UNIVERSITÀ DELLA CALABRIA

UNIVERSITA' DELLA CALABRIA

Dipartimento di Ingegneria Meccanica, Energetica e Gestionale - DIMEG

Scuola di Dottorato

"Pitagora"

Dottorato di Ricerca in Ingegneria Meccanica

CICLO

XXVII

Titolo della tesi

EFFICIENT MODELLING METHODOLOGIES FOR MULTIBODY SIMULATIONS OF VEHICLE DYNAMICS

Settore Scientifico Disciplinare: ING-IND/13 Meccanica Applicata alle Macchine

Coordinatore: Ch.mo Prof. Leonardo Pagnotta

Supervisori: Prof. Domenico Mundo

Ing. Marco Gubitosa

Dottorando: Ing. Mariano Carpinelli

Mariano Carpinelli.

UNIVERSITÀ DELLA CALABRIA

Efficient modelling methodologies for multibody simulations of vehicle dynamics

by

Mariano Carpinelli

Scuola di Dottorato
PITAGORA

in the

Dipartimento di Ingegneria Meccanica, Energetica e Gestionale - DIMEG

Acknowledgements

The research presented in this dissertation has been included in the research program and, in part, financed by the Marie Curie FP7-IAPP project 'INTERACTIVE' (Innovative Concept Modelling Techniques for Multi-Attribute Optimization of Active Vehicles), G.A. no. 285808. I would therefore gratefully acknowledge the European Commission for this unique training and professional opportunity.

In addition, I wish to express my gratitude to my tutor Prof. Domenico Mundo, my co-tutor Ing. Marco Gubitosa and to Dr. Stijn Donders for their guidance and help during the past years.

Contents

List of Figures

List of Tables

Abbreviations

CAE	**C**omputer **A**ided **E**ngineering
ECU	**E**lectronic **C**ontrol **U**nit
RT	**R**eal **T**ime
DAE	**D**ifferential **A**lgebraic **E**quations
LI	**L**inearly **I**mplicit
EOM	**E**quations **O**f **M**otions
CG	**C**enter of **G**ravity
NVH	**N**oise **V**ibration **H**arshness
K&C	**K**inematic & **K**ompliance
HIL	**H**ardware **I**n the **L**oop
MIL	**M**an **I**n the **L**oop
ODE	**O**rdinary **D**ifferenatial **E**quations
SIL	**S**oftware **I**n the **L**oop
DOF	**D**egree **O**f **F**reedom
LHS	**L**eft **H**and **S**ide
GUI	**G**raphic **U**ser **I**nterface
CAS	**C**omputer **A**lgebra **S**oftware
IC	**I**nstant **C**enter of rotation
RHS	**R**ight **H**and **S**ide
RMS	**R**oot **M**ean **S**quare
PSD	**P**ower **S**pectral **D**ensity
FL	**F**ront **L**eft
RL	**R**ear **L**eft

To my Family

Chapter 1

Introduction

1.1 Research Motivations

The design of modern passenger cars is a complex process which requires a systematic use of CAE (Computer Aided Engineering) techniques in order to guarantee the delivery of high-quality products in the shortest possible amount of time. Multibody simulations play a major role in the design process, and they are employed in several development areas such as ride & handling performances optimization and development and validation of active suspension systems. Active and semi-active control systems are massively employed to enhance the ride and handling performances of modern passenger cars as well as to guarantee a safe and reliable dynamic response of the vehicle when dealing with dangerous situations. Because of the complex interaction between the vehicle's mechanical components and the control systems elements, i.e. ECUs (Electronic Control Units), sensors and actuators, the CAE design of the overall mechatronic system has a crucial role in the development of new products. This research work aims at improving the state-of-the-art techniques in efficient multibody multibody simulations for particular applications such as the *validation* and the *design* of suspension systems equipped with active control systems.

RT (Real-time) multibody simulations constitute the basic tool for the *validation* of active suspension systems. The RT integration of the EOM (Equations of Motion) describing complex mechanical systems requires the employment of proper integration schemes which must be able to handle the time integration at each time step in less than an *a priori* fixed sampling time interval. The LI (Linearly Implicit) Euler method has been successfully employed for the RT integration of large stiff systems of DAE (Differential Algebraic Equations) which typically arise from the complex mechanical systems of interest in practical applications. The current demand of automotive industry is to further increase the degree of complexity of multibody models employed in RT applications for the validation of active control systems, pushing researchers to improve the efficiency of the currently available integration methods. In this research work we investigate the improvements in the efficiency of the LI Euler method coming from the conversion of the EOM at each time step from a dependent to an independent coordinates formulation.

Despite the great increase in the computational power available to perform multibody simulations, there are still several applications which require the employment of extremely simplified vehicle models. As an example, simplified concept models are employed in the early stages of the design process when there are only few geometric and component data available, and the definition of detailed multibody models is not feasible. Moreover, simplified multibody models of the vehicle are employed in iterative applications such as optimization and parameters identification, which may require running a high number of simulations. Finally, the *design* of active control systems is still based on extremely simplified suspension models since they are more suitable to set up and tune the logic of the controllers which will be later implemented in the ECUs. The model reduction approach, i.e. the simplification process which leads to simple and efficient multibody models starting from detailed representation of the vehicle, generates two fundamental open issues in the current industrial practise: the inappropriateness of the simplified suspension models in reproducing all the relevant dynamic phenomena actually occurring in a real suspension and the weak link between those simplified models and the detailed multibody models developed for the fine tuning of ride and handling performances. In this work the first of the aforementioned problems is addressed by proposing a trailing-arm multibody concept model of the suspension which is able to properly reproduce both the vertical ride behaviour and the dynamic phenomena occurring during forward acceleration and braking manoeuvres. A parameters identification process is then proposed in order to obtain the unknown design parameters of the trailing-arm concept model starting from the knowledge of the dynamic response of a detailed reference multibody model of the suspension. By using this approach, the design modifications which are constantly adopted in order to tune the ride and handling performances of detailed multibody suspension models can be easily reflected into the proposed trailing-arm concept models and taken into account also in the design of the control logics thus leading to an integrated mechatronic design of the active suspension systems.

1.2 Manuscript overview

The manuscript is organized as follows. In Chapter 2 the state-of-the-art on efficient multibody simulations is first analysed in order to clarify the objective of the present research. The main aspects influencing the efficiency of the multibody simulation are discussed, i.e. the multibody formulation, the numerical integration and the model complexity.

In Chapter 3 the dependent coordinates formulation is described in details and employed to derive the classical implementation of the LI Euler method. An alternative implementation of the LI Euler method is then proposed, which exploits a velocity transformation approach to automatically switch from a dependent to an independent coordinates representation of the EOM of the multibody system. Finally, a non-iterative projection method is described, which allows to minimize the drift from the algebraic constraint conditions using a fixed number of operations at each time step.

In Chapter 4, the two implementations of the LI Euler method presented in Chapter 3 are employed to integrate the EOM describing an industrial rear left multilink suspension of a passenger car. All the relevant aspects of the modelling process are first discussed, and then the integration performances are presented with a particular emphasis on the advantages and drawbacks of the proposed implementation of the LI Euler method.

Chapter 5 proposes a model reduction approach which can be employed to transform a detailed suspension system into an equivalent simplified model. The simplified model is able to accurately reproduce the vertical ride performances and the anti-lift/dive/squat properties of the original detailed system thanks to a trailing-arm representation of the suspension. The quarter-car suspension problem is first addressed and then extended to the full vehicle. Moreover, an identification process is proposed, which allows to identify the design parameters of the concept model starting from the knowledge of the dynamic response of the target model.

Chapter 2

State-of-the-Art on Efficient Multibody Simulations

2.1 Multibody simulations in the automotive industry

Nowadays multibody simulations constitute a crucial tool in the development of new products in the automotive industries. As any other tool employed in the engineering process, multibody simulations are expected to help delivering better products in an ever decreasing amount of time, in order to ensure their competitiveness in the global market. *Less time* and *better quality* are the guidelines which underlay almost any process or action in the design of a new vehicle. Multibody simulations are directly affected by these two contrasting requirements in a variety of different applications which are all crucial in the development of a successful product.

A wide interpretation of the *less time* requirement from a multibody simulations point of view is the need for specific techniques and processes which can improve the speed of the whole design cycle. In the following list, the main multibody simulations based processes currently employed in the development o new vehicles are reported:

- **early stage concept modelling**: it is based on the employment of concept models techniques in the early stages of the design process. These techniques exploit extremely simplified multibody models of the full vehicle and its core subsystems to find out a first set of optimized values for the major design parameters, such as masses and CG (Center of Gravity) positions, suspensions rates and damping properties etc., to speed up the convergence towards an optimal final product in the subsequent design stages.

- **off-line tuning of vehicle dynamics performances**: this is accomplished by using detailed multibody models both at the components and at the full vehicle levels. These models are employed for the fine tuning of the vehicle subsystems attributes, e.g. the K&C (Kinematics & Compliance) characteristics of the suspension, and of the global handling and ride performances of the vehicle. The extensive use of detailed multibody models allows a CAE based design of the vehicle, with a tremendous reduction in the time required to achieve an optimal design configuration which would otherwise been achieved by means of expensive and time consuming tests on real prototypes.
- **loads prediction for NVH (Noise Vibration Harshness) and fatigue analysis**: high fidelity multibody models are employed to obtain the time histories of suspension's loads in particular driving scenarios. These loads are then furnished to Finite Element analysts in order to determine the NVH characteristics of the vehicle and to perform fatigue analysis on critical suspension components.
- **vehicle simulators**: the primary use of these devices in the automotive industry is for preliminary subjective tests of the vehicle performances. As shown in Fig. 2.1 a vehicle simulator is constituted by a moving platform which is actuated in order to reproduce the vehicle motion resulting from the interaction of a human driver with a virtual simulation scenario. This is achieved by simulating in real time the vehicle behaviour by means of a multibody model which receives in input the commands from the human driver. The test on simulators are often used to check the effectiveness of the vehicle tuning obtained with off-line multibody techniques.
- **tuning of active control systems**: this task is achieved by using extremely simplified multibody models of the full vehicle or of its subsystems. Due to their low complexity level, which directly traduces in a low number of states, these models are particularly indicated to set up and tune the control logics at the basis of the active control systems installed on the vehicle.
- **validation of active control systems**: this phase has a crucial importance due to the increasing influence of active control systems on the dynamic behaviour and on the safety characteristics of modern passenger cars. The correct interaction between the various elements of the active control system, i.e. the control logic implemented in the ECU, the sensors and actuators and the vehicle's mechanical system itself must be verified before the realization of the first prototype in order to avoid expensive and time consuming late modifications. This is done by connecting a part of the real hardware of the control system (see Fig. 2.2) to the multibody model of the vehicle which simulates the real behaviour of the car.

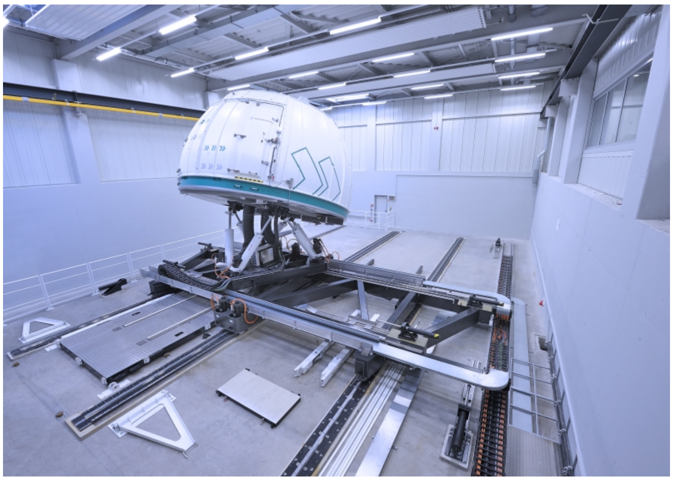


Figure 2.1: FKFS Stuttgart Motion Vehicle Simulator [1]

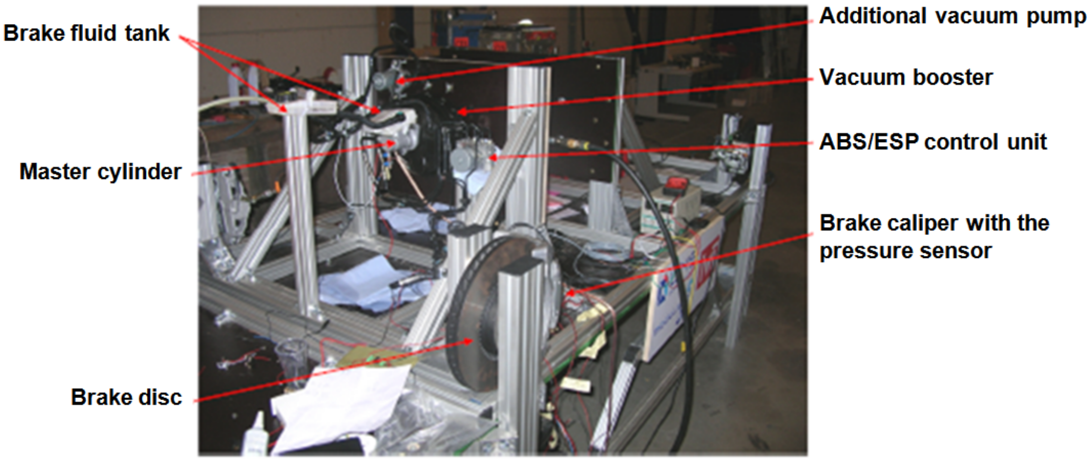


Figure 2.2: HIL set up for testing of an ABS/ESP control unit and brake system

When dealing with multibody simulations, the *better quality* requirement can be translated in *results accuracy*, i.e. the ability of the simulation to deliver results which are as representative as possible of the real phenomenon which has to be simulated. The accuracy of the multibody simulation increases with the complexity of the models employed. However the simulation's accuracy must not be seen as an absolute concept since its acceptable/possible level may significantly vary according to the particular application. In the following list several areas of the vehicle design process are grouped according to the degree of accuracy currently provided by the multibody simulations on which they are based:

- **low accuracy level**

  - *initial optimization and benchmarking analysis*: at this stage the amount of data available for the definition of multibody models is very low. These data are generally carried on from previous models having analogous characteristics to the new vehicle which must be designed. Given the poor underlying database these models are expected to furnish accurate results only for a limited number of global characteristics.

- *design of controllers*: for these applications, extremely simplified multibody models are deliberately taken into account for the design of control laws because of their low complexity level. This means that also in the advanced stages of the design cycle, when detailed geometrical and component data are available, the design of the active controllers is still carried on by means of very simplified models which often correlate very poorly with detailed multibody models used in other design areas.

- **medium accuracy level**

  - *validation of active control systems*: ideally, for these applications, the accuracy level furnished by the multibody simulations should be extremely high in order to be able to detect all possible malfunctioning coming from complex interactions between ECUs, sensors, actuators and mechanical system. However, since these simulations are based on the connection of a piece of real hardware with a virtual multibody model of the remaining part of the vehicle, there are stringent requirements on the speed of the simulations which, as it will be explained in the following sections, prevents the use of extremely detailed multibody models. Several model reduction techniques are then used to decrease the complexity of detailed multibody models in order to allow their employment in HIL (Hardware In the Loop) tests for the validation of active control systems as it will explained later in this Chapter.

  - *vehicle simulators*: also in this case, the multibody models whose real time integration determines the motion of the simulator, should be as accurate as possible in order to feedback to the human driver the most realistic excitations. Once again the real time requirements compel the use of simplified multibody models in vehicle simulators, which limits the effectiveness of these potentially extremely powerful tools.

- **high accuracy level**

  - *fine tuning of components level attributes and of the overall ride and handling performances*: for these tasks extremely detailed multibody models of the vehicle and its subsystems are employed, in order to predict the characteristics of the real vehicle with the highest possible degree of confidence. These detailed models are generally composed by hundreds of bodies, most of them modelled as flexible components. The non-linear characteristics of bushings elements, bump and rebound stops, coil and gas springs are determined experimentally and inserted into the model whose static and dynamics characteristics are in turn correlated against objective measurements. The development and validation of these models is a

difficult and time consuming task but it greatly speeds up the subsequent tuning phase. Indeed, by using these models, vehicle dynamics engineers can test in a virtual environment different settings in order to meet the target performances of the particular vehicle under analysis.
- *loads prediction for NVH and fatigue analysis*: for these applications the highest level of accuracy is requested in the evaluation of suspension's loads. Indeed for NVH and fatigue studies, the accuracy of the models must be guaranteed up to several hundreds Hz.

Multibody simulations are thus a crucial tool in the development of new vehicles, and there are still several areas in which their contribution and their effectiveness can be enhanced in order to improve the overall speed of the design process.

Among them, the issue of avoiding the model simplification process which is currently mandatory in order to achieve real time simulations for HIL validation of active control systems and for MIL (Man In the Loop) tests in vehicle simulators, is one of the most addressed by the multibody research community.

Another open issue, which may potentially decrease the effectiveness of the design of active control systems, is the poor correlation between the simplified models employed in the design of active controllers and the detailed off-line multibody models which are constantly developed and tuned in order to meet the target performances of the vehicle, e.g. its ride and handling attributes. The same correlation issue may be recognized between the simplified concept models employed for initial optimization and benchmarking studies and their predecessors, i.e. the models coming from previous projects which share the same main characteristics with the new vehicle to be developed.

Moreover, by keeping the low complexity level of multibody models employed in the initial concept design phase and in the design of controllers, it would be extremely beneficial to enhance the accuracy level of the simulations by enabling the simplified models to capture more physical phenomena then they currently do.

These open issues will be addressed in the remaining of this manuscript and possible solutions will be developed. As it will become clear in the following sections, the efficiency of the multibody simulation plays a key role in the resolution of these problems. The major factors whose interaction determines the efficiency of the multibody simulation are the **coordinates formulation**, the **numerical integration** and the **model complexity**, as it will be described in the next 3 sections with a particular emphasis on real-time applications.

2.2 Multibody Formulations

2.2.1 Dendependent Coordinates Formulations

The modelling of the multibody system is a crucial factor in the overall efficiency of the dynamic simulation [2, 3]. It involves the selection of a proper coordinates representation to describe the position and orientation of each body in the mechanical system. The 3 most used sets of dependent configuration parameters, whose employment leads to a set of equations of motion expressed in a *descriptor form*, are: the *Cartesian* coordinates, the *relative* coordinates and the *natural* coordinates.

The *Cartesian* coordinates formulation constitutes the basis of many commercial multibody software thanks to its ease of implementation and generality [4–6]. The position and the orientation of each body is uniquely determined by 3 Cartesian coordinates of a reference point, which generally coincides with the CG of the body itself, and by 3 independent angles or 4 dependent Euler parameters describing the orientation of the body reference frame with respect to the inertial frame [7, 8]. The use of the *Cartesian* coordinates formulation results in a large set of loosely coupled EOMs.

The efficiency in the solution of the EOMs can be greatly improved by means of formulations based on *relative* coordinates where the topology of the mechanical system is taken into account by specifying the position of each body in relation with the previous body in the kinematic chain. Historically, the *relative* coordinates formulation was the first one to be use in order to implement a multibody computer code. However this approach requires more intelligence in the preprocessor than the *Cartesian* coordinates formulations, and gives birth to recursive schemes for the assembly of the multibody system of equations [9–11].

The *natural* coordinates formulation is based on the idea that the mechanical system can be represented as a collection of points and vectors [3, 7, 12, 13]. The configuration of the mechanical system is thus defined by the positions of these points which, in turn, are related among them by constant distance constraints coming from the rigid body assumption. As described in [7] the *natural* coordinates formulation may be numerically more efficient than the *Cartesian* coordinates formulation since it takes into account the topology of the mechanical system. However it is not as suitable as the *Cartesian* coordinates formulation for an implementation into a general purpose multibody software and its employment in industrial applications is thus very limited.

In the general case, all the dependent coordinates formulations described above require the definition of algebraic constraint equations in order to numerically model the joint connections in the mechanical system. When the classical method of *Lagrange Multipliers* is employed to obtain the equations of motion of the system [2, 7], a set of Differential-Algebraic equations of index 3 is obtained. The index of the DAE set is generally reduced to 2 and 1 since there are no obvious way to handle the numerical integration of an index 3 DAE system. However this process, as it will be explained in Chapter 2, relies on the differentiations of the constraint equations and thus results in the loss of the constant terms employed to define the constraint conditions. This results in a numerical instability which causes the numerical solution to progressively violate the initial constraint conditions requiring the use of proper stabilization methods to avoid it. Alternatively, the *Penalty Formulation* introduced by Bayo et al. in [14] can be used to assemble the equations of motion of the system. In this formulation the Lagrange multipliers are eliminated from the equations of motion by directly incorporating the constraint conditions as dynamical systems penalized by a large penalty factor [2, 7].

The dependent *Cartesian* coordinates formulation associated with the Lagrange multipliers method represents the most used strategy in the developing of commercial multibody software. For this reason in this research work an approach will be proposed, as it will be described in Chapters 3 and 4, in order to improve its efficiency when dealing with real time simulations of vehicle dynamics. This approach is strictly related to the formulation in independent coordinates which will be introduced in the next section.

2.2.2 Independent Coordinates Formulation

When an independent coordinates formulation is used, the numerical integration is performed only on a limited number of independent coordinates whose number coincides with the number of degrees of freedom of the mechanical system. By using these formulations it is possible to obtain a set of equations of motion expressed in a *state space form*, i.e. only in terms of the independent coordinates. The most known independent coordinates formulations are the coordinates partitioning method [15], the Kane's method [16] and the virtual power with projection matrices method [17]. A state-space representation of the equations of motion can be also obtained using the so called *velocity transformation* methods [18–21].

The great advantage of an independent coordinates formulation is the big reduction in the number of coordinates which must be integrated. Moreover, since the constraint equations are no longer present in the equations of motion, the numerical problems associated to the integration of DAE systems are avoided and classical integration methods for ODE (Ordinary Differential Equations) can be employed.

The main drawback of independent formulations is that they require the computationally expensive solution of the position and velocity problems at each time step in order to univocally determine the configuration of the system. To avoid the numerically inefficient solution of the position problem at each time step, an extended set of differential equations may be solved [22] which takes into account all the velocities and not only the independent ones. In this way the dependent positions at the next time step are directly furnished by the numerical integration process. Finally, independent coordinates formulations are generally less robust then dependent coordinates formulations when dealing with singular configurations and their implementation is much more involved.

As it will be described in Chapters 3 and 4, the velocity transformation method proposed by de Jalon et al in [7] will be employed in this research work to transform the equations of motion expressed in dependent Cartesian coordinates to a state-space form in order to improve the efficiency of the integration process.

2.3 Numerical Integration

2.3.1 Requirements for real time simulations

As pointed out before, HIL and MIL as well as SIL (Software In the Loop) applications, have become standard tools for the design and the development of new vehicles [23–27]. These simulations impose stringent requirements in terms of integration performances. This is dictated by the fact that real hardware must be interfaced to the mathematical models of the full vehicle or of one of its subsystems (e.g. suspension system and steering system) as shown in Fig. 2.3.

The ECUs represent a typical example of the real hardware which is usually tested by means of HIL simulations in automotive applications. Since their control logics are generally not available for commercial reasons, a full virtual prototyping of the mechatronic system is not feasible, causing HIL simulations to be the only tool for analysing the correct interaction between the control system and the mechanical system. Moreover, to accomplish a proper CAE driven design of the vehicle, the test of control units and active control systems must start long before the realization of the first physical prototype of the vehicle. In this case the only possible strategy is to replaced the vehicle by its virtual representation, i.e. a multibody model, thus enabling engineers to test the interaction between controls and mechanical system already in the early stages of the design process. As shown in Fig. 2.3 the numerical model of the vehicle is connected to the real test environment through sensors and actuators signals. The frequency rate of the hardware embedded in the HIL simulation determines the frequency at which

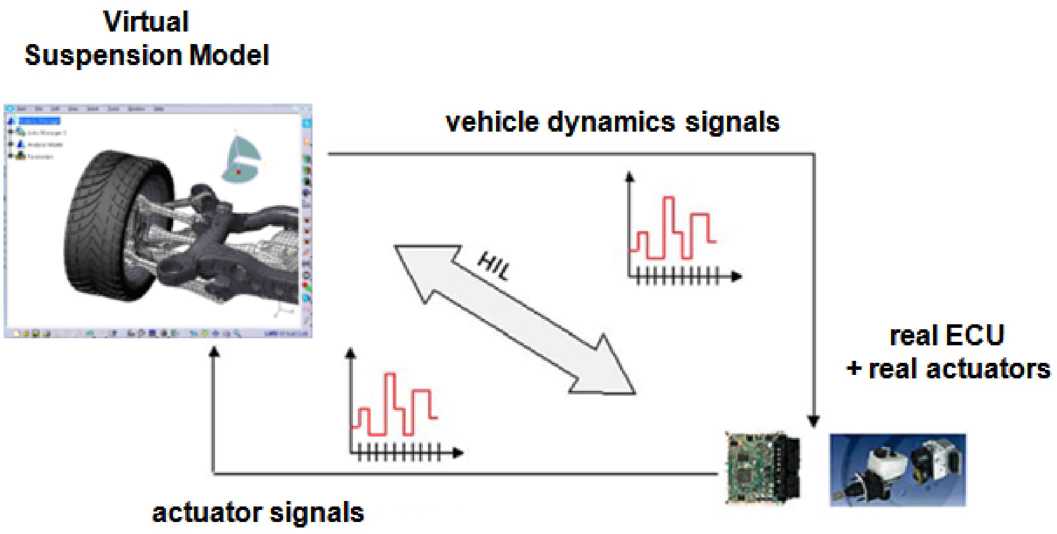


FIGURE 2.3: HIL simulations scheme [30]

signals must be exchanged with the mathematical multibody model of the mechanical system, the current standard for automotive applications being 1 kHz. This determines the crucial requirement for HIL simulations, i.e. the integration process must provide the states of the mechatronic system at the current time step in less than 1 ms.

For a reliable real time simulation to take place, the number of operations carried out by the integration algorithm in each time step must be known a priori. Implicit schemes do not provide such a property since they compute the states of the system at the next time step by iteratively solving a non-linear system of equations. On the contrary explicit integration methods rely on a fixed number of operations at each time step. Among them, low order schemes are generally preferred in RT applications due to the low number of operations executed at each time step. For this reason the explicit Euler method has been extensively used in HIL simulations. However it can be only used for the time integration of non-stiff ODE systems, i.e. only the integration of the EOM related to simplified vehicle models without stiff force elements and ideal joints connections can be addressed using this method.

