I_β and ω_r are the measured signals (necessary to perform the FOC). The bond located on the right side shows the power interchange between the IM and the wheel. The causality on the bond states that the IM model outputs T_e and receives ω_r.

Figure 5 Induction motor representation in MBG



2.1.5 Physical parts assembling

For a complete model to be obtained all the sub-systems previously analysed must be interconnected. The assembling is carried out in the same way the physical interaction occurs; i.e., for rigid connections the frames are directly coupled; otherwise, a joint that allows certain degrees of freedom (relative movements) is interposed. The connection between the suspension system and the chassis is carried out through a revolute joint (it permits relative rotation along one axis), which allows the chassis pitch movement while each suspension remains in vertical position. Similarly, the lower end of the suspension system connects to the wheel's frame via a revolute joint. This connection allows the spinning of the wheel.

Figure 6 shows the complete connection of the left rear wheel and suspension. The connections of the other three wheels are similar except that front wheels do not have traction motors. The incorporation of a non-lineal resistance 'rnl' that models the dragging force only along longitudinal direction can also be observed in the same figure. The inputs of this sub-system are the steering angle (δ), which is always zero for rear wheels, and the reference voltages applied to the IM. Currents and rotor angular speeds are the outputs necessary to implement both FOCs.

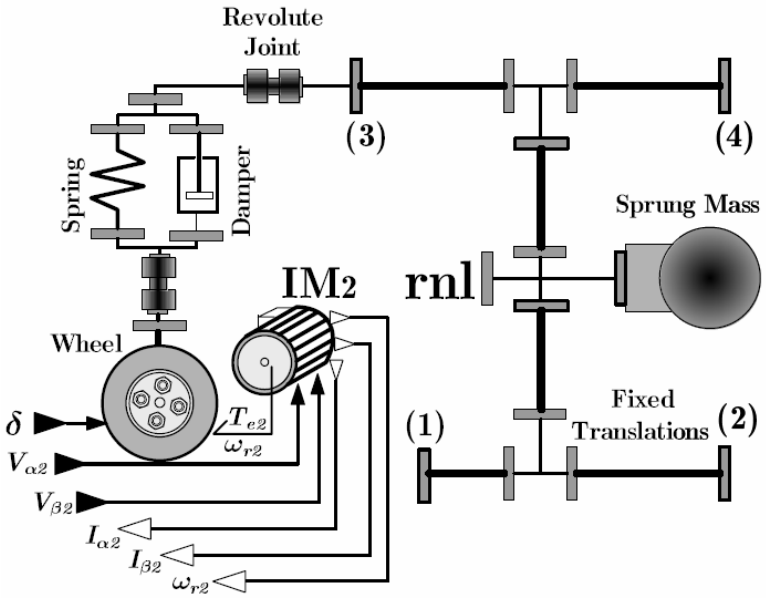
2.2 Model of the traction controller

The entire traction controller scheme was developed in BD and constructed with the aid of the `Simulink` package that is especially designed to interact with MATLAB. To create models in `Simulink` the user can easily drag blocks (grouped in different libraries or ToolBoxes) into a graphical user interface. Then, those blocks are reprogrammed (if needed) and they are interconnected. The internal programming consists either of a lower level BD or an 'embedded function' with instructions that allow programming the algorithms with the same code that a digital processor executes. The resulting model in `Simulink` has a direct graphic relationship with its BD representation (Mathworks, 2009).

The implemented traction control strategy, presented here as an example, consists in maintaining a torque balance in both rear wheels in order to emulate the traditional mechanical differential. This traction control strategy is described in detail in Magallán et al. (2009).

Following is a brief description of the traction control algorithm blocks and how its simulation schemes were implemented using BD.

Figure 6 Complete connection of chassis, left rear suspension, wheel and motor in Dymola



2.2.1 *Electronic differential system*

As it can be observed in Figure 7, inputs to this block are generated by the driver through accelerator and brake pedals. Saturation on the acceleration and brake signals represent the sensor's magnitude limitation. The electronic differential system (EDS) receives the measurements of the wheels' angular velocities, ω_{r1} and ω_{r2}, and controls the IM average speed (proportional to the lineal speed of the vehicle under ideal adherence conditions). The reference speed (ω_{ref}) is provided by the accelerator. When the brake is applied, ω_{ref} is forced to zero and the dynamic saturation in the output of the PI controller varies proportionally to the brake signal. The controller is a discrete PI and was implemented with an embedded function. The sampling time (T_s), is used to update the output synchronously with each new measurement executing the code discretely like the digital processor. The PI controller generates the output I_{qs}^* (proportional to the reference torque) that is an input for both FOCs. So, the same reference torque is given to both IMs.