Numerical stiffness is an intrinsic characteristics of multibody models employed in vehicle dynamics applications. It is often generated by the use of very stiff elastic bushings elements in order to model the suspension's compliance [28, 29]. Indeed, connecting the low mass suspension links by means of stiff force elements results in high frequency vibrations which oppose to the slow dynamics of the remaining part of the system causing numerical stiffness. Numerical stiffness in the EOM can be also caused by the modelling of subsystems with fast dynamics like electric circuits and controllers.

As pointed out in the previous section, the use of dependent coordinates formulations with the Lagrange multipliers method leads to a set of Differntial-Algebraic equations whose index must be reduced from 3 to 2 and 1 by differentiating the constraint equations. This allows the use standard numerical integrators but, at the same time, introduces the issue of a proper stabilization of the constraint equations in order to reduce the numerical drift from the original constraint conditions. The stabilization of the constraint equations must be also taken into account when addressing the real-time integration of the equations of motion in vehicle dynamics applications.

Summarizing, the real time requirements which must be fulfilled by the numerical integrators are:

- *low turnaround time*: the time required to complete all the operations in one time step must be lower then the communication step which is standardised as 1 ms;
- *deterministic*: the number of operations to be performed at each time step must be known *a priori*;
- *suitable for stiff numerical problems*: the solver must be able to handle the numerically stiff EOM describing complex vehicle dynamics models with stiff bushing elements;
- *suitable for DAE problems*: the integrator must be able to minimize the drift from the constraint conditions resulting from the index reduction of the DAE describing constrained vehicle dynamics models;

2.3.2 Linearly Implicit Euler method

The linearly implicit Euler method meets all the requirements imposed by real time simulations. It combines the efficiency of the explicit Euler method and the stability properties of the implicit Euler method and has been successfully employed to achieve RT integration of the EOM associated to high-fidelity multibody systems containing stiff-force elements and ideal joints [30, 31]. The computational effort required by the linearly implicit Euler method is predictable since it requires the solution of a linear system of equations at each time step. Moreover it exploits a linear approximation of the stiff terms in the vector of generalized forces in order to be able to cope with stiff numerical problems. Finally, to minimize the constraint violation, a combined use of the linearly implicit Euler method with a non-iterative projection step and alternatively with a modified Baumgarte stabilization method has been proposed in [30].

A great deal of research has been carried out to enhance the performances of the linearly implicit Euler method with the final aim of make it possible the RT integration of large stiff DAE systems of equations arising from extremely complex models of real mechanical systems [28, 30, 32, 33]. In particular, research efforts have focused on the efficient evaluation of the system Jacobians which have a crucial effect on the stability and accuracy of the linearly implicit Euler method. Classical off-line techniques based on finite-differences approximations are not suitable for RT simulations of large scale models due to the high number of function evaluations they require whenever the Jacobians must be updated. Moreover, in order to guarantee the unconditional stability of the linearly implicit Euler method, only the terms related to the stiff force elements in the vector of generalized forces are required and thus the exact computation of Jacobians is not needed at all. The analytical evaluation of the Jacobians at each time step has been proposed by Rill in [34] where the choice of the stiff terms to be retained is based on physical considerations on the mechanical system. An alternative method has been proposed by Schiela et al. in [35] where, in a pre-processing step, a reference trajectory is selected for the evaluation of the Jacobians whose non-stiff terms are detected by means of numerical considerations and then brought to zero in order to enhance the factorization of the Jacobian matrix. A third approach has been used by Arnold et al. in [30] relying on a pre-processing step during which the Jacobians are evaluated for all the characteristic configurations of the mechanical system. These set of pre-evaluated Jacobians is then used during the RT simulation avoiding any additional computational effort.

2.4 Model reduction techniques

The practice of converting detailed multibody models into equivalent simplified ones is still extensively adopted in the automotive industry in order to increase the efficiency of the multibody simulations in particular applications.

Indeed, despite the continuously increasing amount of computational power available in off-the-shelf workstations, the model reduction approach still represents the only suitable solution for dealing with iterative processes which may require a very high number of simulations such as:

- *parameters identification*: these techniques are often employed in vehicle dynamics applications in order to obtain the unknown parameters of a target vehicle. This latter is represented by a simplified multibody model whose main design parameters, such as mass and inertia of the sprung mass, unsprung masses, suspensions rates and damping properties, are tuned iteratively in order to reproduce the response of the target vehicle [36–38].

- *design optimization*: a common problem in the industrial practise is the finding of an optimal set of design parameters which allows to meet certain desired vehicle performances. A typical example is the tuning of the dampers in passengers cars which may involve a considerable number of iterations. Simplified models are much more adapt then detailed ones to set up automatic optimization processes based on iterative multibody simulations [39–41].
- *tuning of controllers*: the iterative tuning of active controllers is another design area where simplified vehicle models are used extensively. In order to ensure optimal control performances, the parameters in the controllers must be tuned by running numerous tests in different operating conditions [23–27].

Moreover, standard ODE explicit solvers, first of all the explicit Euler method, are still the standard for real-time applications in the automotive industry. For the reasons explained in the previous two sections, the reduction process required to transform a high-fidelity multibody model into one suitable for integration by means of standard explicit ODE integrators is quite a cumbersome and time consuming task. It involves the preparation of proper K&C look-up tables in order to take into account the characteristics of the suspensions linkages without retaining all the suspension links and the stiff bushing connections. Kinematic closed loop (i.e. the leverage in the steering system) must be replaced by kinematic look-up tables while open kinematic loops with ideal joints connections must be expressed in relative coordinates in order to avoid DAE. This simplification process weakens the link between the detailed multibody models and their real-time counterparts. On top of that, every modification in the starting high-fidelity multibody model (e.g. a change in the positions of the suspension's hard points) requires the regeneration of the affected look-up tables in the simplified model. This requires time consuming objective tests on tailored K&C test rigs or their virtual reproduction via multibody simulations as shown in Fig. 2.4.

The research community has intensively addressed the issues of model reduction for real-time applications in the last years trying to improve the look-up tables approach [33, 42, 43]. Rulka et al. proposed an automatic approach (macro joint approach) to transform the set of DAEs describing a detailed multibody model into an equivalent set of ODEs by keeping the same full component parametrization in [44] while Eichberg et al. proposed a non-linear model reduction process in [33, 42] in order to generate real time simulations models from existing detailed multibody models by neglecting the high frequencies contributions in handling models but considering the non-linear compliant kinematic effects in the suspensions by means of look-up tables.

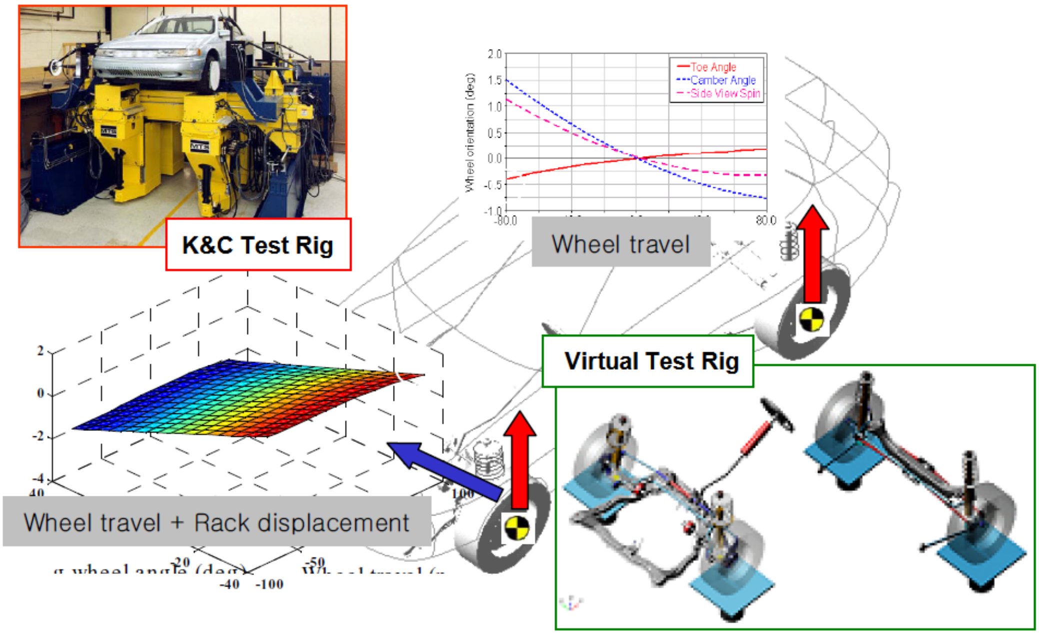


FIGURE 2.4: K&C look-up tables [45]

Summarizing, model reduction techniques constitute a fundamental tool to enhance the efficiency of multibody simulations for vehicle dynamics applications such as parameters identification, design optimization, tuning of controllers and real-time simulations.

2.5 Research Contribution

From the analysis of the state-of-the-art on efficient multibody simulations conducted in this Chapter, the following areas have been identified as possible targets for further investigations:

1. enhancement of the performances of real-time solvers for stiff DAE systems by combining the generality and ease of implementation of the dependent *Cartesian* coordinates approach and the efficiency of a state-space formulation obtainable by means of *velocity transformation* techniques;
2. elimination of look-up tables for the representation of the non-linear kinematics of the suspension systems;
3. automatic transformation of detailed off-line suspension models into simplified ones suitable for design optimization, tuning of controllers and real-time simulations;
4. set-up of an accurate and reliable identification process to find out the unknown design parameters of reduced suspension models in order to enhance their correlation with detailed off-line models and real prototypes.

In particular, in order to address the first of these topics, an investigation has been carried out on the possible efficiency improvements in the linearly implicit Euler method associated to an automating switching from a dependent to an independent coordinates formulation. This approach can potentially improve the integration efficiency if it implies a substantial reduction in the number of DOFs (Degrees Of Freedom) retained in the multibody model. This is particularly the case for multibody systems composed by a large number of bodies whose configuration parameters are related by a large number of constraint equations. Detailed multibody models developed for automotive applications perfectly fit in this category since they are generally built using a considerable number of bodies (more then 150) and ideal joints (e.g. driveline and steering systems connections). Moreover these models are usually obtained using general purpose multibody software based on dependent coordinates formulations in which 4 dependent Euler parameters are employed to describe the orientation of each body. The use of dependent Euler parameters avoids numerical problems due to singular configurations but requires an additional constraint equation for each body in the system causing a great increase in the total number of constraint equations. The proposed approach intends exploiting these intrinsic characteristics to improve the efficiency of the standard implementation of the linearly implicit Euler method for automotive applications. In particular the implementation of the linearly implicit Euler method has been modified in order to carry out the integration process by only considering a reduced set of independent velocities instead of the complete one. To this aim we apply the projection method proposed by the Jalón et al. in [7]. By expressing the EOM of the multibody system in independent coordinates it is possible to obtain a great reduction in the dimensions of the linear system which has to be solved at each time step, since only the current independent accelerations must be found instead of the complete set of dependent accelerations. The proposed methodology has been implemented starting from a symbolic representation of the EOM of the mechanical system. Analytical expressions for the Jacobians are obtained by considering only the stiff terms in the vector of generalized applied forces and the non-iterative projection method proposed in [30] is applied as well in order to minimize the drift off of the numerical solution from the constraint conditions. The proposed implementation of the linearly implicit Euler method is then compared to the classical one based on a dependent coordinates formulation through a numerical test case in order to highlight possible advantages and drawbacks.

The second issue, that is avoiding the use of look-up tables for the modelling of non-linear suspension kinematics, has been addressed by adopting a trailing arm representation of the suspension which enables not only the analysis of the vertical ride behaviour but also the study of the dynamic phenomena occurring during longitudinal acceleration and braking manoeuvres, which are influenced by the anti-squat, anti-lift and anti-dive characteristics of the suspension.

Finally, the 3^{rd} and 4^{th} open issues have been addressed by the implementation of a parameters identification process which allows to estimate the unknown parameters of the trailing arm concept model, both in its quarter car and full vehicle versions, starting from the knowledge of the dynamic response of a reference model. The method can be applied to map detailed off-line multibody models into equivalent simplified trailing arm models which can then be employed in applications such as design optimization, tuning of controllers and real-time simulations. Moreover the identification method has been designed in order to be able to receive in input, as the reference dynamic response, measured data coming from experimental tests on a target vehicle. This is particularly useful when competitors vehicles must be analysed for benchmarking studies, since their main design parameters can be easily obtained without expensive measurements set-ups.

Chapter 3

LI Euler Method: dependent and independent coordinates formulations

3.1 Dependent coordinates formulation of the EOMs of a multibody system

Multibody models are composed by several bodies whose positions and orientations must be univocally described by means of a set of configuration parameters or coordinates [2]. Formulations in dependent coordinates constitute the basis for several general purpose multibody software. When dependent coordinates are used, the position and the orientation of each body in the mechanical system are described by means of a fixed number of configuration parameters while a set of algebraic equations is defined in order to mathematically represent the constraint conditions applied among bodies. This leads to the following system of index-3 DAE:

$$\begin{aligned} \mathbf{f}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}},\lambda,t) = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + {\boldsymbol{\Phi}_{\mathbf{q}}}^{T}(\mathbf{q},t) - \mathbf{Q}(\mathbf{q},\dot{\mathbf{q}},t) = \mathbf{0} \\ \boldsymbol{\Phi}(\mathbf{q},t) = \mathbf{0} \end{aligned} \tag{3.1}$$

In Eq. 3.1 **M** and **Q** represent the mass matrix and the vector of generalized applied forces respectively. This latter contains the gravity force and the force elements applied between bodies, which in vehicle dynamics applications consist generally in tires and shock-absorbers as well as elastic bushing connections and aerodynamics loads. The

n dependent configuration parameters univocally describing the configuration of the multibody system and their time derivatives are indicated as $\mathbf{q}$ and $\dot{\mathbf{q}}$ respectively. Vector $\mathbf{\Phi}$ contains the constraint equations whose number is denoted as m, the constraint Jacobian matrix $\mathbf{\Phi_q}$ groups the derivatives of the constraint equations with respect to the configuration parameters and finally λ is the vector of Lagrange multipliers. Due to the modelling of bushing connections and shock absorbers bump and rebound stops the vector of generalized applied forces in Eq. 3.1 contains stiff terms which lead to a stiff index-3 DAE system whose direct integration cannot be approached by means of standard ODE integrators. For this reason the index of the DAE system is generally reduced to 2 and 1 by substituting the constraint equations at the position level in Eq. 3.1 for the constraints at the velocity coordinates level:

$$\mathbf{\Phi_q}\dot{\mathbf{q}} + \mathbf{\Phi_t} = \mathbf{0} \tag{3.2}$$

and for the constraints at the acceleration coordinates level:

$$\mathbf{\Phi_q}\ddot{\mathbf{q}} + \dot{\mathbf{\Phi}}_\mathbf{t} + \dot{\mathbf{\Phi}}_\mathbf{q}\dot{\mathbf{q}} = \mathbf{0} \tag{3.3}$$

In Eqs. 3.2 and 3.3 the vector $\mathbf{\Phi_t}$ contains the time derivative of the constraint equations while $\dot{\mathbf{\Phi}}_\mathbf{q}$ is the time derivative of the constraint Jacobian matrix. Since the differentiation process cuts out the constant terms present in the constraint equations at the position level, a linear and quadratic growth of $\|\mathbf{\Phi}(\mathbf{q},\mathbf{t})\|$ can be observed in the numerical solutions obtained with the *index*-2 and *index*-1 formulations respectively. In order to avoid this drift away from the manifold defined by the constraint equations the Baumgarte stabilization technique is commonly employed [46]. In the classical Baumgarte approach, the violation in the constraint conditions during the time integration of the EOM expressed in the *index*-1 form is reduced by substituting the constraint at the acceleration level of Eq. 3.3 by a proper linear combination of the constraint conditions at position, velocity and acceleration level:

$$\mathbf{\Phi_q}\ddot{\mathbf{q}} + \dot{\mathbf{\Phi}}_\mathbf{t} + \dot{\mathbf{\Phi}}_\mathbf{q}\dot{\mathbf{q}} + 2\alpha(\mathbf{\Phi_q}\dot{\mathbf{q}} + \mathbf{\Phi}_t) + \beta^2\mathbf{\Phi} = \mathbf{0} \tag{3.4}$$

The same approach can be used to apply the Baumgarte stabilization to the time integration of the EOM expressed in the *index*-2 form [30] by replacing Eq. 3.2 with:

$$\mathbf{\Phi_q}\dot{\mathbf{q}} + \mathbf{\Phi}_t + \gamma\mathbf{\Phi} = \mathbf{0} \tag{3.5}$$

Proper numerical values of the Baumgarte parameters α, β and γ must be assigned in order to reduce the drift-off effect and to limit the introduction of artificial stiffness into the system.

3.1.1 Implicit Integration schemes

The numerical integration of the EOM expressed in the *index*-1 and *index*-2 forms is typically carried out by means of implicit integration schemes in off-line simulations [47–52]. Indeed the presence of stiff terms in the EOM imposes unpractical restrictions to the step-size if explicit integration schemes are used. In order to find the set of $3n+m$ unknowns constituted by positions, velocities, accelerations and Lagrange multipliers at the next time step, $2n$ additional integrals must be added to the EOM of Eq. 3.1:

$$\mathbf{q}_{(i+1)} = \mathbf{q}_{(i)} + \int_{t_{(i)}}^{t_{(i)}+h} \dot{\mathbf{q}} dt, \qquad \dot{\mathbf{q}}_{(i+1)} = \dot{\mathbf{q}}_{(i)} + \int_{t_{(i)}}^{t_{(i)}+h} \ddot{\mathbf{q}} dt \tag{3.6}$$

where h is the time step size and the subscripts i and $i{+}1$ indicate the current and the next time step of the integration process respectively. Implicit integration algorithms rely on the substitution of the integrals of Eq. 3.6 for integration formulas which assume the following form:

$$\mathbf{q}_{(i+1)} = \mathbf{\Lambda}(\mathbf{q}_{\leq i}, \dot{\mathbf{q}}_{\leq i}, \ddot{\mathbf{q}}_{\leq i}, \ddot{\mathbf{q}}_{\leq (i+1)}), \qquad \dot{\mathbf{q}}_{(i+1)} = \check{\mathbf{\Lambda}}(\dot{\mathbf{q}}_{\leq i}, \ddot{\mathbf{q}}_{\leq i}, \ddot{\mathbf{q}}_{\leq (i+1)}) \tag{3.7}$$

In Eq. 3.7 the dependent configuration parameters and the dependent velocities at the next time step are expressed as functions of the states at the current and previous time steps as well as functions of the accelerations at the next time step. The actual form of the integration formulas $\mathbf{\Lambda}$ and $\dot{\mathbf{\Lambda}}$ varies according to the particular implicit scheme adopted. By replacing the configuration parameters $\mathbf{q}$ and their time derivatives $\dot{\mathbf{q}}$ in Eq. 3.1 by the integration formulas of Eq. 3.7, a non-linear system having as unknowns the n accelerations and the m Lagrange multipliers at the next time step is obtained:

$$\mathbf{F}(\ddot{\mathbf{q}}_{(i+1)}, \lambda_{(i+1)}) = \begin{bmatrix} \mathbf{f}(\ddot{\mathbf{q}}_{(i+1)}, \lambda_{(i+1)}) \\ \mathbf{CEs}(\ddot{\mathbf{q}}_{(i+1)}) \end{bmatrix} = \mathbf{0} \tag{3.8}$$

where $\mathbf{CEs}(\ddot{\mathbf{q}}_{(i+1)})$ indicates the constraint equations (at the velocity or acceleration level). The non-linear system in Eq. 3.8 can be solved iteratively by the Newton-Raphson algorithm in order to find the accelerations and the Lagrange multipliers at the next time step. The k^{th} iteration of the Newton-Raphson method can be written as:

$$\begin{Bmatrix} \ddot{\mathbf{q}}^{k}_{(i+1)} \\ \lambda^{k}_{(i+1)} \end{Bmatrix} = \begin{Bmatrix} \ddot{\mathbf{q}}^{k-1}_{(i+1)} \\ \lambda^{k-1}_{(i+1)} \end{Bmatrix} - [\mathbf{J}(\ddot{\mathbf{q}}^{k-1}_{(i+1)}, \lambda^{k-1}_{(i+1)})]^{-1}\mathbf{F}(\ddot{\mathbf{q}}^{k-1}_{(i+1)}, \lambda^{k-1}_{(i+1)}) = \mathbf{0} \tag{3.9}$$

where the iteration matrix J is defined as:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial \mathbf{f}(\ddot{\mathbf{q}}_{(i+1)}, \lambda_{(i+1)})}{\partial \ddot{\mathbf{q}}_{(i+1)}} & \frac{\partial \mathbf{f}(\ddot{\mathbf{q}}_{(i+1)}, \lambda_{(i+1)})}{\partial \lambda_{(i+1)}} \\ \frac{\partial \mathbf{CEs}(\ddot{\mathbf{q}}_{(i+1)})}{\partial \ddot{\mathbf{q}}_{(i+1)}} & \mathbf{0} \end{bmatrix} \tag{3.10}$$

As previously pointed out, the basic requirement in RT simulations is that the number of operations to be carried out at each integration time step must be fixed. Indeed this is the only way to be *a priori* sure that the turnaround time, i.e. the time required to complete the operations at each time step, is lower than the time span of 1 ms imposed by RT standards [30, 35]. Due to their iterative nature implicit integration schemes are not able to fulfill this requirement. For this reason the linearly implicit Euler integration method has been proposed [30, 31, 34] which is able to handle the integration of DAE systems containing stiff terms by performing a fixed number of operations at each time step. This is achieved by means of a linear approximation of the stiff terms contained in the vector of the generalized applied forces **Q** in Eq. 3.1 as it will be explained in the next section.

3.2 Integration of the EOM expressed in dependent coordinates with the linearly implicit Euler method

By discretizing in time the differential equations of motion in Eq. 3.1 and by expressing them at the current time step i one obtains:

$$\mathbf{M}_{(i)}\ddot{\mathbf{q}}_{(i)} + \mathbf{\Phi}^{T}_{\mathbf{q}_{(i)}}\lambda_{(i)} = \mathbf{Q}_{(i)} \tag{3.11}$$

where $\mathbf{M}_{(i)} = \mathbf{M}(\mathbf{q}_{(i)}), \mathbf{Q}_{(i)} = \mathbf{Q}(\mathbf{q}_{(i)}, \dot{\mathbf{q}}_{(i)}, t_{(i)})$ and $\mathbf{\Phi}_{\mathbf{q}_{(i)}} = \mathbf{\Phi}_{\mathbf{q}}(\mathbf{q}_{(i)}, t_{(i)})$. The approximation of the accelerations at the current time step by finite differences leads to the following expression:

$$\mathbf{M}_{(i)}\frac{(\dot{\mathbf{q}}_{(i+1)} - \dot{\mathbf{q}}_{(i)})}{h} + \mathbf{\Phi}^{T}_{\mathbf{q}_{(i)}}\lambda_{(i)} = \mathbf{Q}_{(i)} \tag{3.12}$$

By combining Eq. 3.12 with the finite differences representation of the velocities at the current time step the formulas of the first order *Explicit Euler* method can be obtained:

$$\begin{bmatrix} \mathbf{q}_{(i+1)} \\ \mathbf{M}_{(i)}\dot{\mathbf{q}}_{(i+1)} + h\mathbf{\Phi}_{\mathbf{q}_{(i)}}^{T}\lambda_{(i)} \end{bmatrix} = \begin{bmatrix} \mathbf{q}_{(i)} \\ \mathbf{M}_{(i)}\dot{\mathbf{q}}_{(i)} \end{bmatrix} + h\begin{bmatrix} \dot{\mathbf{q}}_{(i)} \\ \mathbf{Q}_{(i)} \end{bmatrix} \tag{3.13}$$

It is well known that the explicit Euler method cannot be used for the stable integration of stiff ODEs or DAEs systems unless the time step is reduced to unpractical values [53]. An alternative is the use of the *implicit Euler* method which can handle the integration of ODEs or DAEs systems containing stiff terms but requires an *a-priori* unknown number of iterations to find the positions and the velocities at the next time step as the solution of the following system:

$$\begin{bmatrix} \mathbf{q}_{(i+1)} \\ \mathbf{M}_{(i)}\dot{\mathbf{q}}_{(i+1)} + h\mathbf{\Phi}_{\mathbf{q}_{(i)}}^{T}\lambda_{(i)} \end{bmatrix} = \begin{bmatrix} \mathbf{q}_{(i)} \\ \mathbf{M}_{(i)}\dot{\mathbf{q}}_{(i)} \end{bmatrix} + h\begin{bmatrix} \dot{\mathbf{q}}_{(i+1)} \\ \mathbf{Q}_{(i+1)} \end{bmatrix} \tag{3.14}$$

where $\mathbf{Q}_{(i+1)} = \mathbf{Q}(\mathbf{q}_{(i+1)}, \dot{\mathbf{q}}_{(i+1)}, t_{(i+1)})$. In order to avoid the step-size restrictions related to the explicit Euler integration scheme and the iterative solution of a non-linear system of equations associated to the implicit Euler integration scheme, the *linearly implicit Euler method* has been proposed [30, 31] for the real-time integration of the stiff DAE system of Eq. 3.1. The linearly implicit Euler method represents a suitable alternative to the explicit and the implicit Euler methods. Indeed, even if it requires a fixed number of operations for each time step it can handle the integration of stiff EOM by approximating the vector of generalized applied forces as a Taylor expansion truncated at its first term. The variations in the values of the dependent configuration parameters $\Delta\mathbf{q}_{(i)}$ and their time derivatives $\Delta\dot{\mathbf{q}}_{(i)}$ are defined by the following forward differences formulas:

$$\begin{bmatrix} \Delta\mathbf{q}_{(i)} \\ \Delta\dot{\mathbf{q}}_{(i)} \end{bmatrix} = \begin{bmatrix} \mathbf{q}_{(i+1)} - \mathbf{q}_{(i)} \\ \dot{\mathbf{q}}_{(i+1)} - \dot{\mathbf{q}}_{(i)} \end{bmatrix} \tag{3.15}$$