2.2.2 *IM field-oriented control*

Figure 8 represents a direct vector control diagram for IMs (Krishnan, 2001). Input reference I_{ds}^* is used to set the magnetic flux with the upper PI controller. Input I_{qs}^* comes from the EDS and is used to control the torque via the lower PI controller. Measured variables together with the estimated flux angle (θ_λ) are used to express the currents in the field-oriented d,q frame. These currents are used in the PI controllers and, together with ω_r, are used to estimate θ_λ. Finally, the outputs of both PI (voltages in d,q

frame) are transformed to the α,β frame. Thus, the outputs of the FOC are the voltages (V_α and V_β) that must be generated in the IM input voltages through the inverter in order to generate the required currents (I_{ds}^* and I_{qs}^*). All the internal blocks shown in Figure 8 were implemented discretely using embedded functions.

Figure 7 BD representation of the EDS

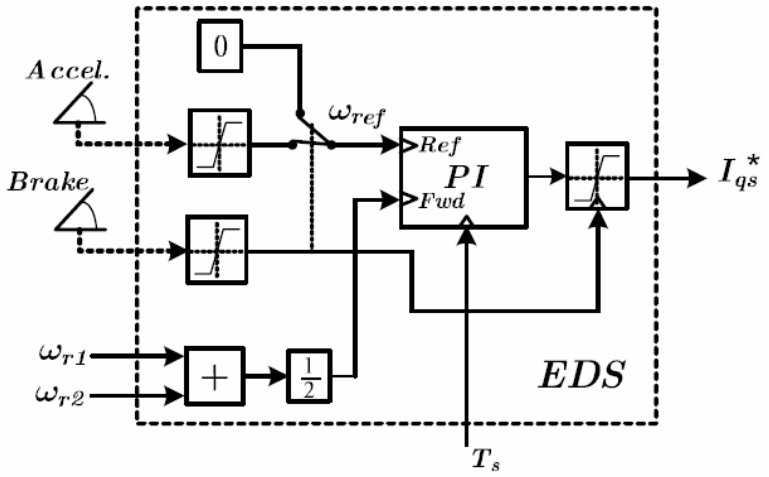
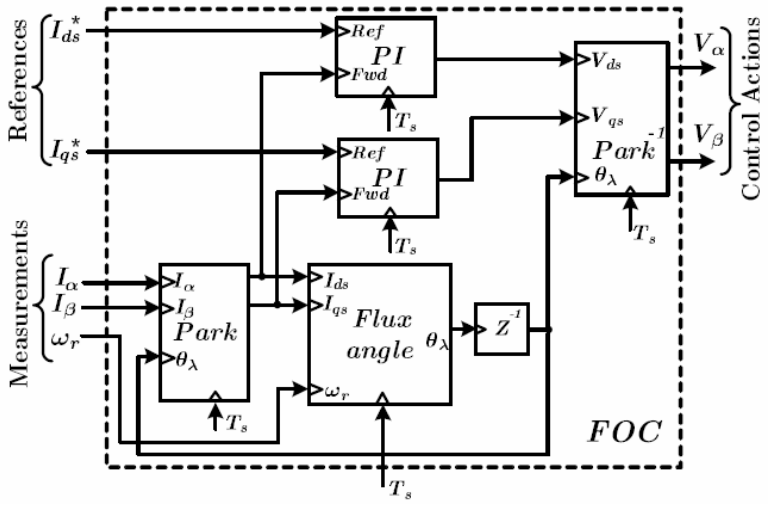


Figure 8 BD representation of the FOC



2.3 Simulation of the complete system

It seems to be impracticable to evaluate under simulation the performance of the complete system explained before because the model of the physical parts of the vehicle was developed in terms of BG but the traction controller was modelled using BD.

This drawback can be overcome either by modelling the control into the Dymola environment (Simic and Bäuml, 2008) or by modelling the dynamics of the plant using BD in the Simulink environment (Butler et al., 1999). If the first solution is adopted, all the advantages provided by the toolboxes of a powerful software such MATLAB cannot be exploited. On the other hand, taking the second solution means losing the physical modelling language approach and involves a huge amount of work to model the complete systems thus making the procedure error-prone.

Along the next section an alternative solution is presented. It keeps each model with its original approach and solves the incompatibility by software. This idea is developed for our particular case and then extended in order to obtain a methodology for rapid prototyping of EV controllers.

3 Proposed methodology

Given that the physical model and the control scheme have been modelled using different approaches, it became necessary to count with a methodology that allows the interconnection between them in order to perform simulations of the whole system.

Once the simulation results were satisfactory, that is, when all the required specifications were satisfied in the simulation results, the following step was to implement the control algorithm into the real hardware installed on the EV.

Finally, the whole prototype had to be conducted under different tests to validate experimentally the simulation results.

In the next subsections, the proposed methodology for conducting the above steps is described using as reference the flow diagram shown in Figure 9.

3.1 From the physical system to the Dymola model

This part of the methodology is represented in Figure 9 by the branch called *A*. On the basis of the specific objectives and a good knowledge of the physical system, the simplification hypotheses were defined and an ideal dynamic forward-looking model of the vehicle was developed. This ideal model was represented by using BG formalism.

In the case of this experience, the BG model was programmed in Dymola that offers special libraries with different BG elements. These elements are dragged into the graphic user interface and connected in order to build the model (see the complete building process in Subsection 2.1). The last step consists of wrapping the model into a single block called 'DymolaBlock' that Dymola provides. It is the programmer duty to provide this block with the appropriate inputs/outputs in order to interact with Simulink in a latter stage.

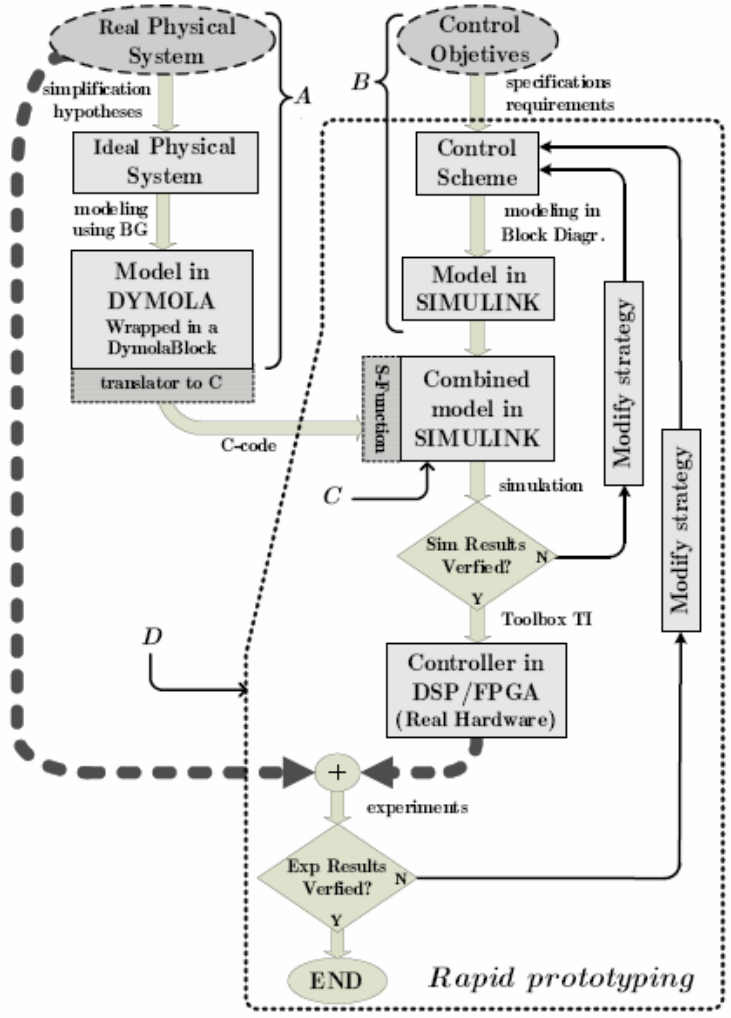
3.2 From the control objectives to the Simulink model

This part of the methodology is represented in Figure 9 by the branch called *B*. On the basis of the objectives and knowledge of the system specifications, a control strategy was decided. The resulting control scheme was represented using BD formalism.

In the case of this experience, the BD model was programmed in MATLAB-Simulink. Simulink is specially oriented to design, programme, and simulate dynamic systems

using BD formalism. The representation in BD was useful to clearly show the relationships between different subsystems, especially inputs and outputs.