By obtaining $\mathbf{q}_{(i+1)}$ and $\dot{\mathbf{q}}_{(i+1)}$ from Eq. 3.15 and substituting their values in Eq. 3.14 the following expression can be obtained:

$$\begin{bmatrix} \Delta\mathbf{q}_{(i)} \\ \mathbf{M}_{(i)}\Delta\dot{\mathbf{q}}_{(i)} + \mathbf{\Phi}_{\mathbf{q}_{(i)}}^{T}\lambda_{(i)} \end{bmatrix} = \begin{bmatrix} h(\dot{\mathbf{q}}_{(i)} + \Delta\dot{\mathbf{q}}_{(i)}) \\ h\mathbf{Q}(\mathbf{q}_{(i)} + \Delta\mathbf{q}_{(i)}, \dot{\mathbf{q}}_{(i)} + \Delta\dot{\mathbf{q}}_{(i)}, t_{(i+1)}) \end{bmatrix} \tag{3.16}$$

As previously anticipated the linearly implicit Euler method relies on the first order approximation of the vector of generalized applied forces in the LHS (Left Hand Side) of Eq. 3.16 by means of the following Taylor expansion:

$$\mathbf{M}_{(i)}\Delta\dot{\mathbf{q}}_{(i)} + \mathbf{\Phi}_{\mathbf{q}_{(i)}}^{T}\lambda_{(i)} = h[\mathbf{Q}_{(i)} + \frac{\partial \mathbf{Q}}{\partial \mathbf{q}}\bigg|_{t(i)}\Delta\mathbf{q}_{(i)} + \frac{\partial \mathbf{Q}}{\partial \dot{\mathbf{q}}}\bigg|_{t(i)}\Delta\dot{\mathbf{q}}_{(i)} + \frac{\partial \mathbf{Q}}{\partial t}h] \tag{3.17}$$

By substituting the first equation of the matrix expression in Eq. 3.16 into Eq. 3.17 and by assuming that the vector of generalized applied forces does not contain any explicitly time-dependent terms one gets:

$$[\mathbf{M}_{(i)} - h\mathbf{J}_{\dot{\mathbf{q}}_{(i)}} - h^2\mathbf{J}_{\mathbf{q}_{(i)}}]\Delta\dot{\mathbf{q}}_{(i)} + \mathbf{\Phi}_{\mathbf{q}_{(i)}}^{T}\lambda_{(i)} = h\mathbf{Q}_{(i)} + h^2\mathbf{J}_{\mathbf{q}_{(i)}}\dot{\mathbf{q}}_{(i)} \tag{3.18}$$

where the Jacobians of the vector of generalized applied forces are indicated as $\mathbf{J}_{\mathbf{q}_{(i)}} = \frac{\partial \mathbf{Q}}{\partial \mathbf{q}}\Big|_{t(i)}$ and $\mathbf{J}_{\dot{\mathbf{q}}_{(i)}} = \frac{\partial \mathbf{Q}}{\partial \dot{\mathbf{q}}}\Big|_{t(i)}$. In order to be able to solve Eq. 3.18 for the variations in the n dependent velocities $\Delta\dot{\mathbf{q}}_{(i)}$, m additional equations are needed since there are m extra unknowns contained in the vector of Lagrange multipliers. As suggested by Arnolds et al. in [30], the m constraint equations at the velocity level can be added to Eq. 3.18 coming up with an *index*-2 DAE formulation. By discretizing in time Eq. 3.2 and by imposing that the velocities at the next time step satisfy the constraints at the velocity level the following expression is obtained:

$$\mathbf{\Phi}_{\mathbf{q}}(\mathbf{q}_{(i+1)}, t_{(i+1)})\dot{\mathbf{q}}_{(i+1)} + \mathbf{\Phi}_{t}(\mathbf{q}_{(i+1)}, t_{(i+1)}) = \mathbf{0} \tag{3.19}$$

The expressions for the positions and velocities at the next time step of Eq. 3.15 can be then substituted in Eq. 3.19 to obtain:

$$\mathbf{\Phi}_{\mathbf{q}_{(i+1)}}[\dot{\mathbf{q}}_{(i)} + \Delta\dot{\mathbf{q}}_{(i)}] + \mathbf{\Phi}_{t(i+1)} = \mathbf{0} \tag{3.20}$$

where $\mathbf{\Phi}_{\mathbf{q}_{(i+1)}} = \mathbf{\Phi}_{\mathbf{q}}(\mathbf{q}_{(i)} + \Delta\mathbf{q}_{(i)}, t_{(i+1)})$ and $\mathbf{\Phi}_{t_{(i+1)}} = \mathbf{\Phi}_{t}(\mathbf{q}_{(i)} + \Delta\mathbf{q}_{(i)}, t_{(i+1)})$. The m additional relations of Eq. 3.20 can then be added to Eq. 3.18 in order to obtain the following system of linear equations which can be solved to find the variations in the n dependent velocities and the m Lagrange multipliers at the current time step:

$$\begin{bmatrix} \mathbf{M}_{(i)} - h\mathbf{J}_{\dot{\mathbf{q}}_{(i)}} - h^2\mathbf{J}_{\mathbf{q}_{(i)}} & \mathbf{\Phi}_{\mathbf{q}_{(i)}}^{T} \\ \mathbf{\Phi}_{\mathbf{q}_{(i+1)}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta\dot{\mathbf{q}}_{(i)} \\ \lambda_{(i)} \end{bmatrix} = \begin{bmatrix} h\mathbf{Q}_{(i)} + h^2\mathbf{J}_{\mathbf{q}_{(i)}}\dot{\mathbf{q}}_{(i)} \\ -\mathbf{\Phi}_{\mathbf{q}_{(i+1)}}\dot{\mathbf{q}}_{(i)} - \mathbf{\Phi}_{t(i+1)} \end{bmatrix} \tag{3.21}$$

Eq. 3.21 provides the function evaluation associated with the linearly implicit Euler method for a constrained multibody system whose EOM are expressed in terms of

Table 3.1: Algorithm 1 and correspondent numerical integration routine

```
**algorithm_1** $[\Delta\mathbf{q}_{(i)} \quad \Delta\dot{\mathbf{q}}_{(i)}] = \textit{Function\_Evaluation\_1}(\mathbf{q}_{(i)},\, \dot{\mathbf{q}}_{(i)},\, t_{(i)})$
$\mathbf{M}_{(i)} = \mathbf{M}(\mathbf{q}_{(i)})$
$\mathbf{Q}_{(i)} = \mathbf{Q}(\mathbf{q}_{(i)}, \dot{\mathbf{q}}_{(i)}, t_{(i)})$
$\mathbf{\Phi}_{\mathbf{q}_{(i)}} = \mathbf{\Phi}_{\mathbf{q}}(\mathbf{q}_{(i)}, t_{(i)}, )$
$\mathbf{J}_{\mathbf{q}_{(i)}} = \mathbf{J}_{\mathbf{q}_{(i)}}(\mathbf{q}_{(i)}, \dot{\mathbf{q}}_{(i)})$
$\mathbf{J}_{\dot{\mathbf{q}}_{(i)}} = \mathbf{J}_{\dot{\mathbf{q}}_{(i)}}(\mathbf{q}_{(i)}, \dot{\mathbf{q}}_{(i)})$
$\mathbf{\Phi}_{\mathbf{q}_{(i+1)}} = \mathbf{\Phi}_{\mathbf{q}}(\mathbf{q}_{(i)} + \Delta\mathbf{q}_{(i)}, t_{(i+1)})$
$\mathbf{\Phi}_{t_{(i+1)}} = \mathbf{\Phi}_{t}(\mathbf{q}_{(i)} + \Delta\mathbf{q}_{(i)}, t_{(i+1)})$
$[\Delta\dot{\mathbf{q}}_{(i)} \quad \lambda_{(i)}]^T = \begin{bmatrix} \mathbf{M}_{(i)} - h\mathbf{J}_{\dot{\mathbf{q}}_{(i)}} - h^2\mathbf{J}_{\mathbf{q}_{(i)}} & \mathbf{\Phi}^T_{\mathbf{q}_{(i)}} \\ \mathbf{\Phi}_{\mathbf{q}_{(i+1)}} & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} h\mathbf{Q}_{(i)} + h^2\mathbf{J}_{\mathbf{q}_{(i)}}\dot{\mathbf{q}}_{(i)} \\ -\mathbf{\Phi}_{\mathbf{q}_{(i+1)}}\dot{\mathbf{q}}_{(i)} - \mathbf{\Phi}_{t_{(i+1)}} \end{bmatrix}$

**Numerical Integration 1** $[\mathbf{q},\, \dot{\mathbf{q}}] = \textit{Num\_Int\_1}(\mathbf{q}_{(0)},\, \dot{\mathbf{q}}_{(0)})$
**for** $i$=1 to $i_{end}$
$[\Delta\mathbf{q}_{(i)} \quad \Delta\dot{\mathbf{q}}_{(i)}] = \textit{Function\_Evaluation\_1}(\mathbf{q}_{(i)},\, \dot{\mathbf{q}}_{(i)},\, t_{(i)})$
$\begin{bmatrix} \mathbf{q}_{(i+1)} \\ \dot{\mathbf{q}}_{(i+1)} \end{bmatrix} = \begin{bmatrix} \mathbf{q}_{(i)} \\ \dot{\mathbf{q}}_{(i)} \end{bmatrix} + \begin{bmatrix} \Delta\mathbf{q}_{(i)} \\ \Delta\dot{\mathbf{q}}_{(i)} \end{bmatrix}$
```

dependent coordinates. The unconditional stability of the integration method is guaranteed as long as the matrices $\mathbf{J}_{\mathbf{q}_{(i)}}$ and $\mathbf{J}_{\dot{\mathbf{q}}_{(i)}}$ approximate all the stiff terms in the exact Jacobians $\left.\frac{\partial\mathbf{Q}}{\partial\mathbf{q}}\right|_{t(i)}$ and $\left.\frac{\partial\mathbf{Q}}{\partial\dot{\mathbf{q}}}\right|_{t(i)}$ [31].

The pseudo-code implemented to perform the function evaluation of Eq. 3.21 is reported in Tab. 3.1 together with the pseudo-code associated to the numerical integration of the EOM. It can be noticed that the number of operations required to perform the function evaluation at each time step is fixed thus meeting the basic requirement for RT simulations.

3.3 Automated independent coordinates switching in the linearly implicit Euler method

The linearly implicit Euler method presented in the previous section is able to handle the integration of the stiff DAE system of Eq. 3.1 by performing a fixed number of operations at each time step. With the aim of improving the numerical efficiency of this method we investigate the use of the projection method based on the matrix $\mathbf{R}$ defining the basis of the null-space of the constraint Jacobian, in order to reduce the dimensions of the linear system to be solved at each time step. In particular we propose the use of the projection method at each iteration of the linearly implicit Euler method to automatically transform the expressions for the function evaluation of Eq. 3.21 from a dependent coordinates formulation to a state space form in terms of a minimal set of independent coordinates, allowing a reduction in the dimensions of the linear system from n+m to f=n-m. This reduction in the number of unknowns can be particularly important if a highly constrained mechanical system is taken into account. The projection method based on the matrix **R** relies on the relationships between independent and dependent velocities and accelerations [7]. In order to obtain these relationships the vectors **b** and **c** must be defined from Eqs. 3.2 and 3.3 as:

$$\begin{aligned} \mathbf{\Phi_q}\dot{\mathbf{q}} &= -\mathbf{\Phi}_t \equiv \mathbf{b} \\ \mathbf{\Phi_q}\ddot{\mathbf{q}} &= -\dot{\mathbf{\Phi}}_t - \dot{\mathbf{\Phi}}_\mathbf{q}\dot{\mathbf{q}} \equiv \mathbf{c} \end{aligned} \tag{3.22}$$

In addition to Eq. 3.22 it must be also considered that the independent velocities $\dot{\mathbf{z}}$ can be obtained as the projection of the dependent velocities on the rows of a matrix **B** of dimensions $f \times n$. The set of independent velocities is determined by the choice of matrix **B**. A particular set of independent coordinates is not adequate to univocally determine the configuration of a mechanism during its entire range of motion due to possible singular configurations. For this reason matrix **B** must be changed whenever the configuration of the mechanism cannot be further described by means of the current set of independent coordinates. However for automotive application, as addressed in this paper, matrix **B** can be considered as constant since no singular configurations are reached during normal working conditions. Different numerical methods can be employed in order to obtain matrix **B** starting from the knowledge of the constraint Jacobian matrix [7, 54]. In particular matrix **B** must have full rank f and its rows must be also linearly independent of the m rows of the constraint Jacobian matrix. The relationships between independent and dependent velocities and accelerations are defined as [7]:

$$\dot{\mathbf{q}} = \begin{bmatrix} \mathbf{\Phi_q} \\ \mathbf{B} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{b} \\ \dot{\mathbf{z}} \end{bmatrix} = \begin{bmatrix} \mathbf{S} & \mathbf{R} \end{bmatrix} \begin{bmatrix} \mathbf{b} \\ \dot{\mathbf{z}} \end{bmatrix} = \mathbf{Sb} + \mathbf{R}\dot{\mathbf{z}} \tag{3.23}$$

$$\ddot{\mathbf{q}} = \begin{bmatrix} \mathbf{\Phi_q} \\ \mathbf{B} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{c} \\ \ddot{\mathbf{z}} \end{bmatrix} = \begin{bmatrix} \mathbf{S} & \mathbf{R} \end{bmatrix} \begin{bmatrix} \mathbf{c} \\ \ddot{\mathbf{z}} \end{bmatrix} = \mathbf{Sc} + \mathbf{R}\ddot{\mathbf{z}} \tag{3.24}$$

Eqs. 3.23 and 3.24 can be rewritten in a discretized form as:

$$\begin{aligned} \Delta\mathbf{q}_{(i)} &= h\mathbf{S}_{(i)}\mathbf{b}_{(i)} + \mathbf{R}_{(i)}\Delta\mathbf{z}_{(i)} \\ \Delta\dot{\mathbf{q}}_{(i)} &= h\mathbf{S}_{(i)}\mathbf{c}_{(i)} + \mathbf{R}_{(i)}\Delta\dot{\mathbf{z}}_{(i)} \end{aligned} \tag{3.25}$$

The columns of the $n \times f$ matrix $\mathbf{R}$ in Eqs. 3.23, 3.24 and 3.25 constitute a basis of the nullspace of the constraint Jacobian matrix. This property can be exploited in order to eliminate the term containing the Lagrange multipliers in Eq. 3.18. Indeed by substituting the second one of Eq. 3.25 into Eq. 3.18 and pre-multiplying by the transpose of the projection matrix $\mathbf{R}$ one obtains:

$$\begin{aligned}&\mathbf{R}_{(i)}^T[\mathbf{M}_{(i)} - h\mathbf{J}_{\dot{\mathbf{q}}_{(i)}} - h^2\mathbf{J}_{\mathbf{q}_{(i)}}]\mathbf{R}_{(i)}\Delta\dot{\mathbf{z}}_{(i)} = \\ &\mathbf{R}_{(i)}^T[h\mathbf{Q}_{(i)} + h^2\mathbf{J}_{\mathbf{q}_{(i)}}\dot{\mathbf{q}}_{(i)}] - \mathbf{R}_{(i)}^T[\mathbf{M}_{(i)} - h\mathbf{J}_{\dot{\mathbf{q}}_{(i)}} - h^2\mathbf{J}_{\mathbf{q}_{(i)}}]h\mathbf{S}_{(i)}\mathbf{c}_{(i)}\end{aligned} \tag{3.26}$$

The linear system of Eq. 3.26 has only f variations in the independent velocities as unknowns and represents the function evaluation needed by the linearly implicit Euler method at each time step. Two possible algorithms can be employed in order to carry on the integration process by using the state space formulation of Eq. 3.26 as suggested by de Jalón et al. in [7]. A first possibility is to employ the variations in the independent coordinates and velocities to find the independent coordinates and velocities at the next time step by means of an Euler integration step. The dependent velocities and positions at the next time step must then be found by solving the velocity and the position problems in order to be able to proceed with the integration process. Indeed, the dependent velocities and positions are required at each time step for the computation of the terms in Eq. 3.26. An alternative approach can be used [7] which allows avoiding the solution of the non-linear position problem at each time step. This approach relies on the integration of an enlarged system of differential equations which practically translates in giving in input the current variations in the dependent coordinates $\Delta\mathbf{q}_{(i)}$ and in the independent velocities $\Delta\dot{\mathbf{z}}_{(i)}$ to the Euler integration step obtaining in output the dependent coordinates $\mathbf{q}_{(i)}$ and the independent velocities $\dot{\mathbf{z}}_{(i)}$ at the next time step.

The pseudo-code implementing the function evaluation proposed in this section is reported in Tab. 3.2. The routine receives in input the constant matrix $\mathbf{B}$, the dependent coordinates and the independent velocities at the current time step and returns in output the current variations in the dependent positions and in the independent velocities. The dimensions of the linear system to be solved in order to find the variations in the independent velocities at the current time step is reduced to f=n-m by transforming the EOM from a dependent coordinates formulation to a state space form in terms of a minimal set of coordinates. However, in order to find the projection matrix $\mathbf{R}$, a matrix of dimensions $n \times n$ must be inverted at each time step in algorithm 2 reducing the advantages of the state space formulation. Notice that the constant matrix $\mathbf{B}$ representing the mapping from dependent to independent velocities is computed just once before the

TABLE 3.2: Algorithm 2 and correspondent numerical integration routine

```
$\textbf{algorithm\_2}\ \begin{bmatrix}\Delta\mathbf{q}_{(i)} & \Delta\dot{\mathbf{z}}_{(i)}\end{bmatrix} = \textit{Function\_Evaluation\_2}(\mathbf{q}_{(i)},\ \dot{\mathbf{z}}_{(i)},\ t_{(i)},\ \mathbf{B})$
$\mathbf{\Phi}_{\mathbf{q}_{(i)}} = \mathbf{\Phi}_{\mathbf{q}}(\mathbf{q}_{(i)}, t_{(i)},)$
$\mathbf{T}_{(i)} = \begin{bmatrix}\mathbf{S}_{(i)} & \mathbf{R}_{(i)}\end{bmatrix} = \begin{bmatrix}\mathbf{\Phi}_{\mathbf{q}_{(i)}} \\ \mathbf{B}\end{bmatrix}^{\mathbf{-1}}$
$\mathbf{b}_{(i)} = -\mathbf{\Phi}_t(\mathbf{q}_{(i)}, \mathbf{t}_{(i)})$
$\mathbf{S}_{(i)}\mathbf{b}_{(i)} = \mathbf{T}_{(i)}\begin{bmatrix}\mathbf{b}_{(i)} & \mathbf{0}\end{bmatrix}^T$
$\Delta\mathbf{z}_{(i)} = h\dot{\mathbf{z}}_{(i)}$
$\Delta\mathbf{q}_{(i)} = h\mathbf{S}_{(i)}\mathbf{b}_{(i)} + \mathbf{R}_{(i)}\Delta\mathbf{z}_{(i)}$
$\dot{\mathbf{q}}_{(i)} = \Delta\mathbf{q}_{(i)}/h$
$\mathbf{M}_{(i)} = \mathbf{M}(\mathbf{q}_{(i)})$
$\mathbf{Q}_{(i)} = \mathbf{Q}(\mathbf{q}_{(i)}, \dot{\mathbf{q}}_{(i)}, t_{(i)})$
$\mathbf{J}_{\mathbf{q}_{(i)}} = \mathbf{J}_{\mathbf{q}_{(i)}}(\mathbf{q}_{(i)}, \dot{\mathbf{q}}_{(i)})$
$\mathbf{J}_{\dot{\mathbf{q}}_{(i)}} = \mathbf{J}_{\dot{\mathbf{q}}_{(i)}}(\mathbf{q}_{(i)}, \dot{\mathbf{q}}_{(i)})$
$\mathbf{c}_{(i)} = \mathbf{c}(\mathbf{q}_{(i)}, \dot{\mathbf{q}}_{(i)}, t_{(i)})$
$\mathbf{S}_{(i)}\mathbf{c}_{(i)} = \mathbf{T}_{(i)}\begin{bmatrix}\mathbf{c}_{(i)} & \mathbf{0}\end{bmatrix}^T$
$\Delta\dot{\mathbf{z}}_{(i)} = [\mathbf{R}_{(i)}^T[\mathbf{M}_{(i)} - h\mathbf{J}_{\mathbf{q}_{(i)}} - h^2\mathbf{J}_{\mathbf{q}_{(i)}}]\mathbf{R}_{(i)}]^{\mathbf{-1}}$
          $\cdot[\mathbf{R}_{(i)}^T[h\mathbf{Q}_{(i)} + h^2\mathbf{J}_{\mathbf{q}_{(i)}}\dot{\mathbf{q}}_{(i)}] - \mathbf{R}_{(i)}^{\mathbf{T}}[\mathbf{M}_{(i)} - h\mathbf{J}_{\mathbf{q}_{(i)}} - h^2\mathbf{J}_{\mathbf{q}_{(i)}}]h\mathbf{S}_{(i)}\mathbf{c}_{(i)}]$

$\textbf{Numerical Integration 2}\ [\mathbf{q},\ \dot{\mathbf{z}}] = \textit{Num\_Int\_2}(\mathbf{q}_{(0)},\ \dot{\mathbf{z}}_{(0)})$
$\textbf{for}\ i$=1 to $i_{end}$
$\begin{bmatrix}\Delta\mathbf{q}_{(i)} & \Delta\dot{\mathbf{z}}_{(i)}\end{bmatrix} = \textit{Function\_Evaluation\_2}(\mathbf{q}_{(i)},\ \dot{\mathbf{z}}_{(i)},\ t_{(i)},\ \mathbf{B})$
$\begin{bmatrix}\mathbf{q}_{(i+1)} \\ \dot{\mathbf{z}}_{(i+1)}\end{bmatrix} = \begin{bmatrix}\mathbf{q}_{(i)} \\ \dot{\mathbf{z}}_{(i)}\end{bmatrix} + \begin{bmatrix}\Delta\mathbf{q}_{(i)} \\ \Delta\dot{\mathbf{z}}_{(i)}\end{bmatrix}$
```

beginning of the simulation using one of the methods proposed in [7, 54] and then given in input to algorithm 2.

3.4 Stabilization of the constraint equations

The numerical solutions obtained by means of the two implementations of the linearly implicit Euler method described in the previous sections are affected by the drift off from the manifold defined by the algebraic constraint equations $\mathbf{\Phi}(\mathbf{q}_{(i)}, t_{(i)}) = \mathbf{0}$. In particular, when the EOM in terms of dependent coordinates are expressed in the *index*-2 form of Eq. 3.21, the constraints at velocity level are used in order to provide the m additional equations required to compute the variations in the dependent velocities and the Lagrange multipliers at the current time step. As previously pointed out, since the expressions of the constraints at velocity level do not contain the constant terms originally present in the constraint equations at position level, a linear growth of $\|\mathbf{\Phi}(\mathbf{q}, t)\|$ takes place as the numerical solution advances in time [53]. Similarly, when the EOM expressed in the state space formulation of Eq. 3.26 are integrated to obtain the dependent positions and the independent velocities at the next time step, a drift off of the solution from the constraint manifold is also observed. This drift is due to the fact that the algorithm avoids the computationally expensive solution of the nonlinear position problem defined by the algebraic constraint equations at each time step. Numerical round-off errors can thus accumulate as the numerical integration advances in time causing a violation of the constraint conditions.