Figure 9 Flow diagram summarising the proposed methodology



3.3 Simulink-Dymola interaction

This part of the methodology is represented in Figure 9 by the block called *C*. The Modelica translator of the `Dymola` environment takes the aforementioned DymolaBlock as an input and generates a differential-algebraic equation (DAE) system and transforms the DAE to state space form by symbolic manipulation. The final equations are stored as plain C code which is compiled and linked as an S-function Mex. This S-function appears in `Simulink` like a single block that inherits the set of inputs/outputs determined in the DymolaBlock. After this, the `Dymola` model was ready to be simulated in `Simulink` by applying the corresponding inputs. When this model was integrated with the traction

controller, the combined model was obtained. This new model was capable to run simulations of the complete system, including both physical parts and controller models, with the simulation parameters (numerical method, integration step, simulation time, tolerance, etc.) given in `Simulink`.

3.4 Rapid prototyping methodology

The algorithmic sequence of steps presented here is the core of the methodology. It is aimed to achieve the final prototype efficiently in terms of time and economical cost. The rapid prototyping methodology is represented in Figure 9 by the blocks that are within the demarcated area called *D*.

Two characteristics of `MATLAB-Simulink` were exploited here:

- Its capability to import and run models from other modelling environments.
- The possibility of transferring an algorithm from `Simulink` directly to a digital processor.

The first characteristic was used to integrate the model of the physical parts with the traction controller whereas the second characteristic accelerated the obtainment of the controller (hardware and software).

Simulation results of the combined model were used to adjust the control parameters and/or modify the control scheme. This process was repeated until all the required specifications were fulfilled.

When simulation results were satisfactory the control algorithm was ready to be transferred from plain `Simulink` diagrams to the prototype digital processor. In this particular application, a Texas Instrument toolbox that generates optimised C code for the TMS320F2812 DSP and transfers it to the real physical DSP hardware was used to implement the real digital controller. This possibility allowed to include the transferring process into the methodology and even to repeat it, if necessary.

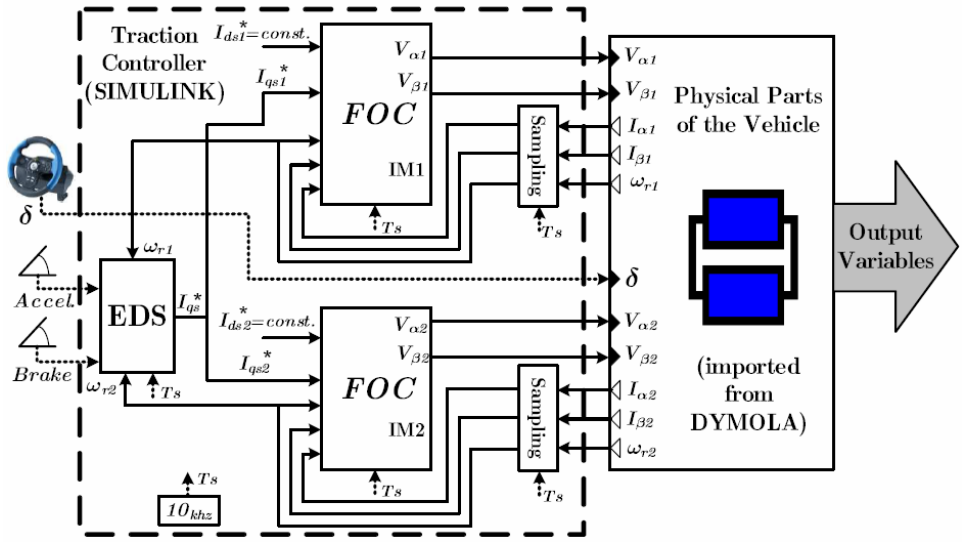
The methodology included a second verification after the controller was mounted on the vehicle, through experimental tests performed with the real vehicle. These experiments were supposed to match simulation results. If the experimental results were not satisfactory, the methodology suggests re-adjusting the controller parameters or propounding an alternative control strategy. The procedure must be resumed with the new control scheme and the previous steps have to be repeated. The rapid prototyping was finished when the experimental results proved that the required specifications were fulfilled.

4 Results

In this section, simulation and experimental results of a practical application of the proposed rapid prototyping methodology are presented. The methodology was used to develop a traction controller of the GEA-UNRC experimental EV.

The first step was to use the combined model to simulate manoeuvres where the features of the controller could be manifested. This allowed obtaining the final controller whose scheme is shown in Figure 10 connected with the imported `Dymola` model of the vehicle. The traction controller contains an EDS that feeds both FOCs with the same torque reference. These FOCs control independently each IM mounted on the rear wheels. Then the controller was translated to the DSP mounted on the EV. Finally, the same manoeuvres that were simulated were performed with the prototype and compared with the simulation results. Simulation and experimental results that correspond to a 'lineal acceleration and free movement', and a 'turning manoeuvre' are presented below. In all the tests, the IMs magnetisation current, I_{ds}, was maintained at their rated value.

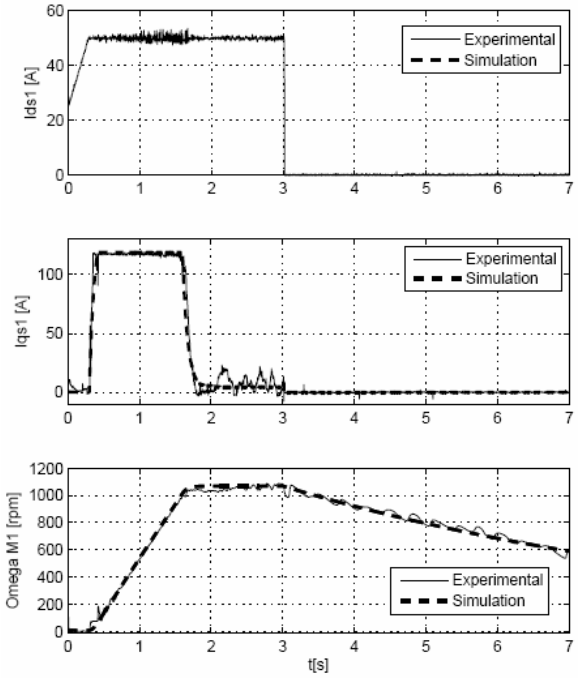
Figure 10 Final scheme of the combined model (controller and vehicle) (see online version for colours)



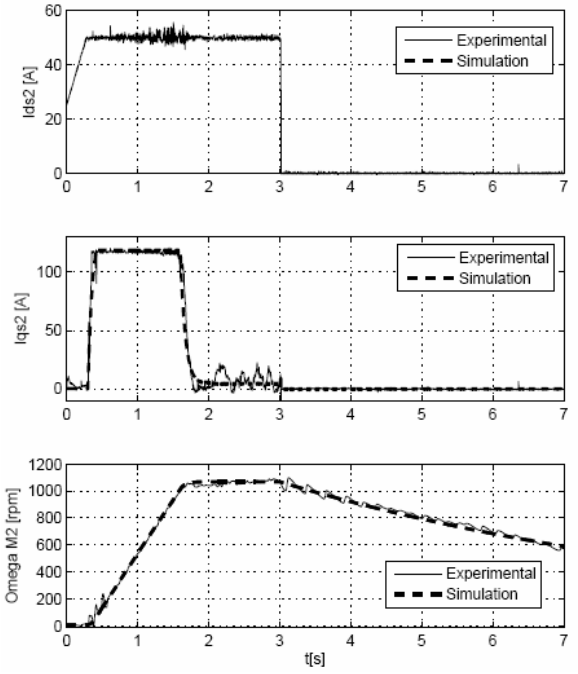
4.1 Lineal acceleration and free movement (no slope)

Figure 11 shows the simulation and experimental results obtained for this experiment. This figure shows the variables of both traction motors. This experiment began with the vehicle stopped. Once I_{ds} reached its rated value ($t \approx 0.3$ s) the maximum torque current, I_{qs}, was imposed and the IMs began to accelerate up to reach 1,000 rpm at $t \approx 1.8$ s. After that the speed was kept constant, consequently the torque current decreased to the value required to balance the friction torque. The IMs remained running at constant speed up to $t = 3$ s, when the motors were disconnected, I_{qs} and I_{ds} dropped to zero, then the vehicle moved forward freely, affected only by its inertia and friction.

Figure 11 Simulation and experimental results for the acceleration and free movement test, (a) top and (b) bottom wheel motor



(a)



(b)

The set of graphics presented shows that the experimental results matched the simulation results obtained previously.

4.2 Turning manoeuvre

Figures 12, 13 and 14 show simulation and experimental results of the steering angle, IMs variables, and average speed respectively obtained in this experiment.