A projection of the numerical solution back to the manifold defined by the constraint equations must be thus applied at the end of both algorithms 1 and 2 in order to maintain an acceptable level of accuracy. Several techniques have been proposed in order to prevent the numerical solution from drift off from the manifold defined by the algebraic constraint equations [30, 46, 53]. A typical approach adopted in off-line simulation is the use of iterative projection methods [55]. Generally the norm of the residual vector in the constraint equations at the next time step is evaluated and, if it is higher than a user defined tolerance, a nonlinear constrained minimization problem is solved iteratively in order to find a new set of dependent configuration parameters $\mathbf{q}^*_{(i+1)}$ which belongs again to the constraint manifold as described in [30, 31, 55]:

$$\begin{aligned} &if \left\|\mathbf{\Phi}(\mathbf{q}_{(i)} + h\dot{\mathbf{q}}_{(i)}, t_{(i+1)})\right\| > TOL \\ &then \min\left\{\left\|\mathbf{q}^*_{(i+1)} - [\mathbf{q}_{(i)} + h\dot{\mathbf{q}}_{(i)}]\right\| : \mathbf{\Phi}(\mathbf{q}^*_{(i+1)}, t_{(i+1)}) = \mathbf{0}\right\} \end{aligned} \tag{3.27}$$

After the iterative solution of the constrained minimization problem of Eq. 3.27 the errors in the velocity constraints are corrected as well by projecting the velocities back to the manifold defined by the constraints at the velocity level:

$$\min\left\{\left\|\dot{\mathbf{q}}^*_{(i+1)} - \dot{\mathbf{q}}_{(i+1)}\right\| : \mathbf{\Phi}_{\mathbf{q}}(\mathbf{q}^*_{(i+1)}, t_{(i+1)})\dot{\mathbf{q}}^*_{(i+1)} + \mathbf{\Phi}_{\mathbf{t}}(\mathbf{q}^*_{(i+1)}, t_{(i+1)}) = \mathbf{0}\right\} \tag{3.28}$$

This linear system can be solved without the need of iterative solution methods in order to find the corrected velocities $\dot{\mathbf{q}}^*_{(i+1)}$. Once again the iterative solution of the minimization problem in Eq. 3.27 must be avoided in RT applications where an *a-priori* known number of operations is required at each time step. As demonstrated by Burgermaister et al. [31] one Newton iteration for the solution of the constrained minimization problem mentioned above is sufficient to avoid the drift off effect providing that the projection is performed at each time step. The noniterative projection method can be applied after the function evaluation in both algorithm 1 and algorithm 2 in order to obtain a corrected set of dependent velocities from the solution of the following linear system [30]:

Table 3.3: Numerical integration routines plus the non-iterative projection step

```
Numerical Integration 1 $[\mathbf{q},\ \dot{\mathbf{q}}]= \mathit{Num\_Int\_1}(\mathbf{q}_{(0)},\ \dot{\mathbf{q}}_{(0)})$
for $i$=1 to $i_{end}$
$\begin{bmatrix}\Delta\mathbf{q}_{(i)} & \Delta\dot{\mathbf{q}}_{(i)}\end{bmatrix} = \mathit{Function\_Evaluation\_1}(\mathbf{q}_{(i)},\ \dot{\mathbf{q}}_{(i)},\ t_{(i)})$
$\begin{bmatrix}\dot{\mathbf{q}}^{*}_{(i)}-\dot{\mathbf{q}}_{(i)}\\ \mu_{(i)}\end{bmatrix} = \begin{bmatrix}\mathbf{M}_{(i)} & \mathbf{\Phi_q}^{T}_{(i)}\\ \mathbf{\Phi_q}_{(i)} & \mathbf{0}\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{0}\\ -\frac{1}{h}\mathbf{\Phi}(\mathbf{q}_{(i)}+h\dot{\mathbf{q}}_{(i)},t_{(i+1)})\end{bmatrix}$
$\Delta\mathbf{q}^{*}_{(i)} = h\dot{\mathbf{q}}^{*}_{(i)}$
$\begin{bmatrix}\mathbf{q}_{(i+1)}\\ \dot{\mathbf{q}}_{(i+1)}\end{bmatrix} = \begin{bmatrix}\mathbf{q}_{(i)}\\ \dot{\mathbf{q}}_{(i)}\end{bmatrix} + \begin{bmatrix}\Delta\mathbf{q}^{*}_{(i)}\\ \Delta\dot{\mathbf{q}}_{(i)}\end{bmatrix}$

Numerical Integration 2 $[\mathbf{q},\ \dot{\mathbf{z}}]= \mathit{Num\_Int\_2}(\mathbf{q}_{(0)},\ \dot{\mathbf{z}}_{(0)})$
for $i$=1 to $i_{end}$
$\begin{bmatrix}\Delta\mathbf{q}_{(i)} & \Delta\dot{\mathbf{z}}_{(i)}\end{bmatrix} = \mathit{Function\_Evaluation\_2}(\mathbf{q}_{(i)},\ \dot{\mathbf{z}}_{(i)},\ t_{(i)},\ \mathbf{B})$
$\begin{bmatrix}\dot{\mathbf{q}}^{*}_{(i)}-\dot{\mathbf{q}}_{(i)}\\ \mu_{(i)}\end{bmatrix} = \begin{bmatrix}\mathbf{M}_{(i)} & \mathbf{\Phi_q}^{T}_{(i)}\\ \mathbf{\Phi_q}_{(i)} & \mathbf{0}\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{0}\\ -\frac{1}{h}\mathbf{\Phi}(\mathbf{q}_{(i)}+h\dot{\mathbf{q}}_{(i)},t_{(i+1)})\end{bmatrix}$
$\Delta\mathbf{q}^{*}_{(i)} = h\dot{\mathbf{q}}^{*}_{(i)}$
$\begin{bmatrix}\mathbf{q}_{(i+1)}\\ \dot{\mathbf{z}}_{(i+1)}\end{bmatrix} = \begin{bmatrix}\mathbf{q}_{(i)}\\ \dot{\mathbf{z}}_{(i)}\end{bmatrix} + \begin{bmatrix}\Delta\mathbf{q}^{*}_{(i)}\\ \Delta\dot{\mathbf{z}}_{(i)}\end{bmatrix}$
```

$$\begin{bmatrix}\mathbf{M}_{(i)} & \mathbf{\Phi_q}^{T}_{(i)}\\ \mathbf{\Phi_q}_{(i)} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\dot{\mathbf{q}}^{*}_{(i)}-\dot{\mathbf{q}}_{(i)}\\ \mu_{(i)}\end{bmatrix} = \begin{bmatrix}\mathbf{0}\\ -\frac{1}{h}\mathbf{\Phi}(\mathbf{q}_{(i)}+h\dot{\mathbf{q}}_{(i)},t_{(i+1)})\end{bmatrix} \tag{3.29}$$

As described in [30, 31] the linear system of Eq. 3.29 can be interpreted as one simplified step of the Newton-Raphson algorithm for the solution of the constrained minimization problem of Eq. 3.28 with initial guess $\mathbf{q}^{*}_{(i)} = \mathbf{q}_i + h\dot{\mathbf{q}}_i$ and $\mu_{(i)} = \mathbf{0}$. The the non-iterative projection step in Eq. 3.29 must be applied after the function evaluations of both algorithms 1 and 2 to obtain the corrected variations in the dependent coordinates $\Delta\mathbf{q}^{*}_{(i)} = h\dot{\mathbf{q}}^{*}_{(i)}$ which can then be used within the integration routines to carry on with the integration process as reported in the pseudo-codes of Tab.3. The use of the projection step of Eq. 3.29 is sufficient to drastically reduce the drift away of the numerical solution from the constraint conditions [30, 31].

3.5 Summary

In this Chapter the theoretical background associated to a dependent coordinates formulation of the stiff differential-algebraic Equations of Motion associated to a complex multibody system has been first resumed.

The main features of the off-line implicit integration schemes have been presented, showing that their iterative nature prevents them from be used in real-time applications where the number of operations to be performed at each time step must be known *a priori.*

The linearly implicit Euler method has been then introduced, which constitutes the current state-of-the-art in RT simulations of complex multibody systems. In particular the theoretical foundations of its classical implementation in terms of dependent coordinates have been described and translated into a numerical algorithm (referred to as algorithm 1), which has been implemented in Matlab.

An alternative implementation of the linearly implicit Euler method has been then proposed which is based on the use of the numerical method proposed by de Jalón et al. in [7] to automatically switch from a dependent coordinates representation of the Equations of Motion to an independent one. Also in this case the theoretical background and the numerical implementation (algorithm 2) have been treated.

Finally the non-iterative projection method employed to avoid the violation of the constraint conditions described in [30, 31] has been discussed together with its numerical implementation.

Chapter 4

Industrial application: modelling and integration performances

In order to test and compare the efficiency and the accuracy of the two implementations of the linearly implicit Euler method presented in Chapter 3, an industrial application case has been analysed. The main goal is to highlight possible advantages of the implementation proposed in this research work, which exploits the projection method based on matrix $\mathbf{R}$ to automatically convert the EOM from a dependent to an independent coordinates representation.

4.1 Industrial Application Case

A rear left multilink suspension of a rear-drive passenger car has been considered as shown in Figure 4.1. The mechanical system is composed by 11 bodies which are the chassis, the rear subframe, the 5 suspension's links, the knuckle, the rim, the differential shaft and the halfshaft. The chassis is constrained in such a way that only its 3 translational DOFs are permitted and it is connected to the rear subframe by means of 4 bushings with linear stiffness and damping properties. The driving torque is delivered from the chassis to the rim through the driveline elements which are the differential shaft and the halfshaft. The differential shaft is connected to the chassis by means of a cylindrical joint and to the halfshaft by means of a universal joint. The halfshaft-rim connection is also modeled as a universal joint. Finally the rim is connected to the knuckle by means of a revolute joint. The shock-absorber is modeled as a spring-damper force element with linear stiffness and damping characteristics. A non-linear tire force element has been also introduced in which the tire-road interaction forces applied to the rim are functions of the indentation between the tire contact-patch and the road

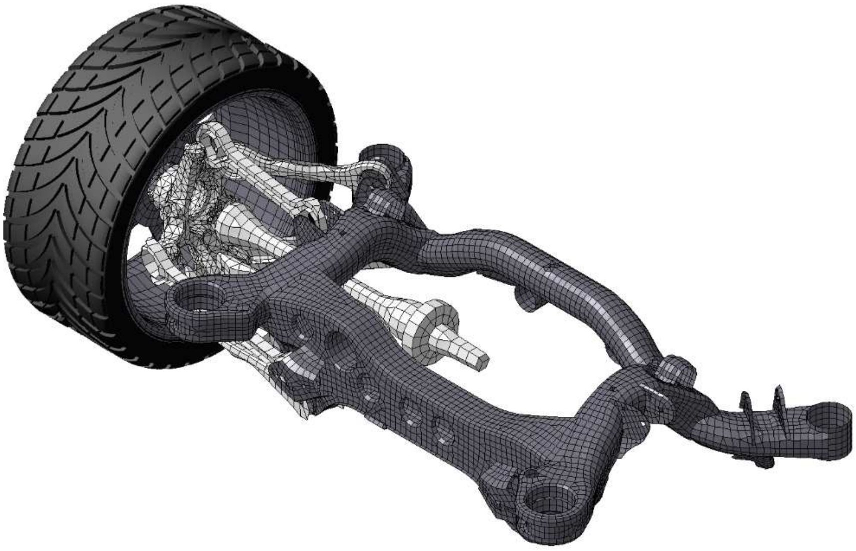


Figure 4.1: Rear multilink suspension of a rear-wheel-drive passenger car

surface. Tangential forces at the tire-road contact patch are computed by the Pacejka's magic formulas while the single contact point transient tire model [56] has been implemented in order to take into account the dynamic phenomena related to the tire carcass compliance.

Starting from the common mechanical system described above 2 models have been obtained, in the first one, which will be referred to as *bushings model* (Fig. 4.2), the connections of the 5 suspension's links with the subframe and the knuckle are defined using 10 bushing force elements with linear stiffness and damping properties. In the second model, which will be indicated as *ideal joints model* (Fig. 4.3), the suspension's links are connected to the subframe by means of 5 universal joints and to the knuckle by means of 5 spherical joints. The *bushings model* and the *ideal joints model* will be used to assess the efficiency of algorithm 1 and algorithm 2 when dealing with stiff DAE problems. In particular their performances will be first tested in the case of a classical suspension model (*bushings model*) and then in the case of a suspension model with a high number of ideal joints connections (*ideal joints model*).

In order to formulate the EOM of the two suspension models under analysis, a dependent Cartesian coordinates approach has been adopted in which the orientation of each body is described by means of 4 dependent Euler parameters. The total number of configuration parameters required to completely describe the position and orientation of each body is thus n=77. The two models have 31 common constraint equations which are the 11 constraint equations imposing the relationships between the dependent Euler parameters; the 3 constraint equations defined to eliminate the rotational DOFs of the chassis; the 17 constraint equations associated to the ideal joints applied among the

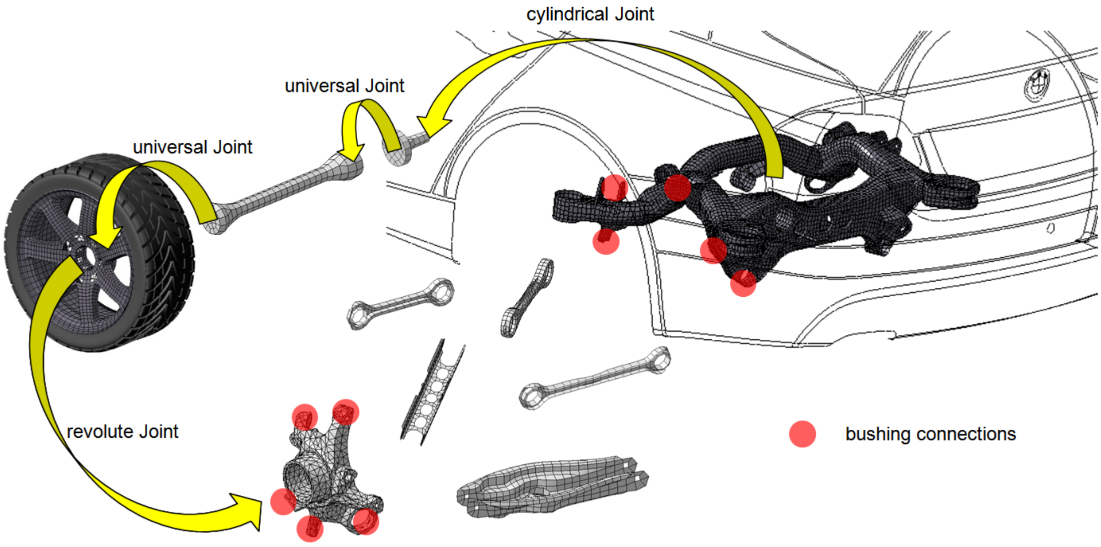


FIGURE 4.2: Bushings model

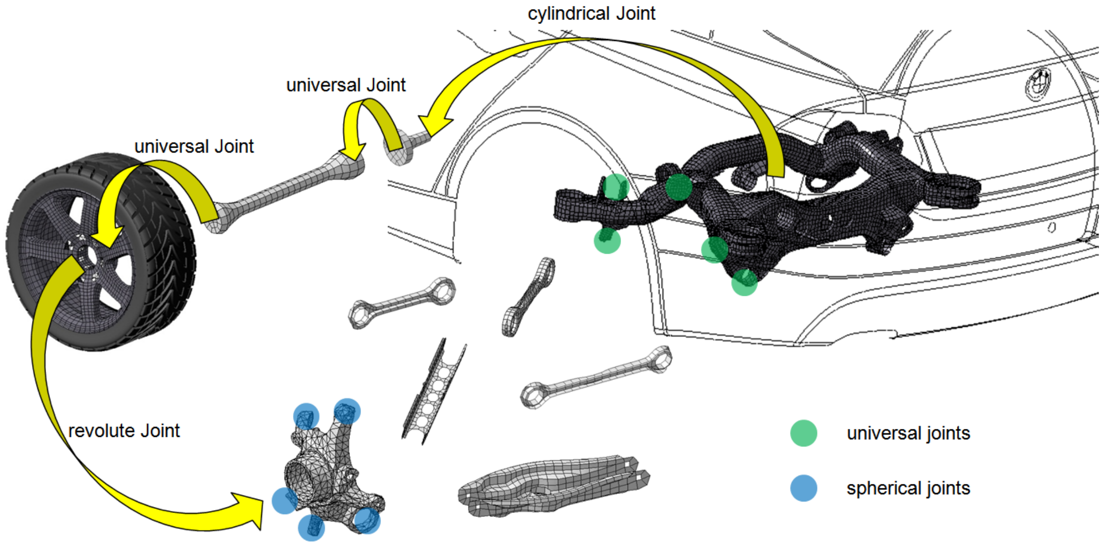


FIGURE 4.3: Ideal joints model

driveline elements. In the ideal joints model 5 additional universal joints are employed to connect the suspenion's links to the rear subframe and 5 additional spherical joints are used to connect the links to the knuckle. 35 additional constraint equations are thus needed for the definition of the *ideal joints model* as resumed in Tab. 4.1.

The implementation of the (*bushings model*) and of the (*ideal joints model*) will be described in the next 2 sections.

TABLE 4.1: Resume of the *bushings model* and the *ideal joints model* properties

	bushings model	***ideal joints model***
# of configuration parameters	n=77	n=77
# of constraint equations	m=31	n=66
# of DOFs	f=46	f=11

4.2 Maple implementation and numerical integration in Matlab

The numerical approach for the modelling of a multibody-system is a quasi-standard choice in general purpose commercial software such as VL Motion, ADAMS, SYMPACK and DADS [4–6, 57]. These software provide a simple and robust way to model complex multibody systems for the most diverse applications. Despite complex models can be easily and quickly defined by means of user-friendly GUIs (Graphic User Interfaces), the accessibility of the underlying EOM is often limited if not possible at all. Moreover the use of a general-purpose multibody software greatly limits the level of control of the user which is not able to choose the coordinates representation to be used in order to define the position and orientation of each body in the system.

An alternative approach is the use of Computer Algebra Software (CAS), such as Maple and Mathematica, to obtain the symbolic representation of matrices and vectors employed in the mathematical definition of the EOM of a multibody system. These latter are generally converted in source codes such as C and Fortran for a fast and efficient numerical integration. The availability of symbolic EOM is particularly important in RT applications where analytical simplifications can be performed in a preliminary step increasing the efficiency of the exported simulation codes. Moreover each term in the EOM can be isolated and exported separately to the numerical simulation environment where it can be further manipulated according to the needs of the user.

In this research work the symbolic approach has been adopted for the modelling of the multibody systems under analysis. In particular the EOM of the two suspensions models have been obtained symbolically by means of the set of methods and procedures collected in the Maple library 'MBSymba' proposed by Lot et al. in [58]. Matalb has been chosen as the numerical simulation environment since it offers a wide range of built-in routines which can be directly used in order to manipulate the terms in the EOM (i.e. matrix multiplication, matrix inversion, etc.). As it will be described in the following paragraphs Matlab has been also used for modelling purposes, in particular to set up the tire force element, and for the coding of the integration routines assessed in

Chapter 3. A brief overview of the modelling process will be now given before expanding the specific issues in the next subsections.

The modelling phase in MBSymba starts with the definition of the spatial configuration of the mechanical system, i.e. all the bodies are defined and positioned with respect to the global reference frame. The constraint and loading conditions must then be described. The MBSymba library offers a variety of commands to define force elements and ideal constraint connections applied among bodies. When the model definition is completed and all bodies, force elements and constraints are properly defined, the 'lagrange_equations' command can be used to obtain the EOM of the multibody system. What this command actually does is to compute the Lagrangian of the system:

$$L = T\left(\mathbf{q}, \mathbf{q}\dot{,} t\right) - V\left(\mathbf{q}\right) + \sum_{k=1}^{m} \lambda_k \phi_k \left(\mathbf{q}, \mathbf{q}\dot{,} t\right) \tag{4.1}$$

where T is the total kinematic energy, V is the potential energy of conservative forces, λ_k are the Lagarnge's multipliers associated to the constraint reactions, and then symbolically compute the associated Lagrange's equations:

$$\frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = Q_i \tag{4.2}$$

Once the EOM have been derived with the 'lagrange_equations' command, all the terms needed to perform the function evaluations in algorithms 1 and 2 have been isolated and then exported in the Matlab environment. In particular the following terms have been extracted:

- the mass matrix $\mathbf{M}(\mathbf{q})$;
- the vector of generalized applied forces $\mathbf{Q}(\mathbf{q}, \dot{\mathbf{q}}, t)$;
- the vector containing the constraint equations in residual form $\mathbf{\Phi}(\mathbf{q})$;
- the jacobian of the constraint equations $\mathbf{\Phi}_{\mathbf{q}}(\mathbf{q})$;
- the vector $\mathbf{c}(\mathbf{q}, \dot{\mathbf{q}})$ coming from the double differentiation of the constraint equations;
- the jacobians of the vector of generalized applied forces $\mathbf{J}_{\mathbf{q}_{(i)}}(\mathbf{q}, \dot{\mathbf{q}}, t)$ and $\mathbf{J}_{\dot{\mathbf{q}}_{(i)}}(\mathbf{q}, \dot{\mathbf{q}}, t)$;

Note that, since in the systems under analysis all the constraint equations are time independent (scleronomous constraints), the term containing the constraint conditions $\mathbf{\Phi}$, its jacobian $\mathbf{\Phi}_{\mathbf{q}}$ and the vector $\mathbf{c}$ do not depend on time. Furthermore the term $\mathbf{\Phi}_{\mathbf{t}}$ vanishes.

The jacobians required by the linearly implicit Euler methods are computed symbolically by taking into account the stiff terms in the vector of generalized applied forces. In the ideal joints suspension models these stiff terms consist only in the reaction forces of the 4 bushings connecting the subframe to the chassis. For the bushings model the number of stiff terms in the vector of generalized forces greatly increases since there are 10 more bushings connections with respect to the ideal joints model. The structures of the jacobians of the vector of generalized forces $\mathbf{J}_{\mathbf{q}_{(i)}}$ and $\mathbf{J}_{\dot{\mathbf{q}}_{(i)}}$ as well as of the constraint jacobians $\mathbf{\Phi}(\mathbf{q})$ for both the bushings and ideal joints models will be described in more details when addressing the actual implementation of the code respectively in subsections 4.2.3 and 4.2.2.

4.2.1 Bodies definition

The first step in developing the suspension models in Maple is the definition of a local *body reference frame* rigidly attached at the CG of each body. The position of the origin of the *body reference frame* associated to the j^{th} body with respect to the global reference frame is defined by the translation vector:

$$\mathbf{O}^j = \begin{bmatrix} x^j & y^j & z^j \end{bmatrix}^T \tag{4.3}$$

The rotation matrix describing the orientation of the j^{th} body in terms of its Euler parameters is defined as:

$$\mathbf{A}^j = \begin{bmatrix} 1-2\,(\theta_2)^2-2\,(\theta_3)^2 & 2(\theta_1\theta_2-\theta_0\theta_3) & 2(\theta_1\theta_3+\theta_0\theta_2) \\ 2(\theta_1\theta_2+\theta_0\theta_3) & 1-2\,(\theta_1)^2-2\,(\theta_3)^2 & 2(\theta_2\theta_3-\theta_0\theta_1) \\ 2(\theta_1\theta_3-\theta_0\theta_2) & 2(\theta_2\theta_3+\theta_0\theta_1) & 1-2\,(\theta_1)^2-2\,(\theta_2)^2 \end{bmatrix} \tag{4.4}$$

The *body reference frame* associated to the j^{th} body is then defined by the following 4×4 matrix:

$$\mathbf{T_{ref}}^j = \begin{bmatrix} & \mathbf{A}^j & & \mathbf{O}^J \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.5}$$

Once the j^{th} *body reference frame* has been defined, it can be given in input, together with the mass and inertia moments, to the command make_BODY in order to complete the definition of the j^{th} body.

Having defined the bodies in the mechanical systems, the correspondent configuration parameters must be ordered and collected in a single vector $\mathbf{q}$. To this aim, the position coordinates and the Euler parameters of the j^{th} body in the system can be first grouped in a body coordinates vector:

$$\mathbf{q}^j = \begin{bmatrix} x^j & y^j & z^j & \theta_0^j & \theta_1^j & \theta_2^j & \theta_3^j \end{bmatrix}^T \tag{4.6}$$

Then the 11 coordinate vectors correspondent to the 11 bodies in the suspension model can be stored together in $\mathbf{q}$ which thus contains all the 77 dependent coordinates describing the configuration of the mechanical system:

$$\mathbf{q} = \begin{bmatrix} (\mathbf{q}^1)^T & \cdots & (\mathbf{q}^j)^T & \cdots & (\mathbf{q}^n)^T \end{bmatrix}^T \tag{4.7}$$

In particular the ordering of the bodies coordinate vectors in the total vector $\mathbf{q}$ is the following:

- j=1 → upper arm;
- j=2 → leading arm;
- j=3 → control arm;
- j=4 → trailing arm;
- j=5 → lower arm;
- j=6 → knuckle;
- j=7 → rim;
- j=8 → differential outshaft;
- j=9 → halfshaft;
- j=10 → chassis;
- j=11 → subframe;

The mass matrix $\mathbf{M}$ of the complete system presents the pattern showed in Fig. 4.4. In particular the squared 7×7 portion of the mass matrix associated to one specific body presents 3 diagonal translational inertia terms each one containing the mass of the body and a 4×4 lower-right matrix containing the inertia terms associated to the Euler parameters. These terms are functions of the moments of inertia of the body along the axis of the *body reference frame* attached to its CG.

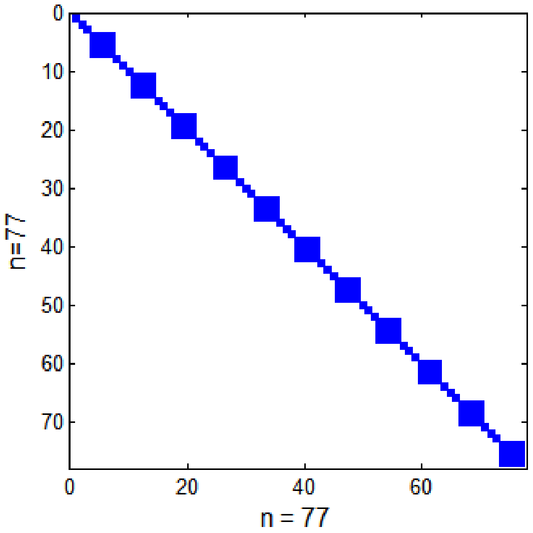


FIGURE 4.4: Mass matrix

4.2.2 Constraint equations and constraint jacobian

As discussed earlier, the use of 4 dependent Euler parameters requires taking into account an algebraic constraint equation for each body in the system. Indeed the four dependent Euler parameters in the orientation matrix of Eq. 4.4 satisfy the following relation:

$$\sum_{k=0}^{3} (\theta_k)^2 = 1 \tag{4.8}$$

Besides the relationships between Euler parameters, the algebraic equations describing the ideal joints in the system must be also taken into account and stored within the vector $\mathbf{\Phi}$. A brief description of ideal joints constraint equations required for the modelling of the suspensions under analysis is given in the followings.

In order to restraint the rotational degrees of freedom of the chassis, 4 constraint equations are required to prevent its Euler parameters from vary with respect to their initial values which are defined as: $\theta_{\mathit{0}}^{chassis} = 1$; $\theta_{\mathit{1}}^{chassis} = 0$; $\theta_{\mathit{2}}^{chassis} = 0$; $\theta_{\mathit{3}}^{chassis} = 0$. These 4 relationships have been used to eliminate the rotations of the chassis. It is worth pointing out that since these 4 constraint equations already fix the values of the chassis Euler parameters, the constraint condition in 4.8 is not required for this particular body.

The built-in commands 'spherical_joint', 'cylindrical_joint' and 'revolute_joint' available in the MSymba library have been used to model the ideal joint connections shown in Figs. 4.2 and 4.2. These commands provide in output respectively 3, 4 and 5 algebraic constraint equations. Since a universal joint command is not available in the MSymba library, these joint connections have been modelled by first applying a revolute joint at the origin of the auxiliary reference systems defining the position and orientation of the joint in the two bodies (3 constraint equations) and then imposing the normal condition between their axes (1 constraint equation).