Figure 12 Steering angle evolution during the turning manoeuvre

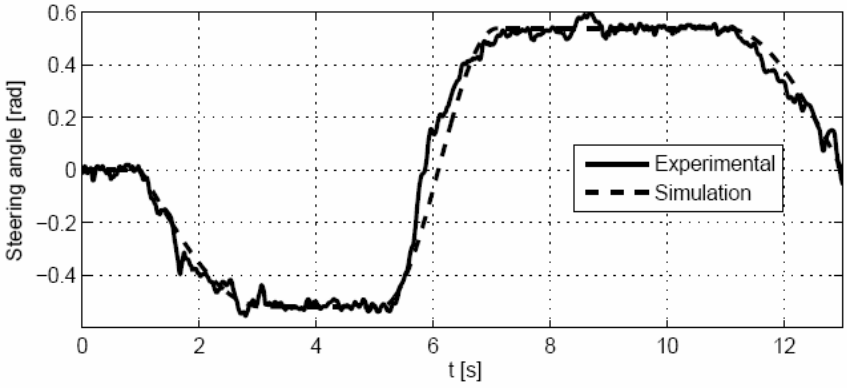
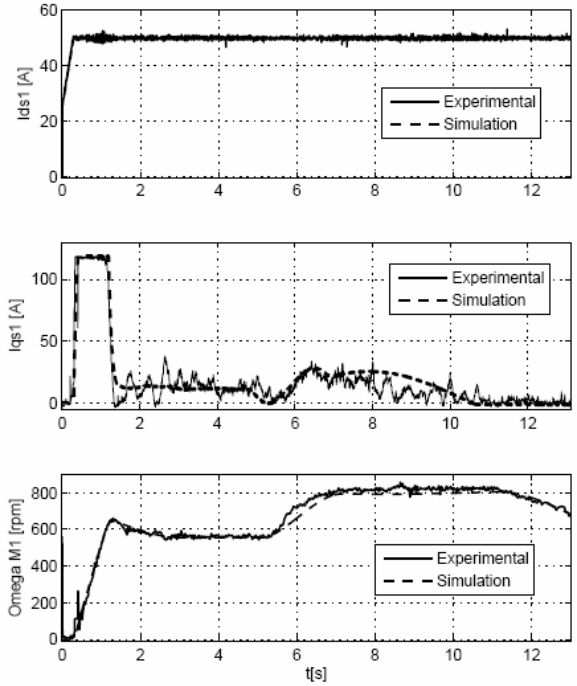
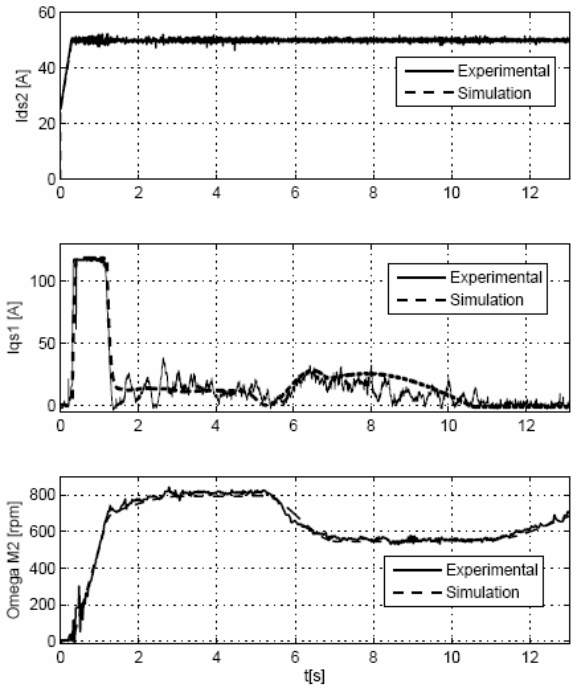


Figure 13 Simulation and experimental results for the turning manoeuvre test, (a) top and (b) bottom wheel motor



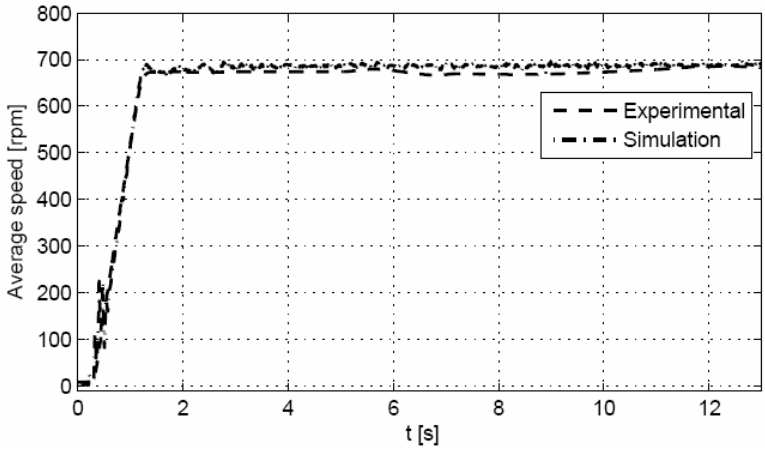
(a)

Figure 13 Simulation and experimental results for the turning manoeuvre test, (a) top and (b) bottom wheel motor (continued)



(b)

Figure 14 Average speed during turning manoeuvre



This experiment began with the vehicle stopped at $t = 0$ s. Then, the vehicle accelerated following a straight line path during 1.4 s. After that, the accelerator pedal was kept fixed until the end of the experiment. Figure 12 shows that the vehicle began turning right, then it began turning left and finally came back to a straight direction. Figure 13 shows that

while the vehicle was turning right, the inner curve wheel speed (IM1) was smaller than that of the outer one (IM2) and then changed when the vehicle turned left.

Figure 14 shows that during turning manoeuvres, the average speed remained constant.

Despite some small discrepancies between simulation and experimental results, wheel torques (I_{qs}) were the same for both motors, during all the experiment and the controller was capable to follow the accelerator pedal reference (constant average speed). This proved that the traction controller met the specifications validating the methodology.

5 Conclusions

A low-cost rapid prototyping methodology to develop EV controllers was presented in this paper. This proposal allows exploiting the advantages of two powerful software tools, `Dymola` for modelling complex dynamical physical systems and `Simulink` for designing and modelling control systems.

This methodology allows to evaluate the complete system behaviour and to verify rapidly any modification on the vehicle or in the controller, reducing time and development costs.

The proposed methodology is not limited only to develop EV controllers. It can be extended to develop controllers for any complex physical system.

Further research works will improve the `Dymola` model with the inclusion of the power electronics model and certain non-modelled phenomena. Additionally, sensors will be installed in the prototype in order to develop advanced strategies for traction control using the methodology presented in this work.

References

Alam, M.S. and Gao, D.W. (2006) 'Conversion of a golf car into a hybrid fuel cell Li-ion battery vehicle and validation of the vehicle model for the 2006 GEM eL Golf car using PSAT', *IEEE International Conference on Industrial Technology, 2006 (ICIT 2006)*, pp.1224–1229.

Butler, K.L., Ehsani, M. and Kamath, P. (1999) 'A Matlab-based modeling and simulation package for electric and hybrid electric vehicle design', *IEEE Transactions on Vehicular Technology*, Vol. 48, No. 6, pp.1770–1778.

Cheng, Y., Van Mierlo, J. and Lataire, P. (2008) 'Test platform for hybrid electric vehicle with the super capacitor based energy storage', *International Review of Electrical Engineering (IREE)*, Vol. 3, No. 3, pp.466–478.

Dempsey, M. (2006) 'Dymola for multi engineering modelling and simulation', *IEEE Vehicle Power and Propulsion Conference, 2006 (VPPC '06)*, pp.1–6.

Fritzson, P. and Bunus, P. (2002) 'Modelica – a general object-oriented language for continuous and discrete-event system modeling and simulation', *35th Annual Simulation Symposium*, April, pp.365–380.

Gao, D.W., Mi, C. and Emadi, A. (2007) 'Modeling and simulation of electric and hybrid vehicles', *Proceedings of the IEEE*, Vol. 95, No. 4, pp.729–745.

Gawthrop, P.J. and Bevan, G.P. (2007) 'Bond graph modeling', *IEEE Control Systems Magazine*, Vol. 27, No. 2, pp.24–45.

Gökdere, L.U., Benlyazid, K., Dougal, R., Santi, E. and Brice, C. (2002) 'A virtual prototype for a hybrid electric vehicle', *Mechatronics*, Vol. 12, No. 4, pp.575–593.

Hoffmann, K., Heeler, F. and Abel, D. (2006) 'Rapid control prototyping with dymola and matlab for a model predictive control for the air path of a boosted diesel engine', *E-COSM – Rencontres Scientifiques de l'IFP*, pp.33–40.

Hou, J. and Guo, X. (2008) 'Modeling and simulation of hybrid electric vehicles using HEVSIM and ADVISOR', *IEEE Vehicle Power and Propulsion Conference, 2008 (VPPC '08)*, pp.1–5.