Once they have all been defined, the algebraic constraint equations can be stored in the vector $\mathbf{\Phi}$ which contains a total of m=66 and m=31 constraint equations respectively in the case of the *ideal joints model* and the *bushings model.*

The order of the constraint equations in the vector $\mathbf{\Phi}$ related to the *ideal joints model* is the following:

- $\mathit{\Phi}_{1} \rightarrow \mathit{\Phi}_{35}$: links connections;
- $\mathit{\Phi}_{36} \rightarrow \mathit{\Phi}_{52}$: driveline connections;
- $\mathit{\Phi}_{53} \rightarrow \mathit{\Phi}_{62}$: Euler parameters relationships;
- $\mathit{\Phi}_{63} \rightarrow \mathit{\Phi}_{66}$: chassis constraints;

For the *bushings model* the first 35 links connections are missing:

- $\mathit{\Phi}_{1} \rightarrow \mathit{\Phi}_{17}$: driveline connections;
- $\mathit{\Phi}_{18} \rightarrow \mathit{\Phi}_{27}$: Euler parameters relationships;
- $\mathit{\Phi}_{28} \rightarrow \mathit{\Phi}_{31}$: chassis constraints;

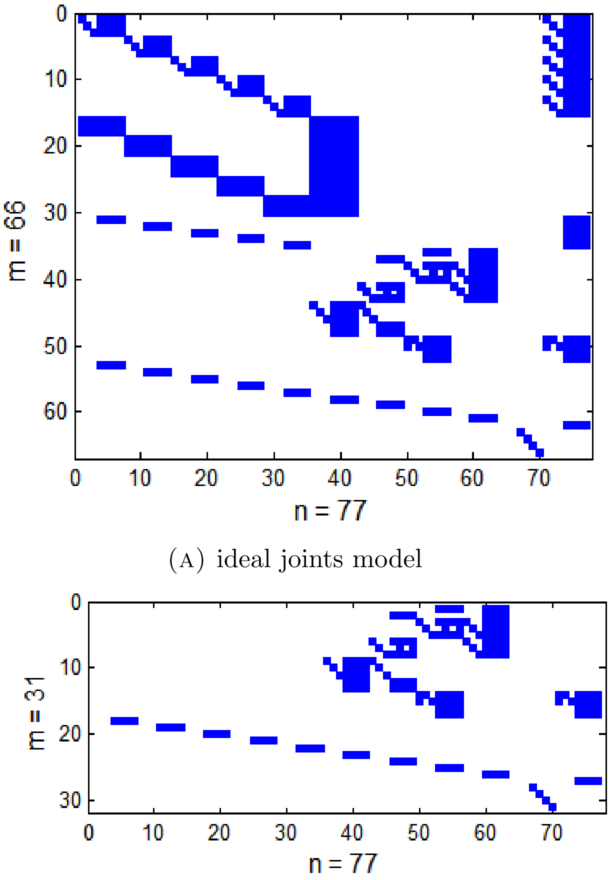
At this point it is possible to compute the constraint jacobian which is defined as:

$$\Phi_{\mathbf{q}} = \begin{bmatrix} \mathit{\Phi}_{q\,11} & \mathit{\Phi}_{q\,12} & \cdots & \mathit{\Phi}_{q\,1n} \\ \mathit{\Phi}_{q\,21} & \mathit{\Phi}_{q\,22} & \cdots & \mathit{\Phi}_{q\,2n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathit{\Phi}_{q\,m1} & \mathit{\Phi}_{q\,m2} & \cdots & \mathit{\Phi}_{q\,mn} \end{bmatrix} \tag{4.9}$$

where $\mathit{\Phi}_{q\,ij} = \partial \mathit{\Phi}_i / \partial q_j$.

The actual patterns of the constraint jacobians are shown in the next figures. In particular Fig. 4.6a shows the constraint jacobian for the *ideal joints models* while in Fig. 4.6b the constraint jacobian related to the *bushings model* is reported.

It is possible to notice that the 31×77 constraint jacobian matrix associated to the *bushings models* is exactly the same as the lower 31×77 part of the constraint jacobian matrix associated to the *ideal joints model* whose total dimensions are 66×77. This is due to the particular ordering adopted to collect the constraint equations in the vectors $\mathbf{\Phi}$ in the 2 models.



(A) ideal joints model

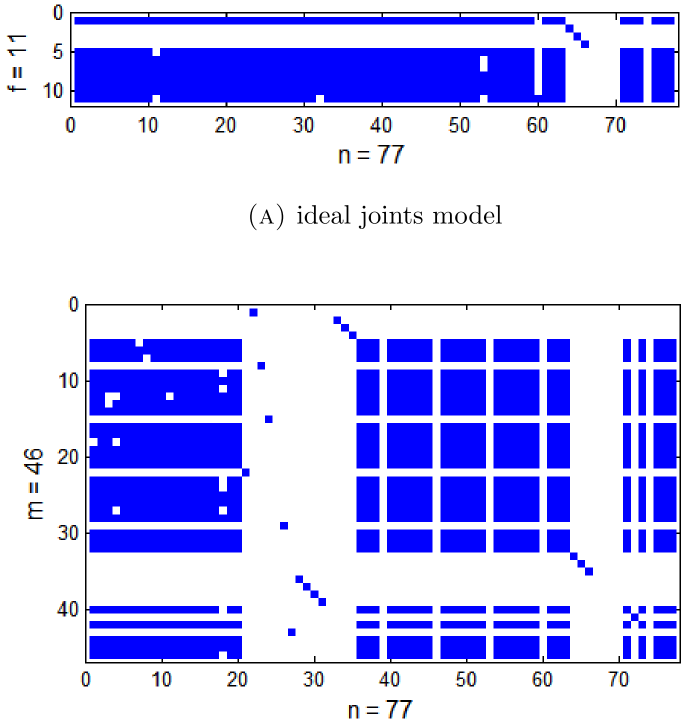
(B) bushings model

FIGURE 4.5: Patterns of the constraint jacobians

As described in Section 2 the matrix $\mathbf{R}$ is obtained by inverting the the $n \times n$ matrix whose $m \times n$ upper partition is the constraint jacobian matrix and whose $f \times n$ lower part is denoted as matrix $\mathbf{B}$:

$$\begin{bmatrix} \mathbf{S}_{(i)} & \mathbf{R}_{(i)} \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi}_{\mathbf{q}(i)} \\ \mathbf{B} \end{bmatrix}^{-1} \tag{4.10}$$

Matrix $\mathbf{B}$ must have full rank f and its rows must be linearly independent from the rows of the constraint jacobian. There are several numerical methods available to compute matrix $\mathbf{B}$ starting from the knowledge of the constraint jacobian matrix as described in [7, 54]. As pointed out in Chapter 3, matrix $\mathbf{B}$ defines the mapping between dependent and independent coordinates and can be computed just once off-line, before the starting of the actual integration process. Indeed, since the suspension doesn't reach any singular configuration during normal operating conditions, the transformation from dependent to independent coordinates can be effectively described using a constant matrix $\mathbf{B}$ computed at the initial configuration of the system. In this work the QR decomposition of



(A) ideal joints model

(B) bushings model

FIGURE 4.6: **B** matrix patterns

the constraint jacobian matrix has been employed to compute the matrix **B**. In particular the constraint jacobian matrix has been first evaluated at the initial configuration of the suspension $\mathbf{\Phi_q}(\mathbf{q_0})$ where $\mathbf{q_0}$ is the vector of configuration parameters correspondent to the initial configuration of the suspension. The Matlab command *qr* has been then employed to decompose its transpose as:

$$[\mathbf{C_1}, \mathbf{C_2}] = qr(\mathbf{\Phi_q}^T(\mathbf{q_0})) \tag{4.11}$$

where the $n \times n$ orthogonal matrix **C1** and the $n \times m$ upper triangular matrix **C2** satisfy the following relationship:

$$\mathbf{\Phi_q}^T(\mathbf{q_0}) = \mathbf{C_1} * \mathbf{C_2} \tag{4.12}$$

Finally, as described in [7, 54], matrix **B** has been obtained as the transpose of the $n \times f$ right partition of matrix **C1**. The patterns of matrix **B** are reported in Fig. 4.6 for both the *ideal joints model* and the *bushings model*. These matrices have been computed just once off-line and then used in the computations related to algorithm 2 as discussed in the Chapter 3.

4.2.3 Standard force elements and bushings

The standard force elements in the suspension models, i.e. the shock-absorber and the traction torque applied from the chassis to the differential shaft, have been defined using the built-in commands available in the MBSymba library. However, MBSymba does not provide a dedicated bushing routine for the modelling a three dimensional elastic force elements connecting two bodies. Moreover, as pointed out previously, the bushing force elements constitute the only source of numerical stiffness in the models under analysis, and their contributions to the jacobians of the vector of generalized applied forces $\mathbf{Q}$ must be computed as well. A bushing routine has been thus implemented using both the MBSymba library in Maple and proper Matlab coding. The ultimate goal is the implementation of a routine which receives in input:

- the coordinates and velocities of 2 generic bodies in a complex mechanical system;
- the local position of the auxiliary reference frames defining the bushing position in each of the 2 bodies;

and returns in output:

- the contributions of the bushing element to the total vector of generalized applied forces $\mathbf{Q}$;
- the contributions of the bushing element to the jacobian matrices $\mathbf{J_q}$ and $\mathbf{J_{\dot{q}}}$;

The routine has been developed by considering 2 isolated bodies i and j which are connected by a general bushing force element, having 3 translational and 3 rotational stiffness and damping characteristics.

Body i and body j are connected by the bushing element at the attachment points P^i and P^j. An auxiliary reference frame is attached at the connection point in each body. The auxiliary reference frames are instrumental to compute the relative kinematic quantities (i.e. relative positions and velocities) which determine the forces and torques of the bushing element. In particular the expressions for the 6 relative translational and angular displacements and the 6 relative translational and angular velocities between the 2 reference frames have been obtained symbolically in Maple using the built-in MBSymba functions. These relative kinematic quantities are only functions of the configuration parameters of the two bodies $\mathbf{q}^i$ and $\mathbf{q}^j$ and of their time derivatives $\dot{\mathbf{q}}^i$ and $\dot{\mathbf{q}}^j$.

Once the relative kinematic quantities are available, the elastic and damping reactions acting on the 2 bodies can be created using the MBSymba commands make_FORCE and make_TORQUE. The EOM of the simplified multibody system only constituted by bodies i and j and by the elastic and damping reactions of the bushing, have been then obtained using the lagrange_equations command in order to find the vector of generalized forces $\mathbf{Q}^{bush_{ij}}$ of dimensions 14×1. This generalized force vector, which contains only the terms associated to the bushing reactions, can be now differentiated in order to find its jacobian matrices. In particular the command jacobianF has been used in Maple to compute the jacobians ${\mathbf{J}_{\mathbf{q}}}^{bush_{ij}}$ and ${\mathbf{J}_{\mathbf{q}}}^{bush_{ij}}$ which are defined as:

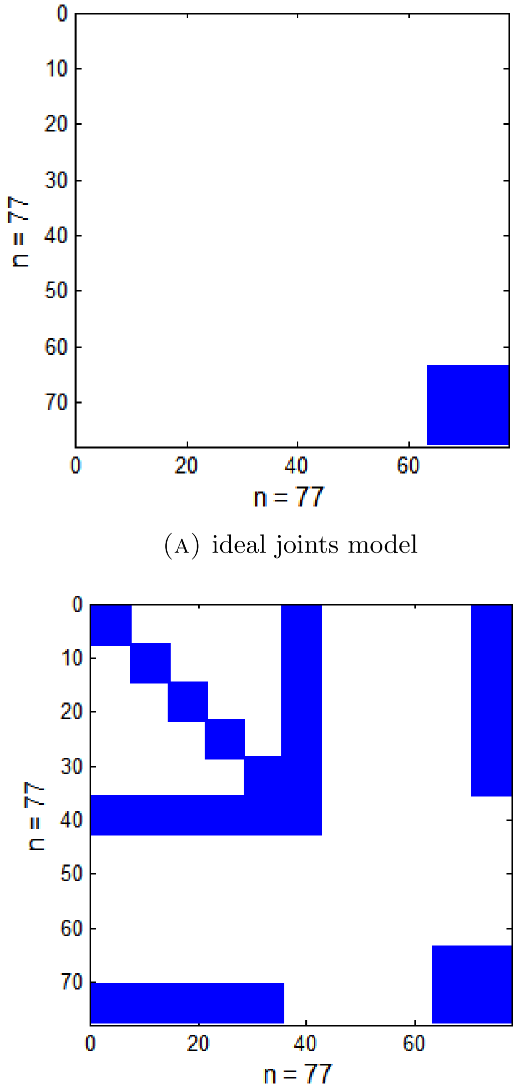
$$\mathbf{J}_{\mathbf{q}}^{bush_{ij}} = \begin{bmatrix} J_{q_{11}} & J_{q_{12}} & \cdots & J_{q_{1n}} \\ J_{q_{21}} & J_{q_{22}} & \cdots & J_{q_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ J_{q_{n1}} & J_{q_{n2}} & \cdots & J_{q_{nn}} \end{bmatrix}, \mathbf{J}_{\dot{\mathbf{q}}}^{bush_{ij}} = \begin{bmatrix} J_{\dot{q}_{11}} & J_{\dot{q}_{12}} & \cdots & J_{\dot{q}_{1n}} \\ J_{\dot{q}_{21}} & J_{\dot{q}_{22}} & \cdots & J_{\dot{q}_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ J_{\dot{q}_{n1}} & J_{\dot{q}_{n2}} & \cdots & J_{\dot{q}_{nn}} \end{bmatrix} \tag{4.13}$$

where $J_{q\,hk} = \partial Q_h^{bush_{ij}} / \partial q_k$, $J_{\dot{q}\,ij} = \partial Q_h^{bush_{ij}} / \partial \dot{q}_k$ and the total number of configuration parameters is n=14 since this custom force routine has been developed in the general case of a bushing connecting two isolated bodies.

When the bushing routine is employed to define a bushing connection between two bodies in a complex mechanical system the issue of properly locate the elements of $\mathbf{Q}^{bush_{ij}}$, $\mathbf{J}_{\mathbf{q}}^{bush_{ij}}$ and $\mathbf{J}_{\dot{\mathbf{q}}}^{bush_{ij}}$ in the total vector of generalized forces and in the total jacobian matrices must be addressed.

This can be done by specifying in input to the bushing routine the structure of vector $\mathbf{q}$, i.e. the location of the coordinates of the 2 bodies affected by the force element. For example if a bushing connecting the chassis to the subframe has to be defined, the position of the first coordinate of the chassis in the vector $\mathbf{q}$ must be specified, together with the location of the first coordinate of the subframe.

The general bushing routine has been used to define the 4 chassis-subframe connections, which are common to both the *bushings model* and the *ideal joints model*, as well as to define the 10 bushes exclusively used in the *bushings model* to attach the suspension links to the subframe and to the knuckle. The patterns of the total jacobian matrices $\mathbf{J}_{\mathbf{q}}$ and $\mathbf{J}_{\dot{\mathbf{q}}}$ associated to the two models are shown in Fig. 4. By comparing these two matrices it can be noticed that in the case of the *ideal joints model* only the elements related to the chassis and subframe coordinates are populated since the bushing connections are defined only between these two bodies.



(A) ideal joints model

(B) bushings model

FIGURE 4.7: Patterns of the total jacobian matrices $\mathbf{J}_{\mathbf{q}}$ and $\mathbf{J}_{\dot{\mathbf{q}}}$

4.3 Tire force element

The tire behaviour is one of the crucial factors governing the dynamics of the whole vehicle. Tires are responsible for the transmission of vertical, longitudinal and lateral forces as well as torque reactions from the road surface to the vehicle. A good understanding and modelling of tires characteristics and of their influence on the whole vehicle dynamics is mandatory in order to perform reliable handling and ride simulations. The issue of properly modelling the tires behaviour has been extensively addressed in literature and well established techniques are available to properly reproduce the dynamic behaviour of a real tire via virtual models at different levels of accuracy. In the next subsections the main issues related to the tire modelling will be addressed, particularly focusing on:

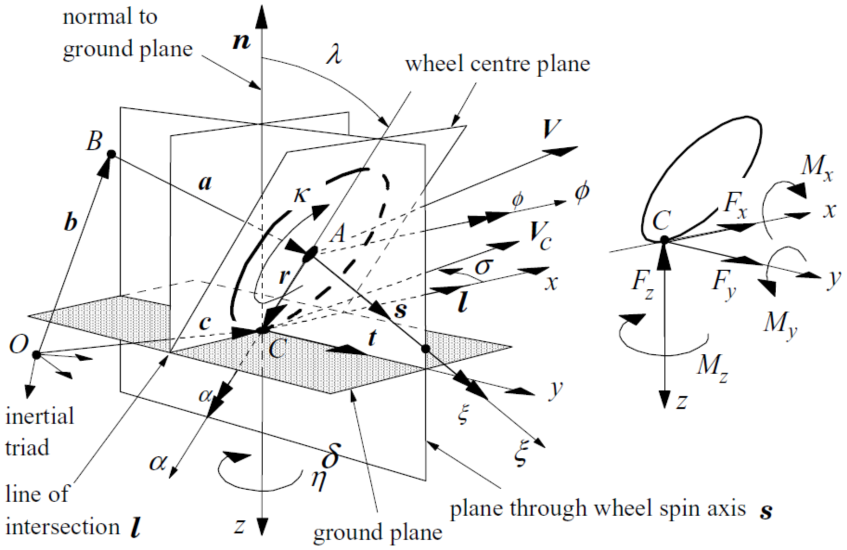


Figure 4.8: Main tire parameters and tire reactions [56]

- proper evaluation of the interaction between the contact patch and the road surface;
- relationship between the inputs and the corresponding reaction forces and moments in steady-state conditions;
- dynamic phenomena due to the tire carcass compliance.

Proper modelling approaches will be presented together with the basic assumptions adopted in the tire model developed in this work. The relevant details of the Matlab implementation of the complete tire model will be described together with its validation w.r.t. a reference tire model.

The sign convention used for tire reactions and other relevant tire kinematics quantities are shown in Fig. 4.8 where several planes are defined such us the wheel-centre-plane and the road plane whose intersection determines the longitudinal direction of the wheel, i.e. the unit vector **l**. In particular the road plane is identified by its normal **n** at the contact point C. Two planes normal to the ground plane are also highlighted, one containing the vector **l** and one containing the spin axis of the wheel **s**. The unit vector **t** is associated to the lateral direction of the wheel which is determined by the intersection of the ground plane and the plane normal to the ground and passing through the spin axis **s**. The unit vectors **l** and **t** are therefore perpendicular. The length of vector **r** connecting the wheel centre A to the contact point C is the loaded tire radius. The position and orientation of the wheel with respect to the global reference frame is univocally determined once the position of the wheel centre A and the vector **s** defining the wheel-spin axis are known. These quantities can be determined once the 3 translational coordinates and the 4 Euler parameters associated to the knuckle (i.e. the wheel-carrier) are known.

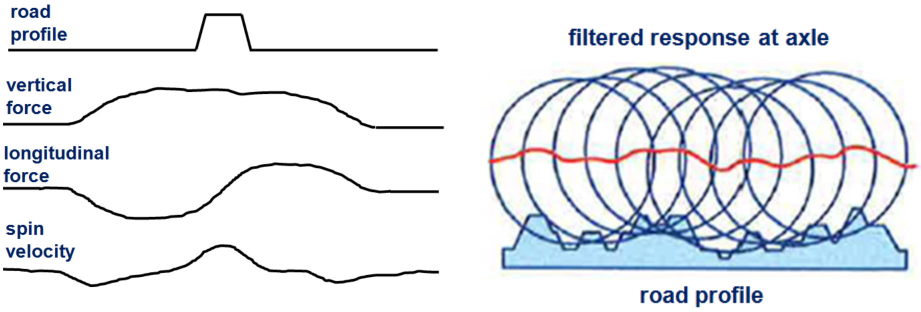
Fig. 4.8 also shows the speed of travel of the wheel center which is indicated as **V**, the longitudinal slip and the side slip angle denoted as κ and α respectively and the camber angle denoted as γ. The 6 tire reactions are also shown, in particular F_x, F_y and F_z are the longitudinal, lateral and normal forces while M_x, M_y, and M_z are the overturning, the rolling-resistance and self-aligning moments respectively.

4.3.1 Simplified Tire Enveloping model

The primary source of excitation for the tire dynamics is the road roughness. For short wavelength unevenness, the tyre enveloping behaviour plays a major role in determining the dynamic response of the tire itself. Fig. 4.9a shows three distinct responses caused by a short wavelength obstacle on the road:

- a variation in the vertical force;
- a variation in the longitudinal force;
- a variation in the spin velocity of the wheel.

It is important to notice that the tyre is in contact with the obstacle before and after the wheel centre is actually positioned above the obstacle and that the shape of the dynamic responses are completely different from the obstacle shape. In order to take into account all this phenomena a proper tyre enveloping model must be adopted [56, 59]. The main idea behind the tire enveloping model is that the tire behaves like a filter which modifies the road unevenness transmitted from the contact patch to the spindle as shown in Fig. 4.9b. A special road filter has thus been proposed [56, 59] in order to take into account the tire enveloping properties. The obtained filtered road surface is referred to as the effective road surface.



(A) Rolling over a short obstacle: tire responses

(B) Tandem cam enveloping model

FIGURE 4.9: Tire enveloping characteristics

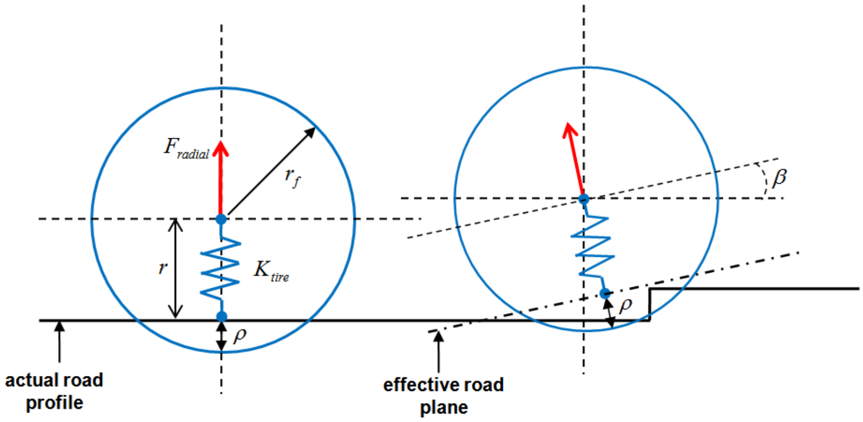


FIGURE 4.10: Effective road plane

In order to better understand the concept of effective road surface it is useful to refer to Fig. 4.10 where it can be appreciated that, if the effective road plane is known, a tyre model with a single point tire-road interface can be used to calculate the tire response (i.e. the magnitude of the contact force at the wheel axle and its direction). In order to obtain the effective road plane a semi-empirical enveloping model has been developed [59] which is based on the basic consideration that the enveloping properties of the tire are mainly determined by the deformation of the tread in the contact zone. In particular the semi-empirical model takes into account the following physical properties of the tire:

- the curvature near the edges of the contact patch does not vary significantly with vertical load;
- the vertical stiffness in the centre of the contact patch is very low if compared to the stiffness of the bended belt near the edges of the contact patch;

The tandem cam tire model reported in Fig.4.11 is able to reproduce the empirical properties listed above. It is composed by two rigid elliptical cams representing the front and rear edges of the tire in the contact zone. The cams lie in the wheel centre plane and they are allowed to move vertically. By knowing the longitudinal position x of the wheel centre, the longitudinal positions of the vertical axes z^r and z^f along which the two cams are allowed to move are determined as well. The angle β, which defines the effective road plane, is found by letting the cams slide vertically on the wheel centre plane in order to adapt to the road profile height as shown in Fig.4.11. Once the cams have moved vertically according to the particular road shape, β can be found by simple geometric considerations, i.e. tacking into account the elliptical profile of the cams, and then used to compute the tire radial deflection ρ defined as:

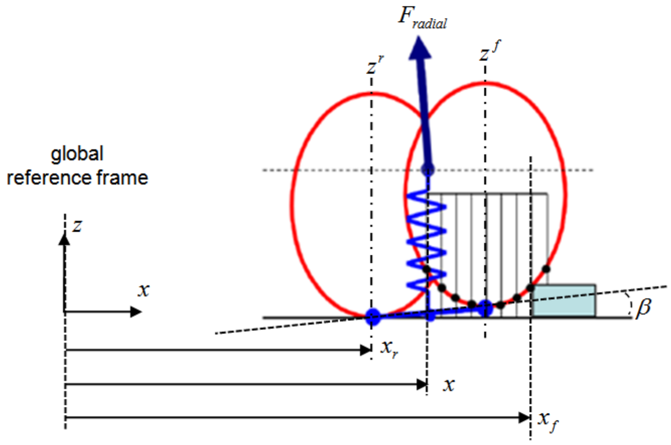


FIGURE 4.11: Tandem cam enveloping model: Contact detection

$$\rho = r_f - r \tag{4.14}$$

where r_f is the unloaded radius of the tire. The main drawback of the tandem cam model is that in order to find the distances x_f and x_r defining the longitudinal positions of the contact points between the cams and the road, the height of the road must be compared with those of the cams profiles. This is actually done for a certain number of points along the longitudinal direction as shown in Fig. 4.11.

In order to avoid the computational burden associated to these operations, a simplified two cams tire model has been used in this research work, in which the number of points where the height of the road profile is compared to that of the cams profiles is reduced to one for each cam. The two cams model thus reduces to a simplified 2-points follower model as shown in Fig. 4.12. Even if the accuracy of the two-points follower model is reduced with respect to the tandem cam model, its computationally cheapness makes it prone to be used in real-time simulations. The 2-points follower model has been implemented in a proper Matlab function. This latter receives in input the road profile height as a function of the longitudinal distance travelled by the wheel, the position of the rim center and the rotation matrix which defines the orientation of the rim-carrier in the global reference frame. The routine returns in output the angle β defining the effective road plane and the radial deflection of the tire ρ. The magnitude of the radial contact reaction force which must be applied to the rim centre is obtained as ρ times the radial stiffness of the tire. Once the magnitude of the radial contact force has been computed, its projections along the longitudinal and vertical directions of the rim-carrier are determined by the effective road plane angle β. It is important to notice that Fig. 4.11 represents a lateral view of the tire in the wheel centre plane which has been defined

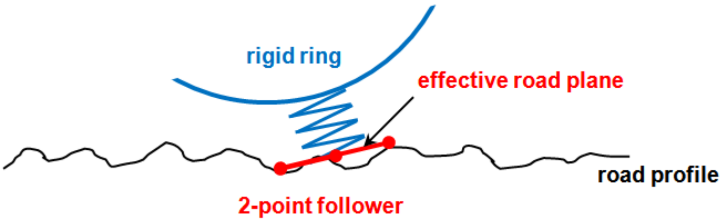


FIGURE 4.12: 2-point follower enveloping model

in Fig. 4.8 and that the camber angle of the tire is taken into account by means of the rotation matrix of the rim-carrier.

4.3.2 Pacejka Steady-State Formula

The steady-state reactions of the tire can be determined by means of the Magic Formula model proposed by Pacejka [56]. The modelling approach adopted by Pacejka is semi-empirical since it is based on both measured data and physical considerations on the tire behaviour. According to this approach the tire can be modelled as a non-linear system with multiple inputs (slip quantities, tire's angles and load force) and outputs (longitudinal and lateral forces and tire moments). The relationships between inputs and outputs are defined by means of the Magic Formula model as it will be explained in this section. The slip quantities in input to the Magic Formula are the longitudinal slip and the lateral slip angle. The longitudinal slip is defined as:

$$\kappa = -\frac{V_x - \omega_{spin} \cdot r}{V_x} \tag{4.15}$$

where V_x is the component of the wheel centre speed along the longitudinal direction defined by the unit vector $\mathbf{l}$ in Fig. 4.8, ω_{spin} is the spin angular velocity of the wheel and r is the effective rolling radius. The lateral slip angle α is defined in a similar way as:

$$tan\left(\alpha\right) = -\frac{V_y}{V_x} \tag{4.16}$$

where V_y is the component of the wheel centre speed along the lateral direction defined by the unit vector $\mathbf{t}$ in Fig. 4.8. These 2 slip quantities can be given in input to the semi-empirical Magic Formula which assumes the following general form:

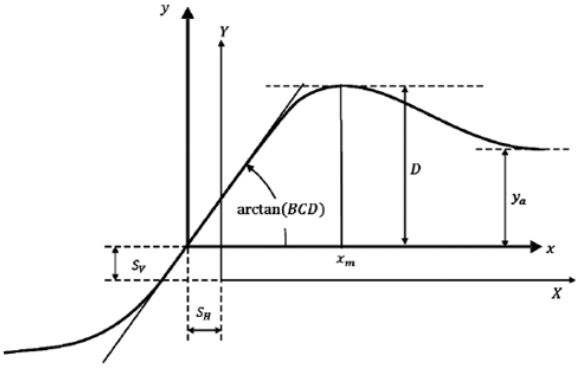


Figure 4.13: Parameters of the magic formula

$$y = D \cdot sin(C \cdot arctan(B \cdot x - E(B \cdot x - arctan(B \cdot x)))) \tag{4.17}$$

In Eq.4.17 the term B is a stiffness factor, C is a shape factor, D is the peak value and E is a curvature factor which must be determined experimentally. The generic independent variable x in Eq. 4.17 is replaced by κ and α to obtain the longitudinal and lateral tire reactions respectively. Due to ply-steer, conical effects and wheel camber, an offset may arise both in x and y in the general formula of Eq. 4.17. These offsets can be taken into account by using the modified coordinates:

$$\begin{aligned} Y(X) &= y(x) + S_v \\ X &= x + S_H \end{aligned} \tag{4.18}$$

The meaning of the coefficients retained in the general Magic Formula of Eq.4 can be appreciated in Fig. 4.13. The steady-state Magic Formula model just described has been implemented in Matlab. The values of the coefficients described above have been extracted from a tire parametric file as will be further described in the followings.

4.3.3 Computation of the spin angular velocity of the wheel

In order to be able to compute the longitudinal slip of the tire a proper sequence of operations has been implemented to find the spin angular velocity of the wheel ω_{spin}. Indeed the spin angular velocity is not directly available among the states of the multibody model when dependent coordinates are used. The relative angular velocity between the rim and the knuckle must be thus computed using the corresponding states of these two bodies. By denoting with $\mathbf{A}_{rim}$ and $\mathbf{A}_{knuckle}$ the rotation matrices of the rim and of the knuckle respectively, the skew symmetric matrices associated with the angular velocity of the rim and the knuckle in the absolute reference frame can be computed as [8]:

$$\begin{aligned}\widetilde{\omega}_{rim} &= \dot{\mathbf{A}}_{rim} \cdot \mathbf{A}_{rim}^{T} \\ \widetilde{\omega}_{knuckle} &= \dot{\mathbf{A}}_{knuckle} \cdot \mathbf{A}_{knuckle}^{T}\end{aligned} \tag{4.19}$$

The relative angular velocity between rim and knuckle in the global reference frame can then be computed as:

$$\widetilde{\omega}_{kr}^{global} = \widetilde{\omega}_{knuckle} - \widetilde{\omega}_{rim} \tag{4.20}$$

If the relative angular velocity is expressed in the local reference frame associated with the knuckle body, the spin angular velocity can be simply obtained by selecting its y-component. In order to express the relative angular velocity between rim and knuckle in the reference frame of the knuckle it must be pre and post multiplied by the rotation matrix of the knuckle [8]:

$$\widetilde{\omega}_{kr}^{knuckle} = \mathbf{A}_{knuckle}^{T} \cdot \widetilde{\omega}_{kr}^{global} \cdot \mathbf{A}_{knuckle} = \begin{bmatrix} 0 & -\omega_3 & \omega_{spin} \\ \omega_3 & 0 & -\omega_1 \\ -\omega_{spin} & \omega_1 & 0 \end{bmatrix} \tag{4.21}$$

Eqs. 4.19, 4.20 and 4.21 have been implemented in Matlab in order to extract the desired value of the spin angular velocity ω_{spin} which can be then used to compute the values of the longitudinal and turn slips by means of Eqs. 4.15 and 4.16.

4.3.4 Single Contact Point Transient Tire Model

The applicability of the Magic Formula model is limited to steady-state conditions. This practically means that for given longitudinal and lateral slips the corresponding longitudinal and lateral forces and tire reaction moments computed by means of the Magic Formula model build up instantaneously. This is in contrast with what actually happens in a real tire where, due to the compliance of the tire carcass, a certain delay occurs between the occurrence of the slip conditions and the building up of the corresponding tire reactions. The concept of *relaxation length* plays a key role in the modelling of such a transient behaviour. The relaxation length can be interpreted as the distance a tire must roll before it can actually develop a reaction force or moment. In order to take into account this phenomenon the single contact point transient tire model proposed by Pacejka [56] and reported in Fig. 4.14 can be employed. In this model the contact patch

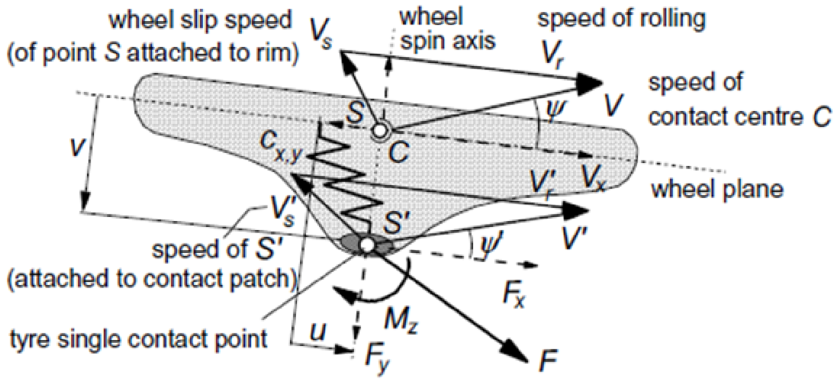


FIGURE 4.14: Single contact point model [56]

is reduced to a single point S' which is connected to the wheel by a longitudinal spring and a lateral spring. This allows the contact point to slip with respect to the ground in the longitudinal and lateral directions reproducing in a simplified way what actually happens in a real tire due to the carcass compliance.

In Fig. 4.14 it is possible to distinguish the two points S and S' which are located in the plane normal to the road and passing through the wheel spin axis. The point S moves on the road surface with the same lateral and longitudinal velocities of the wheel's centre while the velocity of the point S' represents the velocity of the contact patch. The difference in the velocities of S and S' causes a longitudinal and a lateral deflection in the tire carcass which are denoted respectively as u and v. The changes in the longitudinal and lateral tire deflections can be written as:

$$\begin{aligned} \frac{\mathrm{d}u}{\mathrm{d}t} &= -(V_{SX} - V'_{SX}) \\ \frac{\mathrm{d}v}{\mathrm{d}t} &= -(V_{SY} - V'_{SY}) \end{aligned} \tag{4.22}$$

After several conversions Eqs. 4.22 can be rewritten taking into account the longitudinal slip relaxation length σ_κ and the side slip relaxation length σ_α as:

$$\begin{aligned} \frac{\mathrm{d}u}{\mathrm{d}t} + \frac{1}{\sigma_\kappa} \cdot |V_x| \cdot u &= -V_{SX} \\ \frac{\mathrm{d}v}{\mathrm{d}t} + \frac{1}{\sigma_\alpha} \cdot |V_x| \cdot v &= -V_{SY} \end{aligned} \tag{4.23}$$

In Eq. 4.23 the longitudinal relaxation length is defined as $\sigma_\kappa = C_{F\kappa}/C_{Fx}$ with $C_{F\kappa}$ being the longitudinal tire stiffness at the road level and C_{Fx} being the longitudinal slip stiffness. Similarly the lateral relaxation length is defined as $\sigma_\alpha = C_{F\alpha}/C_{FY}$ where $C_{F\alpha}$ is the lateral tire stiffness and C_{FY} is the lateral slip stiffness. The dynamic differential equations of the single point transient tire model reported in Eq. 4.23 can be implemented in order to find the longitudinal and lateral deflections of the carcass u and v. Once these deflections have been computed a further step is needed to determine the tire reactions, that is the computation of the following transient slip quantities:

$$\kappa' = \frac{u}{\sigma_\kappa}, \alpha' = \frac{v}{\sigma_\alpha} \tag{4.24}$$

Once the transient slip quantities of Eqs. 4.24 have been computed they can be given in input to the classical steady-state Magic Formula model in order to obtain the tire reactions.

4.3.5 Matlab implementation of the tire model

Fig. 4.15 shows the scheme of the tire implementation that has been performed within the Matlab environment. The routine for the computation of the tire reactions receives in input the states of the rim and of the knuckle coming from the multibody representation of the suspension system.

The simplified 2-points tire enveloping model is then used to take into account the interaction between the tire and the road profile. In particular a road file containing the information about the road profile is loaded during the simulation and proper interpolation functions are used to compute the height of the road at the points of interest along the longitudinal direction. The simplified tire enveloping model computes the amount of the radial force developed by the tire and the correspondent direction of application which is defined by the effective road plane. The differential equations associated to the single contact point transient tire model are then solved in order to obtain the longitudinal and lateral deflections of the tire carcass which are later used to obtain the transient slip quantities. These latter, with the radial force developed by the tire are finally given in input to the Magic Formula steady-state model which allows the computation of the tire reactions that must be applied to the wheel centre in the multibody model of the suspension.

4.3.6 Validation of the tire model w.r.t. the TNO MF-Tire

The MF-Tire and the MF-Swift tire model by TNO Delft-Tyre represent the current state-of-the-art in automotive industry for the modelling of tires in handling simulations. The MF-Tire model implements the Pacejika Magic Formula to compute the tire's forces and moments, the semi-empirical enveloping model with elliptical cams for the detection of the tire-road contact force and the single contact point transient tire model for the

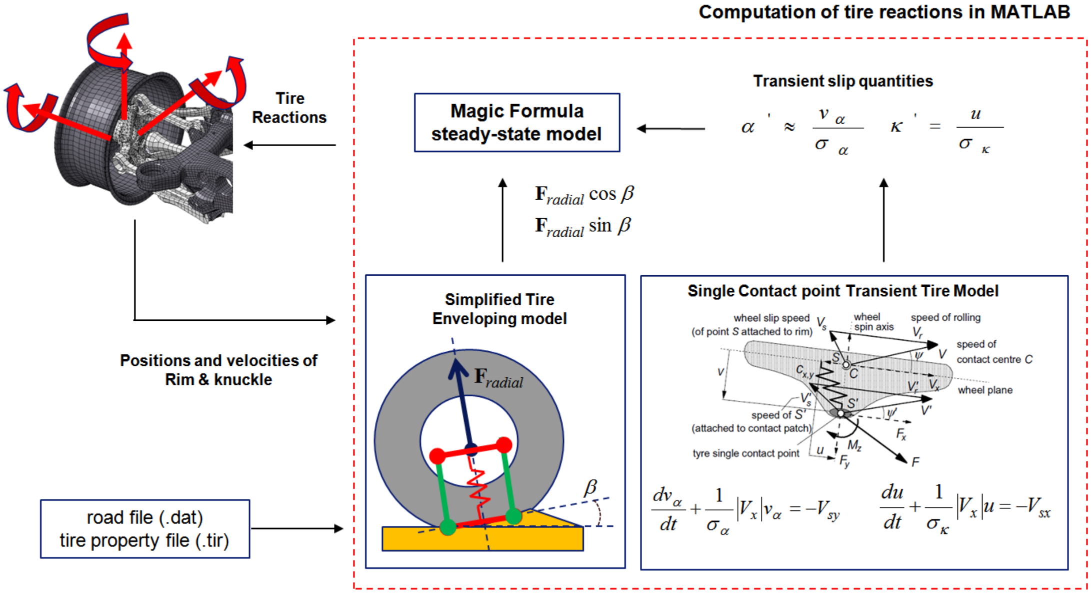

Computation of tire reactions in MATLAB
Tire Reactions
Magic Formula steady-state model
Transient slip quantities
$\alpha' \approx \frac{v_\alpha}{\sigma_\alpha}$
$\kappa' = \frac{u}{\sigma_\kappa}$
$\mathbf{F}_{radial} \cos\beta$
$\mathbf{F}_{radial} \sin\beta$
Positions and velocities of Rim & knuckle
Simplified Tire Enveloping model
$\mathbf{F}_{radial}$
β
Single Contact point Transient Tire Model
wheel slip speed (of point S attached to rim) V_s
wheel spin axis
speed of rolling V_r
speed of contact centre C
V
ψ
S
C
$c_{x,y}$
v
V_x
V'_r
wheel plane
V'_s
S'
V'
ψ'
speed of S' (attached to contact patch)
F_x
M_z
tyre single contact point
u
F_y
F
$\frac{dv_\alpha}{dt} + \frac{1}{\sigma_\alpha}|V_x|v_\alpha = -V_{sy}$
$\frac{du}{dt} + \frac{1}{\sigma_\kappa}|V_x|u = -V_{sx}$
road file (.dat)
tire property file (.tir)

FIGURE 4.15: Scheme of the Matlab implementation of the tire model

modelling of dynamic phenomena due to carcass compliances. Additionally, in the MF-Swift model the tire belt is assumed to behave like a rigid ring thus enabling a correct representation of the tire dynamics in the frequency range where the bending modes of the tire belt can be neglected, i.e. approximately up to 60 − 100 Hz. The visual representation of the MF-Swift model is reported in Fig. 4.16 where the cams enveloping model, the single contact point transient slip model and the rigid model can be clearly distinguished. The aim of this section is the validation of the simplified tire model described in the previous sections against the results furnished by the MF-Tire model. To this purpose the multibody package VL.Motion has been used which provides a MF-Tire force element. Since we are only interested in the tire responses, a trivial suspension model has been set up in VL.Motion, as described in Fig. 4.17, which is composed by only two bodies: the wheel carrier and the rim.

The wheel carrier is allowed to translate along the 3 directions and is free to rotate around the vertical axis (yaw motion) while a revolute joint has been used to constraint the rim to the wheel carrier. The driving torque is directly applied to the rim body from the wheel carrier while the yaw angular displacement of the wheel carrier is controlled using an angular position driver. The TNO tire force element has been defined in VL.Motion using the following settings:

- Dynamics mode: Relaxation behavior, non-linear;
- Contact Method: 2D road;
- Slip forces - Magic Formula evaluation: combined forces/moments;

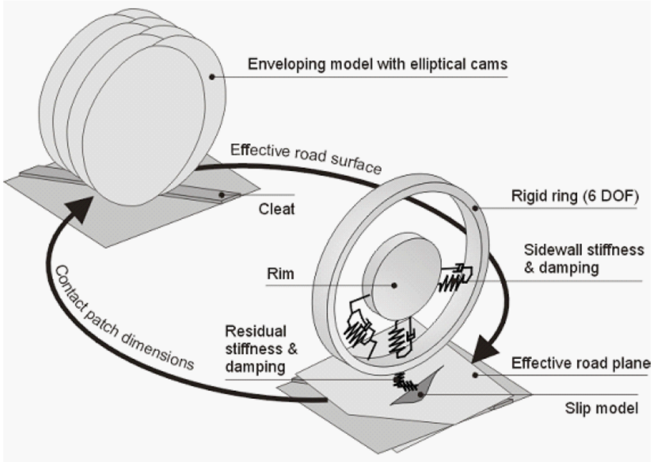


FIGURE 4.16: Scheme of the TNO MF-Swift Tire [59]

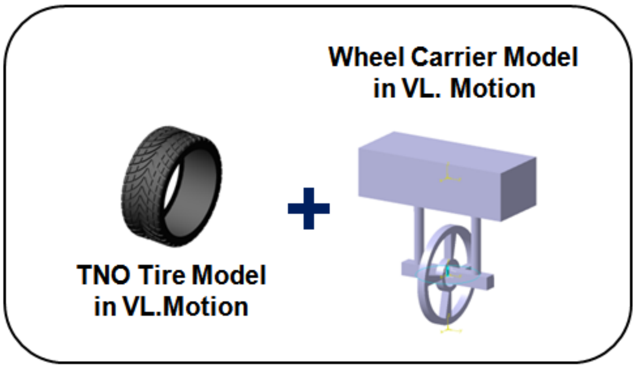

Wheel Carrier Model in VL. Motion
TNO Tire Model in VL.Motion
+

FIGURE 4.17: Tire test manoeuvre: simplified suspension model

When selecting the non-linear relaxation behaviour, the compliance of the tire carcass is taken into account by means of the single contact-point transient slip model which gives accurate results from 0 Hz to approximately 10 Hz. Setting the Contact method to ’2D road’ implies that for the computation of the road height,the travelled distance is used which is the distance the wheel centre has travelled with respect to the origin of the global coordinate system. This approach is the same used in the Matlab tire model implemented in this work. Finally the combined forces/moments options for the slip forces computation in the Magic Formula takes into account the combined effect of longitudinal and lateral slips. All the parameters required for the definition of the TNO MF-Tire are grouped into a tire property file which is structured in different sections:

- Dimension/Inertia;
- Vertical;
- Structural;
- Contact Patch;
- Longitudinal Coefficients;
- Overturning Coefficients;

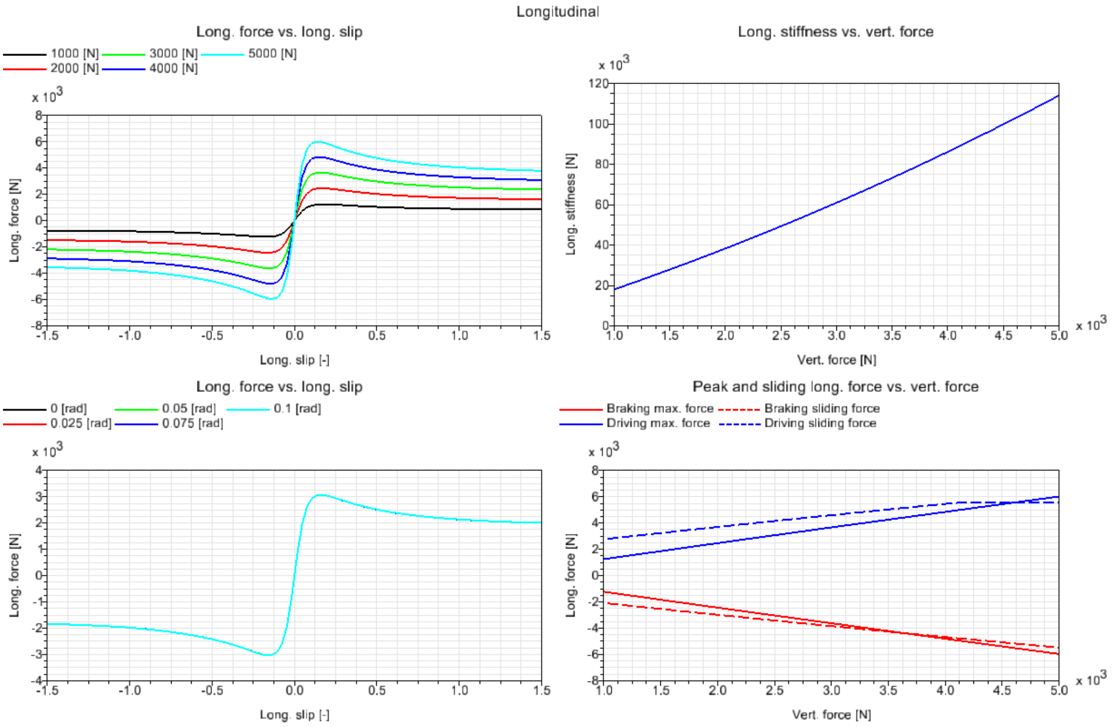


Figure 4.18: Magic Formula: longitudinal properties

- Lateral Coefficients;
- Rolling Coefficients;
- Aligning Coefficients;

Fig. 4.18 shows the longitudinal properties of the TNO tire force elements under analysis. In the top-left graph it can be appreciated the influence of the vertical load on the tire 'longitudinal slip – longitudinal force' curve for this particular tire. The top-right graph shows the relationship between the vertical force and the longitudinal stiffness of the tire while the bottom-left figure shows that the influence of the wheel camber angle on the 'longitudinal slip – longitudinal force' curve is negligible. Finally the bottom-right graph shows the variations of the maximum braking and driving forces as well as the variations of braking and sliding forces correspondent to a change in the wheel vertical load.

The lateral properties of the tire are shown in Fig. 4.19, in particular top-left and top-right graphs show respectively the 'slip angle – lateral force' and the 'slip angle – aligning moment' curves for different values of the vertical wheel load. Additionally the bottom-left and bottom-right graphs show respectively the 'slip angle – lateral force' and the 'slip angle – aligning moment' curves for different values of wheel camber angles. It is important to highlight at this point that the same coefficients and tire properties used in the definition of the TNO tire force element in VL.Motion have been used in the Matlab tire model implemented in this work.

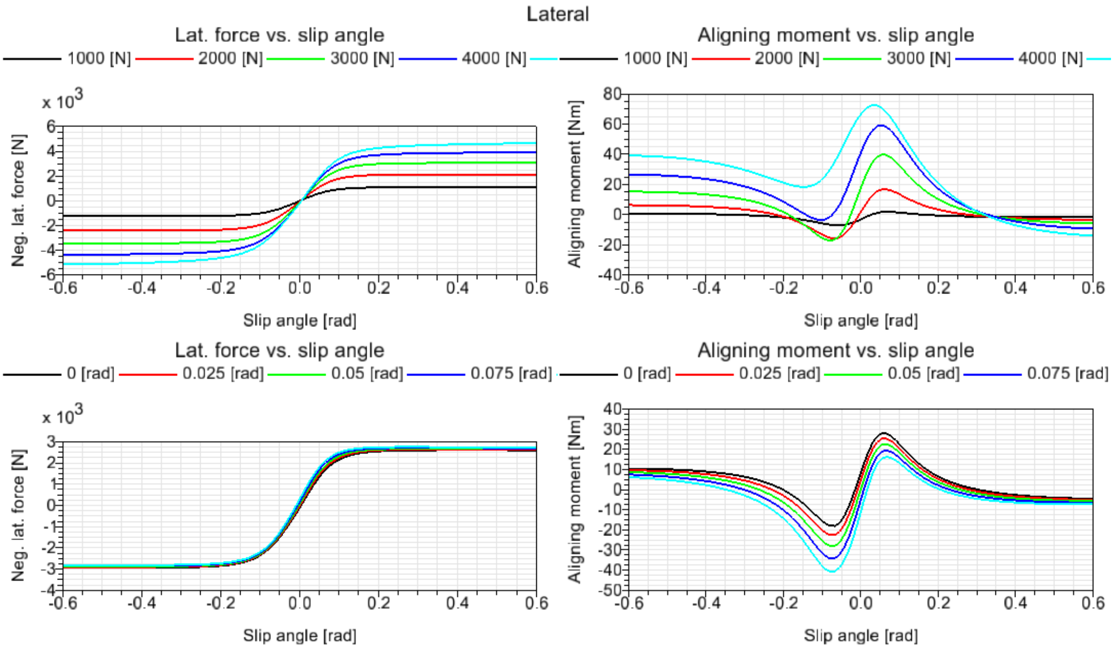


FIGURE 4.19: Magic Formula: lateral properties

In order to highlight the longitudinal and lateral responses of the tire as well as its enveloping properties, a test manoeuvre has been performed with the basic suspension model previously described and reported in Fig. 4.16. In particular a driving torque has been applied to the rim to assess the response of the tire during the acceleration phase. The driving torque is then removed allowing the tire to roll freely on the road surface. While it is freely rolling on the road surface, the tire hits a bump obstacle which is positioned at 15 m from the origin of the global reference frame (i.e. the wheel centre must travel 15 m before encountering the obstacle). The bump has 1 cm height and 2 cm length and it has been inserted in the test manoeuvre to highlight the enveloping properties of the tire model. Finally a trapezoidal yaw angle has been imposed through the angular position driver in order to assess the lateral response of the tire. The driving torque, yaw angle and road profile applied during the tire test manoeuvre are shown in Fig. 4.20.

The reference test manoeuvre has been simulated using the bdf solver available in VL.Motion. In order to be able to compare in a consistent way the Matlab implementation of the tire force element, which will be used for the scopes of this research work, to the reference TNO MF-Tire force element, a co-simulation has been set up between Simulink and VL.Motion as shown in Fig. 4.21.