Huang, A.R.W. and Chihsiuh, C. (2003) 'A low-cost driving simulator for full vehicle dynamics simulation', *IEEE Transactions on Vehicular Technology*, Vol. 52, No. 1, pp.162–172.

Junco, S. (1993) 'Real and complex power bond graph modeling of the induction motor', *International Conference on Bond Graph Modeling and Simulation (ICBGM)*, pp.323–328.

Krishnan, R. (2001) *Electric Motor Drives: Modeling, Analysis, and Control*, Prentice Hall, Upper Saddle River, New Jersey.

Magallán, G.A., De Angelo, C.H. and Garca, G.O. (2009) 'A neighborhood electric vehicle development with individual traction on rear wheels', *Int. Journal of Electric and Hybrid Vehicles (IJEHV)*, Vol. 2, No. 2, pp.115–136.

Mathworks (2009) *Simulink Getting Started Guide*, The Mathworks, Inc., MA, USA.

Monti, A., Santi, E., Dougal, R.A. and Riva, M. (2003) 'Rapid prototyping of digital controls for power electronics', *IEEE Control Systems Magazine*, Vol. 18, No. 3, pp.915–923.

Onoda, S. and Emadi, A. (2004) 'PSIM-based modeling of automotive power systems: conventional, electric, and hybrid electric vehicles', *IEEE Transactions on Vehicular Technology*, Vol. 53, No. 2, pp.390–400.

Pacejka, H.B. and Sharp, R.S. (1991) 'Shear force development by pneumatic tyres in steady state conditions: a review of modelling aspects', *Vehicle Systems Dynamics*, Vol. 20, Nos. 3–4, pp.121–176.

Ramaswamy, D., McGee, R., Sivashankar, S., Deshpande, A., Allen, J., Rzemien, K. and Stuart, W. (2004) 'A case study in hardware-in-the-loop testing: development of an ECU for a hybrid electric vehicle', *SAE 2004 World Congress & Exhibition*, pp.2004-01-0303.

Rizzoni, G., Guzzella, L. and Baumann, B.M. (1999) 'Unified modeling of hybrid electric vehicle drivetrains', *IEEE/ASME Transactions on Mechatronics*, Vol. 4, No. 3, pp.246–257.

Sanchez, J.J.C. and Garcia, J.J.R. (1993) 'A new approach on the modelling of multibody systems using multibond graphs', *American Control Conference*, June, pp.2606–2610.

Silva, L.I., Magallán, G.A. and De Angelo, C. (2008) 'Vehicle dynamics using multi-bond graphs: four wheel electric vehicle modeling', *34th Annual Conference of IEEE Industrial Electronics, 2008 (IECON 2008)*, November, pp.2846–2851.

Simic, D. and Bäuml, T. (2008) 'Implementation of hybrid electric vehicles using the VehicleInterfaces and the SmartElectricDrives libraries', *6th International Modelica Conference*, pp.557–563.

Zimmer, D. (2006) 'A modelica library for multibond graphs and its application in 3D-mechanics', Master thesis, ETH Zrich.

Appendix

Table 1 Vehicle parameters

Characteristic	*Value*
Vehicle mass	670 kg
Tyres, type and size	145 70 R13 S
Distance between wheels	1.10 m
Distance from the CG to the frontal axis	1.02 m
Distance from the CG to the rear axis	0.68 m
Tyre radio (with no load)	0.268 m
Sprung mass (w/wheel)	38.42 kg
Tyre vertical rigidness	150,000 N/m
Wheel inertia	0.4 kgm^2
Friction coefficient	483 Ns/m
Suspension rigidness	23,600 N/m
Yaw inertia	352 kgm^2
Pitch inertia	356 kgm^2
Roll inertia	152 kgm^2
Aerodynamic coefficient	0.5
Frontal area	1.4 m^2
Air density	1.225 kg/m^3

Table 2 IM parameters

Characteristic	*Value*
Rated power	3 kW 4 HP
Rated line voltage	28 V_{rms}
Rated current	81.56 A_{rms}
Rated frequency	50 Hz
Poles	4
Rated speed	1,500 rpm
Efficiency	0.85
Stator resistance (R_s)	0.010476 Ω
Magnetising inductance (L_m)	0.0154 Hy
Leakage inductance (L_{ls})	8.903×10^{-5} Hy
Rotor resistance (R_r)	0.022231 Ω