In particular, the Matlab tire model schematized in Fig. 4.15 has been translated in a Simulink block and connected to the kinematic structure of the simplified suspension (wheel carrier and rim) modelled in VL.Motion. The co-simulation setup allows for the

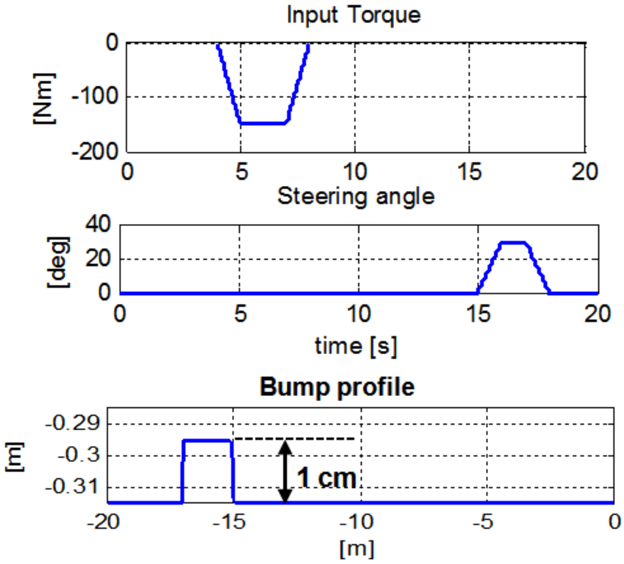


FIGURE 4.20: Tire test manoeuvre: inputs

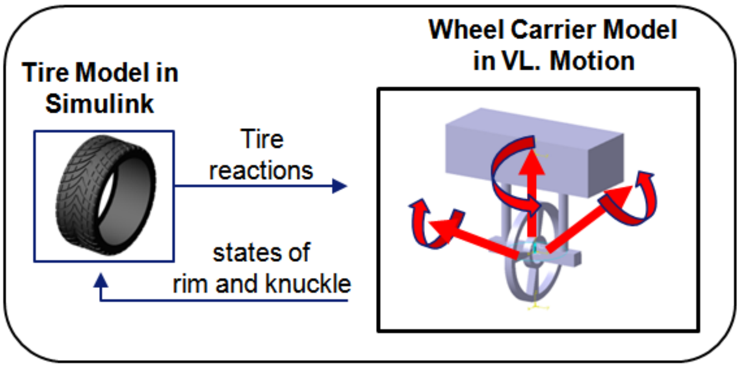


FIGURE 4.21:
Co-simulation: Simulink - VL.Motion

use of the VL.Motion bdf solver also for the numerical integration of the EOM of the simplified suspension model equipped with the tire model implemented in Simulink. Using the same solver is a basic requirement in order to be sure that potential differences in the tire responses are only due to differences in the tire models. Indeed this allows excluding any influence of numerical issues such as solver characteristics and accuracy from the comparison. The responses of the TNO tire force element and of the implemented Simulink tire model are compared in Fig. 4.22. In Fig. 4.23 the zoom of the tire responses in the time window from 9.5 sec to 10.5 sec is reported, which highlights what occurs when the tire hits the bump. The differences in the responses of the 2 models are due to the substantial simplification adopted in the implemented tire-road contact model with respect to the complex elliptical cams enveloping model employed in the TNO MF-Tire force element. However, since the modelling of an accurate tire-road contact model is beyond the scope of this research work, the global level of accuracy

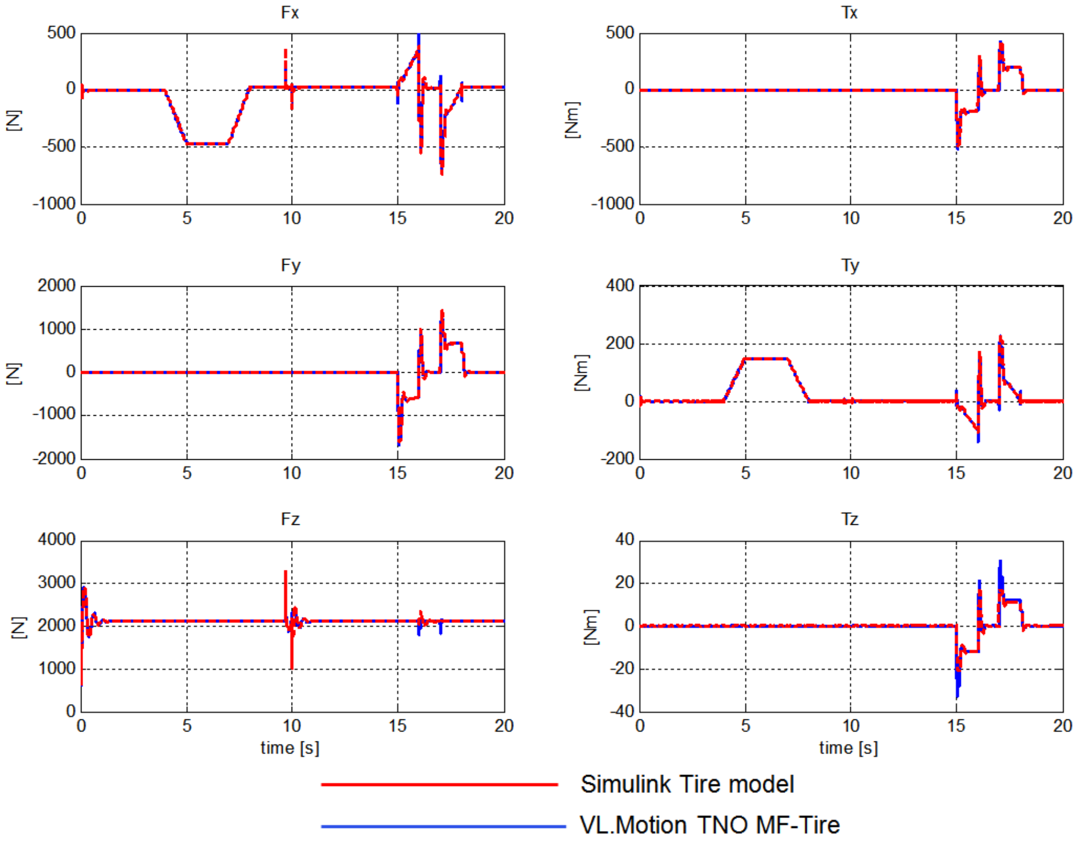


Figure 4.22: Test manoeuvre: tire responses

achieved by the simplified model in reproducing the bump responses can be considered acceptable. Similarly, as it can be appreciated in the time window from 14 sec to 19 sec in Fig. 4.24, the level of accuracy reached by the lateral responses of the implemented tire model is acceptable for the purposes of this work.

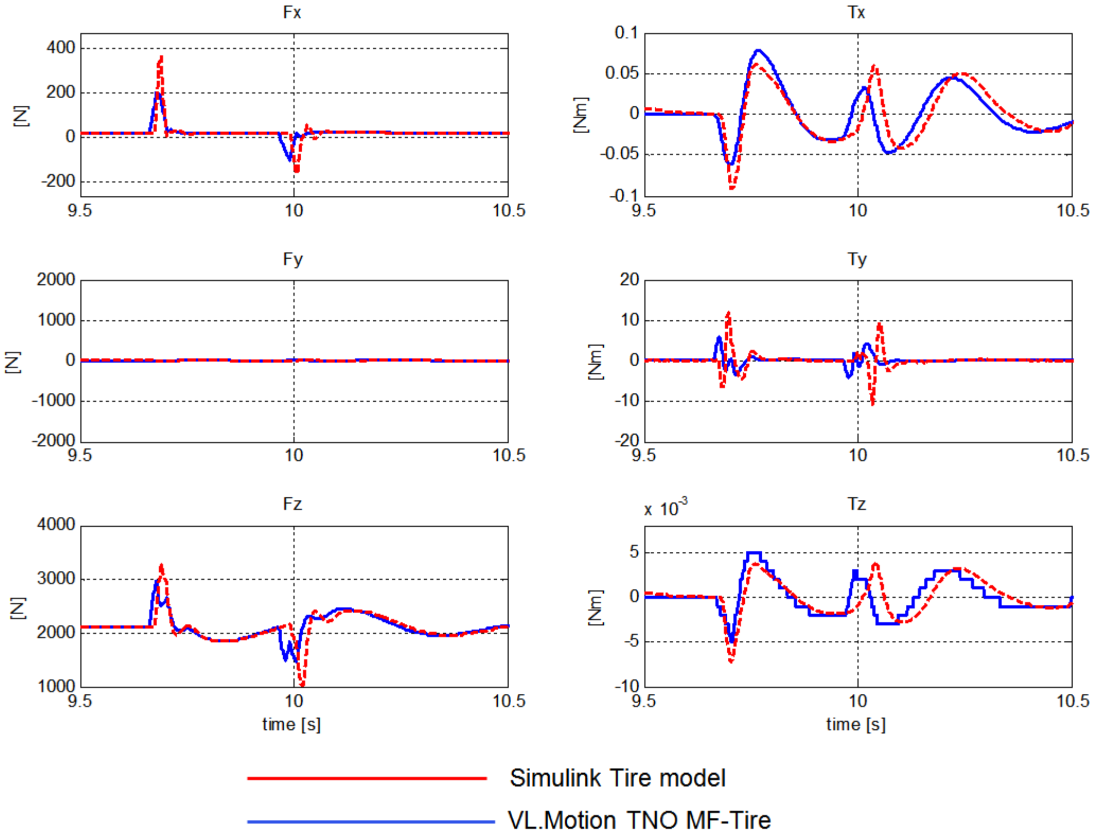


FIGURE 4.23: Test manoeuvre: bump induced responses

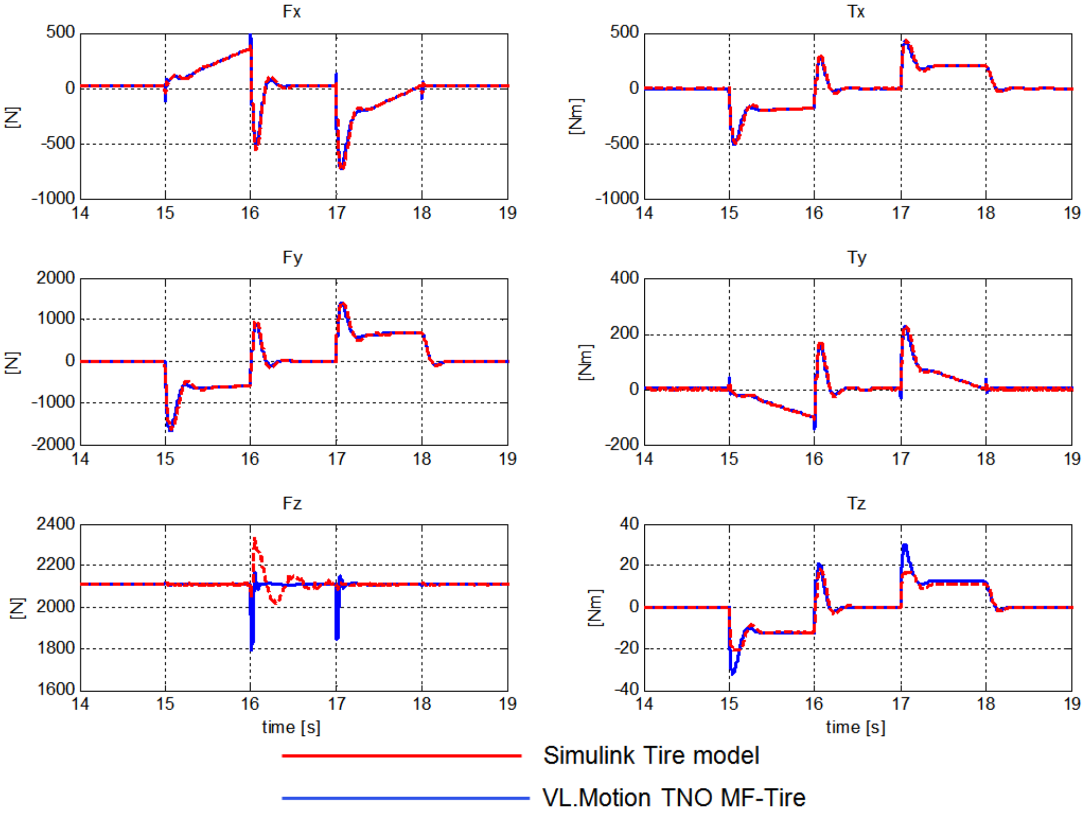


FIGURE 4.24: Test manoeuvre: response to the yaw input

4.4 Numerical results

The detailed multibody models of the rear-left suspension, i.e. the *ideal joints model* and the *bushings model*, have been implemented using the techniques described in sections 4.2 and 4.3. These 2 models can be employed at this point to carry out numerical experiments in order to assess the performances of the linearly implicit Euler method in its classical implementation (algorithm 1) which makes use of a dependent coordinates formulation, and in the modified version proposed in this research work (algorithm 2), which exploits an automatic switching from dependent to independent coordinates.

4.4.1 Test manoeuvre

A test scenario has been selected as shown in Fig. 4.25. In the simulated test manoeuvre, the vehicle falls and then settles on a flat road and then, from the rest condition, it accelerates due to a step input torque of 100 Nm applied from the chassis to the differential shaft. After the acceleration phase the vehicle incurs in a square bump with 2 cm height and 2 m length. The total simulated time is of 10 seconds. Algorithms 1 and 2 have been used for the numerical integration of the EOM associated to the *bushings model* and the *ideal joints model* in the test manoeuvre scenario under analysis using an integration time step of 1 ms.

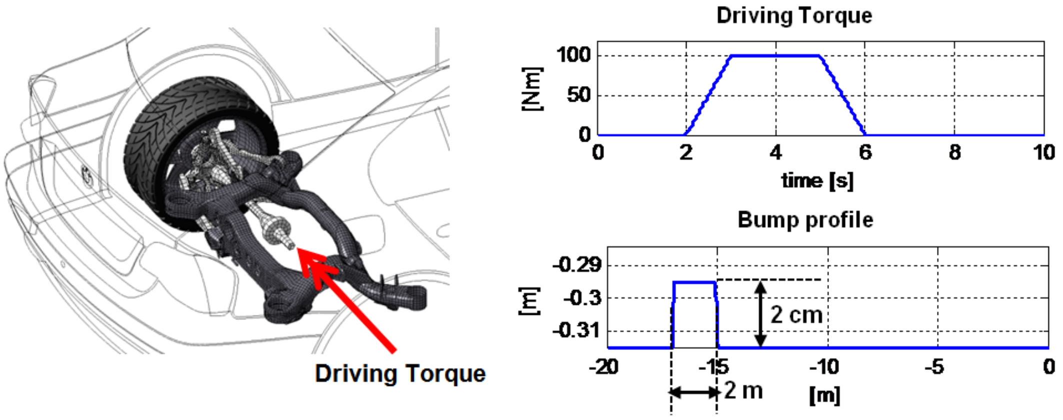


FIGURE 4.25:
Test Scenario

4.4.2 Computational time

In order to evaluate the performances of the 2 algorithms, the non-common operations in the correspondent pseudo-codes shown in sections 3 and 4 have been isolated and the associated computational times have been checked. In Tab.5 the solving steps exclusively required by algorithm 1 are reported. In particular the computational time required by each operation during the numerical integration of the *bushings model* and the *ideal joints model* is shown. It can be observed that the computational time required to find the variations in the dependent velocities and the Lagrange multipliers at the current time step greatly increases when the ideal joints model is considered. Indeed, when algorithm 1 is employed, an increase in the number of constraint equations implies the solution of a bigger system of linear equations at each time step. In the case under analysis the dimensions of the linear system to be solved at each time step increase from n+m=108 in the case of the *bushings model* to n+m=143 for the *ideal joints model.* The solving steps which are only related to algorithm 2 and the corresponding computational times employed for integrating the EOM of the *bushings model* and of the *ideal joints model* are reported in Tab.6. As pointed out before, the computation of the projection matrix $\mathbf{R}$ requires the inversion of a $n \times n$ matrix at each time step. However the associated computational burden remains constant as the number of constraint equations increases when moving from the *bushings model* to the *ideal joints model.* On the other hand, the dimensions of the linear system to be solved at each time step to find the variations in the independent velocities decrease as the number of constraint equations increases. In the case under analysis they drop from n-m=46 in the *bushings model* to n-m=11 in the *ideal joints model.* The additional numerical burden to evaluate the projection matrix $\mathbf{R}$ in algorithm 2 is thus compensated by a great reduction in the computational time required to find the variations in the independent velocities when the highly-constrained mechanical system represented by the *ideal joints model* is considered. Indeed by cross-checking the total computational times in Tab. 4.2 and Tab. 4.3 it can be noticed that the automated switching of the EOM to an independent coordinates representation improves the efficiency of the linearly implicit Euler method in the *ideal joints model* case.

Table 4.2: Algorithm 1: computational time

algorithm 1	*bushings model*		*ideal joints model*	
solving steps	time [s]	% of time	time [s]	% of time
$\mathbf{\Phi}_{\mathbf{q}_{(i+1)}}$	0.38	9.1	0.62	8.08
$\left[\Delta\dot{\mathbf{q}}_{(i)} \quad \lambda_{(i)}\right]^T$	3.81	90.9	7.05	91.92
Tot Time	4.19		7.67	

TABLE 4.3: Algorithm 2: computational time

algorithm 2	*bushings model*		*ideal joints model*	
solving steps	time [s]	% of time	time [s]	% of time
$\mathbf{T}_{(i)} = \begin{bmatrix}\mathbf{S}_{(i)} & \mathbf{R}_{(i)}\end{bmatrix} = \begin{bmatrix}\mathbf{\Phi}_{\mathbf{q}(i)} \\ \mathbf{B}\end{bmatrix}^{-\mathbf{1}}$	3.44	68.8	3.7	68.8
$\Delta\mathbf{z}_{(i)} = h\dot{\mathbf{z}}_{(i)}$	0.03	0.6	0.04	0.7
$\Delta\mathbf{q}_{(i)} = h\mathbf{S}_{(i)}\mathbf{b}_{(i)} + \mathbf{R}_{(i)}\Delta\mathbf{z}_{(i)}$	0.08	1.6	0.06	1.1
$\dot{\mathbf{q}}_{(i)} = \Delta\mathbf{q}_{(i)}/h$	0.03	0.6	0.03	0.6
$\mathbf{c}_{(i)}$	0.32	6.4	0.99	18.4
$\mathbf{S}_{(i)}\mathbf{c}_{(i)} = \mathbf{T}_{(i)}\begin{bmatrix}\mathbf{c}_{(i)} & \mathbf{0}\end{bmatrix}^T$	0.16	3.2	0.18	3.3
$\Delta\dot{\mathbf{z}}_{(i)}$	0.94	18.8	0.38	7.1
Tot Time	5		5.38	

4.4.3 Solution accuracy

Fig. 4.26 shows the norm of the vector of the constraint equations during the 10 seconds of simulation. For both algorithms 1 and 2 the errors in the constraint conditions remain bounded throughout the whole simulation if the non-iterative projection step described in Section 5 is applied. On the contrary, when the stabilization step is not applied (dotted lines), an unbounded growth of the error in the constraint conditions takes place.

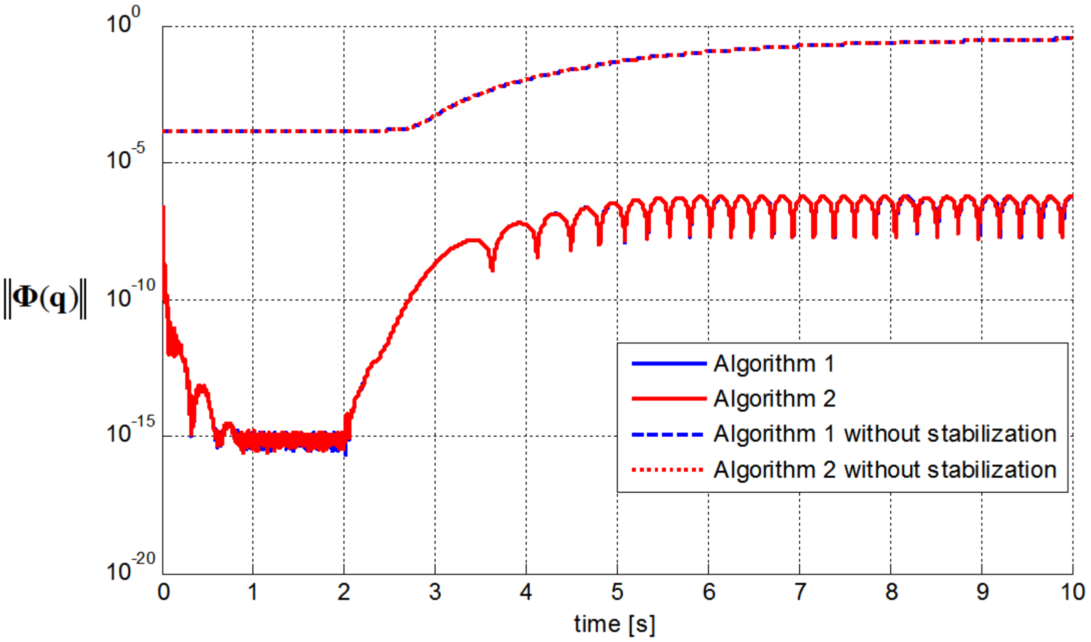


FIGURE 4.26: Constraint violations

In order to obtain a reference solution, a co-simulation has been set up between the commercial multibody software VL.Motion and Simulink as shown in Fig. 4.27. The co-simulation set-up allows for the use of the same tire model in both the reference simulation and the test simulations obtained by means of algorithm 1 and algorithm 2. In particular the non-linear tire model previously described has been implemented in a Simulink block and integrated in the remaining part of the suspension modelled using VL.Motion. Two reference solutions have been obtained, one for the *bushings model* and one for the *ideal joints model*.

To verify the accuracy of the solutions obtained using algorithms 1 and 2 the displacements, velocities and accelerations at the CG of the chassis along the vertical direction are compared with the reference solution in Fig. 4.28. A zoom of the responses in the time window between 8 and 9.5 sec is shown in the graphs on the right which highlights the vertical response of the chassis when the wheel hits the bump. Results reported in Fig. 4.28 show that the responses obtained using algorithm 1 and 2 correctly match the reference solution. Note that in Figure 4.28 and in the remaining of this Chapter only the simulation results related to the *bushings model* will be considered since they have shown the same level of accuracy of results related to the *ideal joints model*.

The bushing reaction forces at the fore-left bushing connection between chassis and subframe are compared in Figure 4.29. When the wheel hits the bump, the results obtained with both the implementations of the linearly implicit method poorly match the reference solution as can be appreciated in the zooms shown in the right graphs. This is in line with what has been shown in literature on the accuracy of the linearly implicit Euler integration method [30] and is caused by the fixed step size of 1 ms employed during the numerical integration. The same behaviour has been observed for the bushing reaction torques as shown in Fig.4.30.

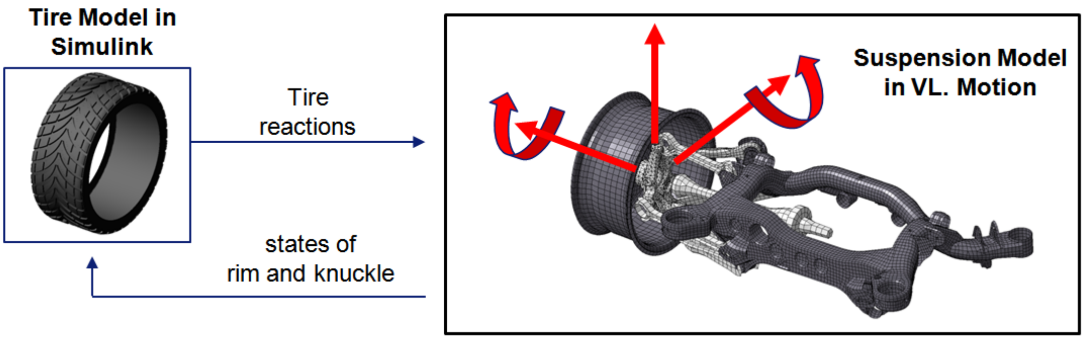


FIGURE 4.27:
Co-simulation set up

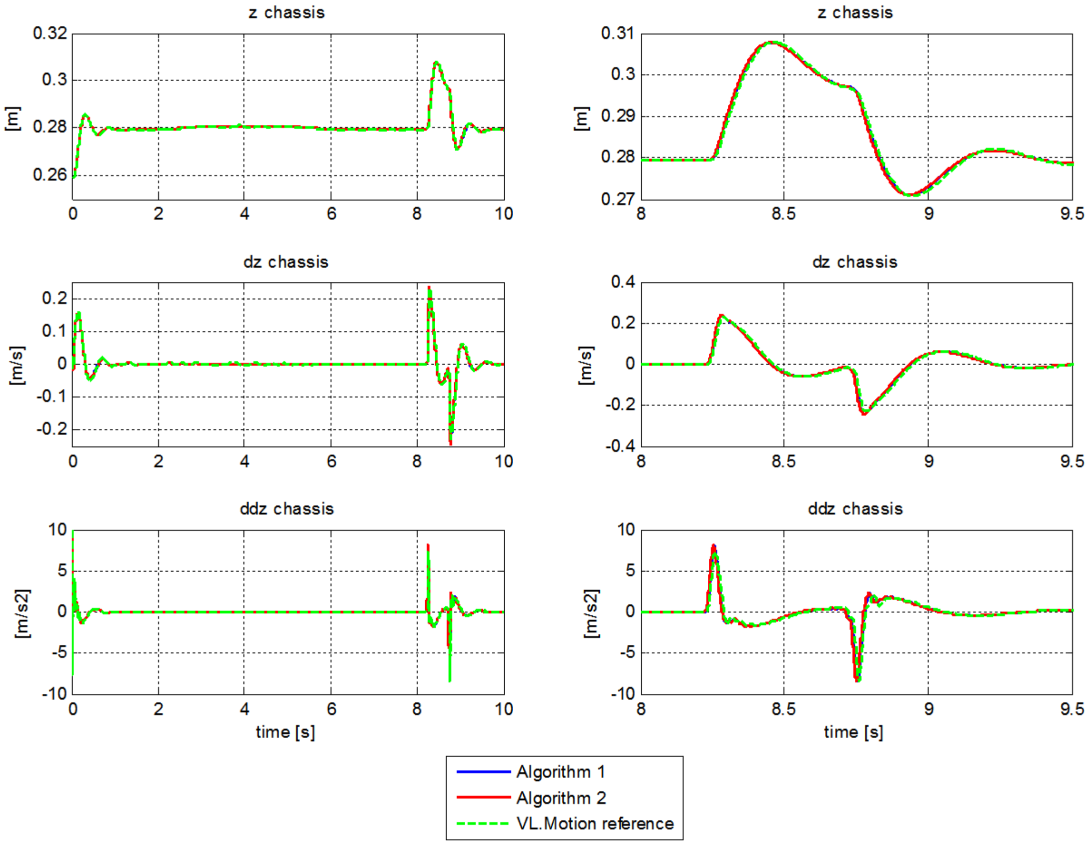


FIGURE 4.28: Vertical position, velocity and acceleration at the chassis CG

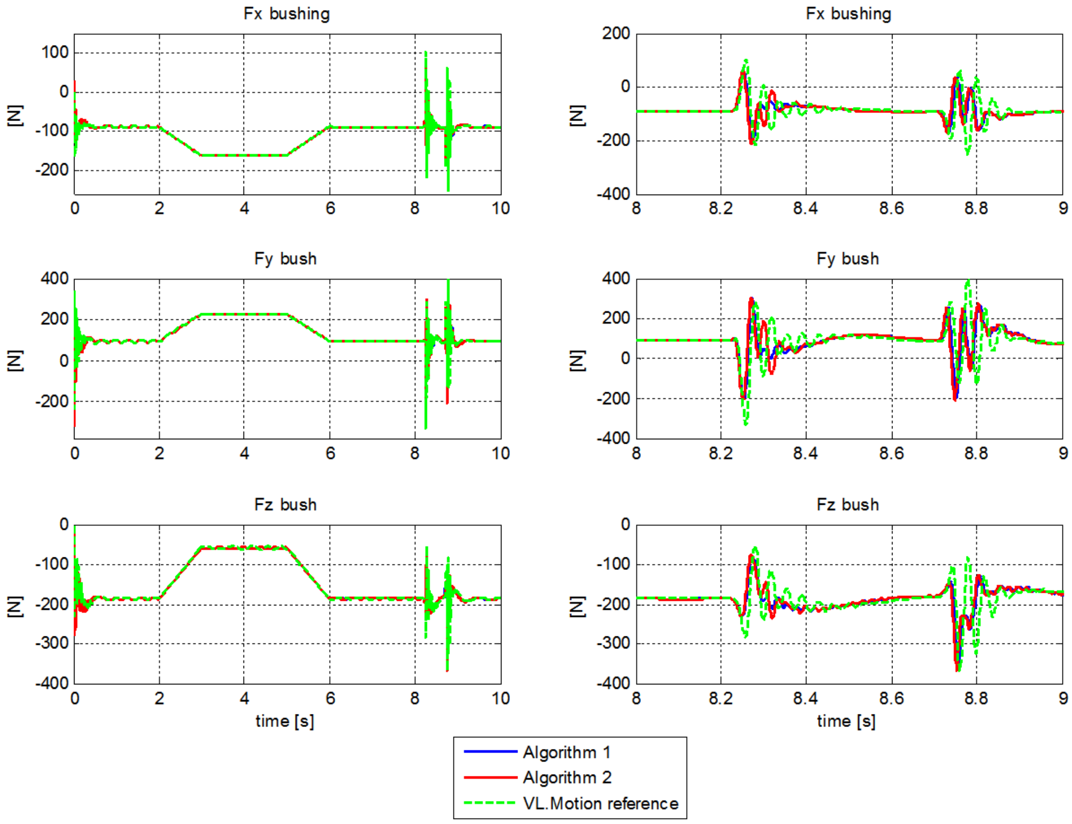


FIGURE 4.29: Reaction forces at the fore-left bushing connection

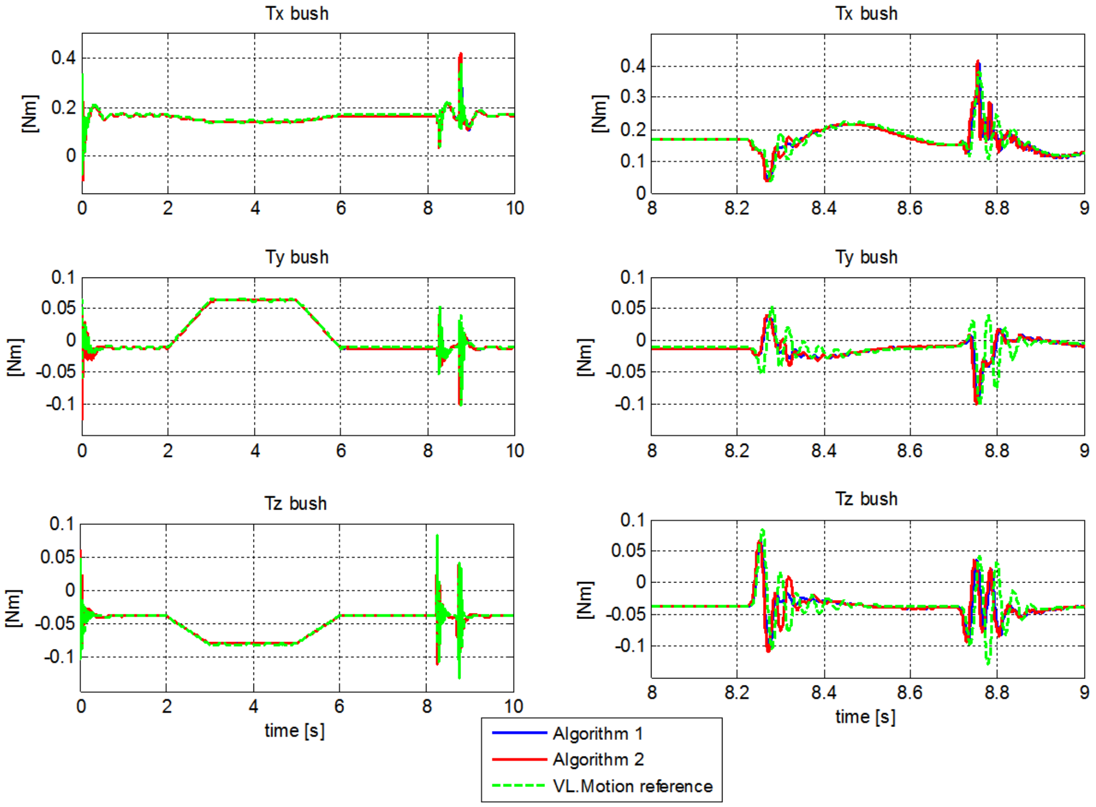


FIGURE 4.30: Reaction torques at the fore-left bushing connection

4.5 Summary

In this Chapter the two implementations of the linearly implicit Euler method discussed in Chapter 3 have been employed for the numerical integration of the stiff differential-algebraic equations of motion associated to an industrial rear left suspension of a passenger car in a selected test scenario. In particular two models have been developed, the *ideal joints model* and the *bushings model* which have the same number of configuration parameters n but a considerably different number of constraint equations m.

Several issues associated to the modelling of these 2 multibody systems, which has been carried out by the combined use of the symbolic algebra software Maple and the Matlab numerical computing environment, have been addressed. The process required to obtain the symbolic equations of motion of the 2 models has been described, starting from the definition of bodies, constraint equations and associated jacobians. The definition of the force elements in the vector of generalized forces $\mathbf{Q}$ has been also described, particularly focusing on the implementation of the bushing routine which not only computes the reactions of the bushings elastic connections, but populates also the jacobians matrices of vector $\mathbf{Q}$. Given the crucial importance of the tire force element on the accuracy of the overall dynamic simulation, a thoroughly description of the tire model implemented in this research work has been also provided together with its validation against the TNO MF-Tire reference model.

The numerical results in terms of integration time have been presented, showing that the efficiency of the linearly implicit Euler method is enhanced by the automatic switching from dependent to independent coordinates proposed in Chapter 3 when dealing with a highly constrained multibody systems.

Also the violation of the constraint conditions has been checked showing that the use of the non-iterative projection step described in Chapter 3 guarantees the errors in the constraint conditions to remain bounded throughout the whole simulation.

Finally the accuracy of both algorithms 1 and 2 has been checked by comparison against the results obtained using the commercial multibody software VL.Motion in co-simulation with the tire model implemented in Simulink. Results in terms of vertical positions, velocities and accelerations at the CG of the chassis show that the two implementations of the linearly implicit Euler method guarantee the same acceptable level of accuracy. However an analysis of the bushing reactions has shown that both the implementations of the linearly implicit Euler method fail in accurately reproduce the bushing's forces and torques reactions when a severe dynamic event occurs (i.e. the tire hitting the bump). This is however in line with what has been shown in other researches on the accuracy of the linearly implicit Euler method.

Chapter 5

Efficient Concept Modelling of the Suspension

After investigating a possible approach to increase the efficiency of the LI Euler integration method by means of an automated switching from a dependent to an independent coordinates formulation in Chapters 3 and 4, the reduction of a complex multibody model of the suspension system into an equivalent simplified concept model will be addressed in this Chapter.

Concept modelling techniques are based on the efficient modelling of the suspension system by means of a simplified concept model, this leading to a great reduction in the complexity of the associated EOM while still reproducing the overall physical behaviour of the original system with a satisfactory degree of accuracy. Reducing the complexity of the multibody system while preserving its ability to accurately reproduce the dynamics of the starting system represents a challenge which has been addressed by a multitude of research works on real-time applications. Moreover a great deal of research is still addressing this approach since it remains the most adopted in industrial automotive applications due to the advantages offered by simplified concept models in applications such as design optimization, controller tuning and parameters identification. Indeed, due to their iterative nature these applications may require a considerably high number of multibody simulations which could not be performed using detailed multibody models because of the prohibitive amount of integration time they would require.

An investigation on the potential benefits coming from a trailing-arm conceptual representation of the suspension system will be conducted in this Chapter. The ultimate goal is the developing of trailing-arm concept models of the quarter-car suspension and of the full vehicle which are able to furnish accurate predictions of both the ride behaviour and the dynamic phenomena occurring during longitudinal acceleration and longitudinal braking manoeuvres. An identification procedure will be also proposed in order to estimate the unknown design parameters of the concept models using the outputs of ride tests performed on a reference model.

5.1 Trailing-arm concept modelling of a quarter-car suspension system

Before addressing the full vehicle problem, the quarter-car rear left suspension presented in Chapter 4 and reported in Fig. 5.1 will be considered as the detailed reference suspension model to be converted into an equivalent trailing-arm concept suspension. The final goal is the definition of a very simplified concept model which can be used for design and optimization studies of the vertical ride behaviour of the suspension as well as of its performances during acceleration and braking on a straight path.

The 2 DOFs quarter-car model in Fig. 5.2, which is generally employed in the design of active suspension controllers, cannot be used to study the longitudinal dynamics of a detailed suspension model since it does not contain any information on the kinematics of the suspension which determines how the longitudinal forces developed at the tire road contact patch are transmitted into the chassis as it will be explained in the next section.

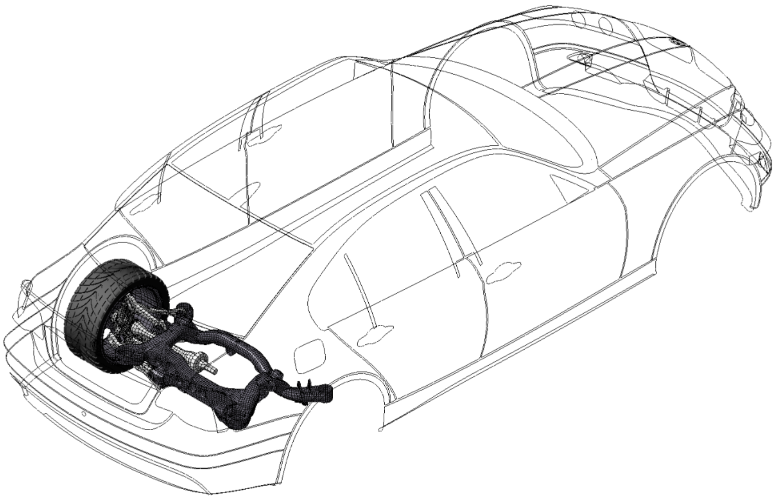


FIGURE 5.1: Detailed quarter-car suspension

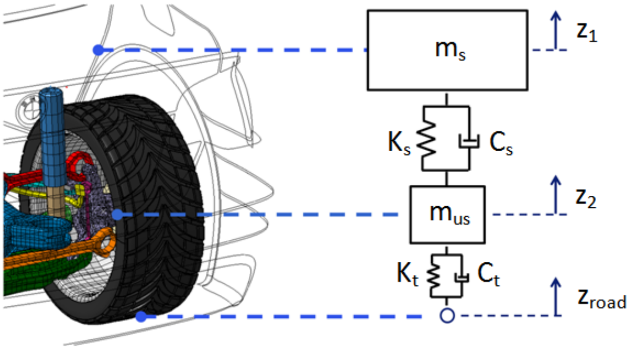
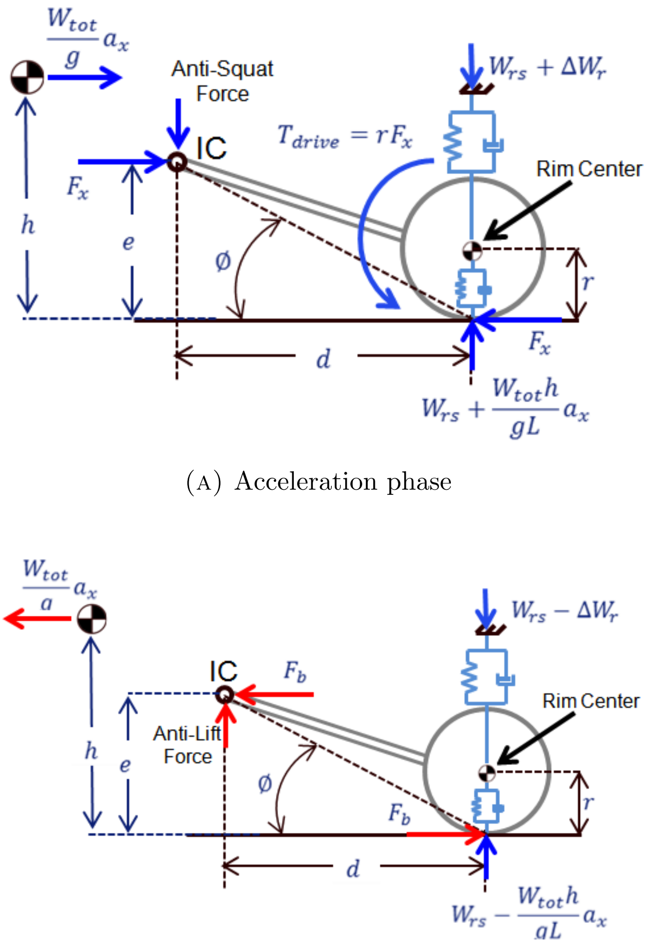


FIGURE 5.2: Quarter-car model

5.1.1 Squat, Dive and Lift Phenomena

In order to understand how the suspension kinematics influences the vehicle dynamic behaviour during acceleration and breaking it is convenient to analyse the suspension system in the side view plane which is defined as the vertical plane passing through the wheel center and parallel to the vehicle centreline. For any suspension it is possible to identify an IC (Instant Center of rotation) which is a virtual reaction point where the reactions of the suspension's links on the chassis can be solved into equivalent longitudinal and vertical forces. Any suspension system can be represented in the side view plane as an equivalent trailing-arm hinged to the chassis at the IC point. The position of the IC in the side view plane varies with the wheel's vertical travel according to the kinematics of the suspension linkage.

In Fig. 5.3a the side-view free-body diagram of a rear independent suspension during the acceleration phase is shown. The vertical load acting on the wheel is given by the static load on the rear axle W_{rs} plus the vertical load transferred from the front to the rear axle due to the longitudinal acceleration a_x. The amount of vertical load transfer depends only on the longitudinal acceleration, the height of the vehicle's center of gravity h_{CG} and the wheelbase L and it is not influenced by the suspension kinematics. The longitudinal traction force F_x developed at the tire-road contact patch is due to the drive torque T_{drive} imposed to the wheel by the differential through the half shaft. Since the differential is mounted on the vehicle body the drive torque must also be inserted into the free-body diagram of the suspension. Due to the forward acceleration there is an increase in the vertical load acting on the suspension's spring which causes a deflection in the rear suspension. This bump movement of the rear suspension is called *power squat* and is influenced by the positioning of the IC in the Side View plane. By applying the second Newton's law for the moments with respect to the IC point, it is possible to find the change in the vertical load acting on the rear suspension's spring:



(A) Acceleration phase

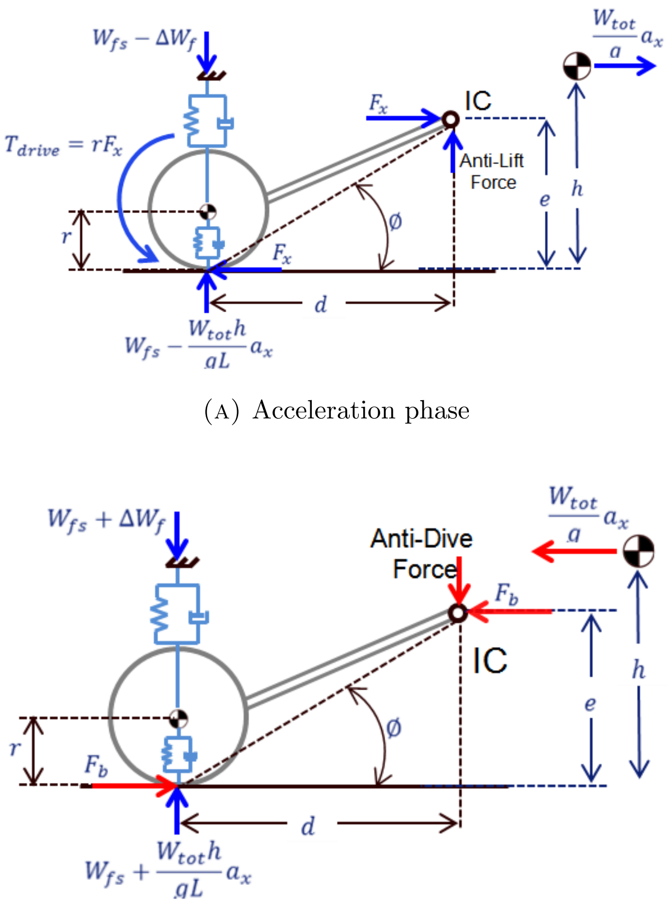
(B) Braking phase

Figure 5.3: Rear suspension free-body diagrams in the side view plane during acceleration and braking

$$\Delta W_r = \frac{W_{tot} \cdot h_{CG}}{g \cdot L} \cdot a_x - F_x \frac{(e-r)}{d} = \Delta F_z - F_{anti-squat} \tag{5.1}$$

The first term in the RHS (Right Hand Side) of Eq. 5.1 is the vertical load transferred from the front to the rear axle. The second term, which depends on the horizontal and vertical positions e and d of the IC in the side view plane, is called *anti-squat* force since it reduces the change in the vertical load and consequently the compression of the rear suspension's spring during acceleration.

A similar analysis can be conducted in the case of braking as shown in the bottom free-body diagram of Fig. 5.3b. When outboard brakes are considered, as in the present case, the braking torque is not present into the free-body diagram since brakes are mounted on the knuckles and the braking torque is thus transmitted to the chassis by the suspension



(A) Acceleration phase

(B) Braking phase

FIGURE 5.4: Front suspension free-body diagrams in the side view plane during acceleration and braking

links. During forward braking the rear suspension's links can develop an anti-lift force which reduces the suspension rebound or lift movement:

$$\Delta W_r = \frac{W_{tot} \cdot h_{CG}}{g \cdot L} \cdot a_x - F_b \frac{e}{d} = \Delta F_z - F_{anti-lift} \tag{5.2}$$

By conducting the same analysis on the front suspension it is possible to verify how the IC position in the side view plane determines the amount of anti-lift force developed during the acceleration phase for a front wheel drive vehicle (Fig. 5.4a) as well as the amount of anti-dive force which opposes to the suspension bump or dive movement during forward braking (Fig. 5.4b). In particular the reduction in the vertical load acting on the front suspension during acceleration is given by:

$$\Delta W_r = \frac{W_{tot} \cdot h_{CG}}{g \cdot L} \cdot a_x - F_x \frac{e}{d} = \Delta F_z - F_{anti-lift} \tag{5.3}$$

while the increase in the vertical load on the suspension springs during braking is given by:

$$\Delta W_r = \frac{W_{tot} \cdot h_{CG}}{g \cdot L} \cdot a_x - F_b \frac{e}{d} = \Delta F_z - F_{anti-dive} \tag{5.4}$$

5.1.2 Implementation of the trailing-arm quarter-car model

Based on the analysis conducted in the previous section, a trailing-arm representation of the suspension system has been chosen in order to develop a concept model suitable for both vertical ride studies and assessment of the longitudinal acceleration and braking performances of the suspension.

The model, which is shown in Fig. 5.5, has been implemented in Maple/Matlab using the same approach adopted for the implementation of the detailed rear left suspension as described in Chapter 4. In particular the model is composed by 3 bodies which are the chassis, the trailing-arm and the rim whose positions and orientations are described by a total of $n = 3 \cdot 7 = 21$ configuration parameters. As in the case of the detailed suspension model, the rotations of the chassis are restrained using 4 algebraic constraint equations. Two more constraint equations define the dependencies between the Euler parameters of the trailing-arm and of the rim. Two revolute joints are used to connect the trailing-arm to the rim and to the chassis. The total number of constraint equations is thus m=16 leaving a total of f=5 dynamic degrees of freedom which are the 3 translations of the sprung mass, the relative rotation of the trailing-arm w.r.t. the sprung mass and the rotation of the rim. The tire force element discussed in Chapter 4 has been used in the model. A spring-damper force element has been defined between the trailing-arm and the chassis in order to reproduce the global stiffness and damping properties of the suspension. Finally, since we are dealing with a rear wheel drive vehicle, the traction torque T_{drive} has been applied from the chassis to the trailing-arm and from the trailing-arm to the rim. The revolute joint applied between trailing-arm and chassis defines the position of the suspension's IC in the side-view plane. In particular the position of the IC can be controlled by means of the design parameters d_r and e_r. It is important to notice that the variation of the IC position in the side view plane due to the changes in the suspension travel is not taken into account in the proposed model.

The list of design parameters which completely define the trailing-arm quarter-car model of Fig. 5.5 is the following:

- sprung mass and its CG position
- Unsprung mass and its CG position

- d_r which defines the longitudinal position of the IC in the side view plane with respect to the Unsprung CG;
- e_r which defines the vertical position of the IC in the side view plane with respect to the Unsprung CG;
- K_r which defines the linear stiffness of the suspension;
- L_{0r} which defines the vertical preload in the suspension;
- C_r which defines the linear damping in the suspension;
- tire parameters;

The sprung and unsprung masses and the correspondent CG positions in the design configuration can be easily extracted from the detailed suspension model. In particular the masses of the suspension's links can be equally split between sprung and unsprung masses. Also the tire properties can be inherited from the detailed suspension model.

The remaining design parameters, which define the position of the IC in the side view plane (d_r, e_r) and the global elastic and damping properties of the suspension (K_r, L_{0r}, C_r) are not directly available in the detailed suspension model. Indeed the global stiffness and damping properties of the suspension are synthesis parameters depending on the shock-absorbers characteristics as well as on the contribution of the elastic bushing connections at the suspension's links. The kinematic structure of the suspension also influences its global stiffness and damping properties as described in [60, 61]. Moreover the IC position in the side view plane can be estimated only by means of a detailed kinematic analysis of the suspension linkage which depends on the type of suspension under analysis and requires the availability of accurate geometric data defining the positions of the suspension's hard points.

In order to obtain these unknown parameters in a straightforward way, an identification process has been set up as it will be described in the next subsection.

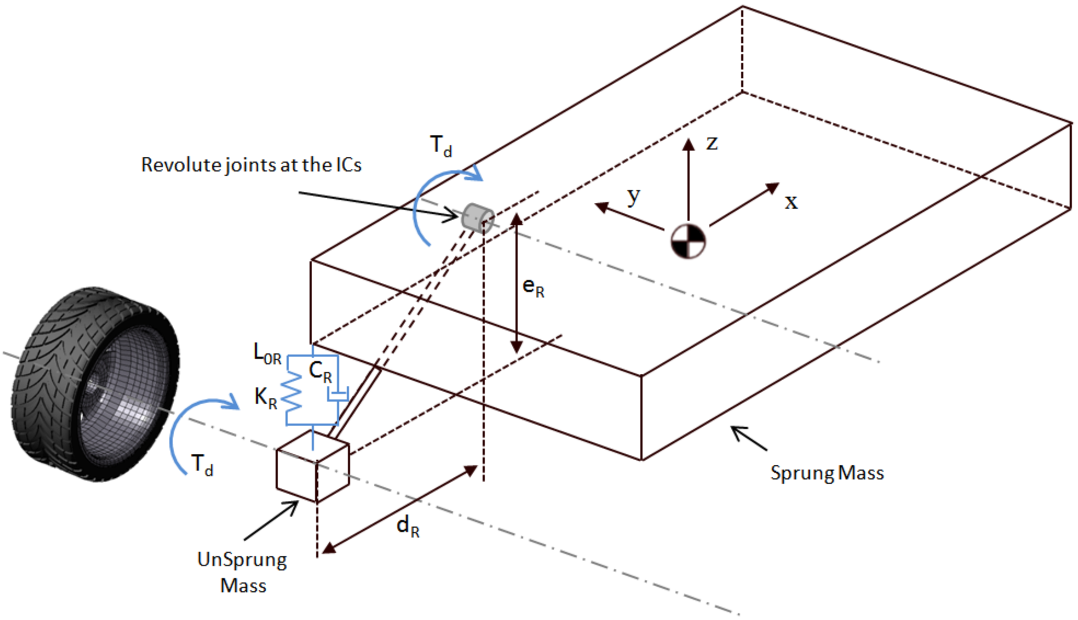


FIGURE 5.5: Trailing-arm quarter-car model

5.1.3 Parameters identification of the trailing-arm quarter-car model

The identification procedure employed to estimate the unknown design parameters is based on the comparison of the dynamic response of the concept model with that of the reference model. Once the dynamic behaviour of the reference model has been measured during a certain test manoeuvre, the same manoeuvre can be reproduced on the multibody concept model and its design parameters can be tuned in order to obtain the same dynamic response shown by the reference model.

A proper test manoeuvre has been selected in order to highlight the anti-properties and the vertical ride behaviour of the detailed quarter-car reference model. The test manoeuvre is constituted by 4 different phases:

- **phase 1 [0-6 sec]**: the quarter-car suspension settles on a rough road classified as B according to the ISO 8608 [62].
- **phase 2 [6-10 sec]**: a trapezoidal driving torque input with a maximum value of 500 Nm is applied from the chassis to the differential shaft producing a longitudinal acceleration of the quarter-car suspension.
- **phase 3 [10-15 sec]**: the driving torque in input is removed and the quarter-car suspension is allowed to proceed freely along the forward direction simulating the engaging of the neutral gear.
- **phase 4 [15-18 sec]**: a braking torque of 350 Nm is applied from the knuckle to the rim causing a longitudinal deceleration of the quarter-car suspension.

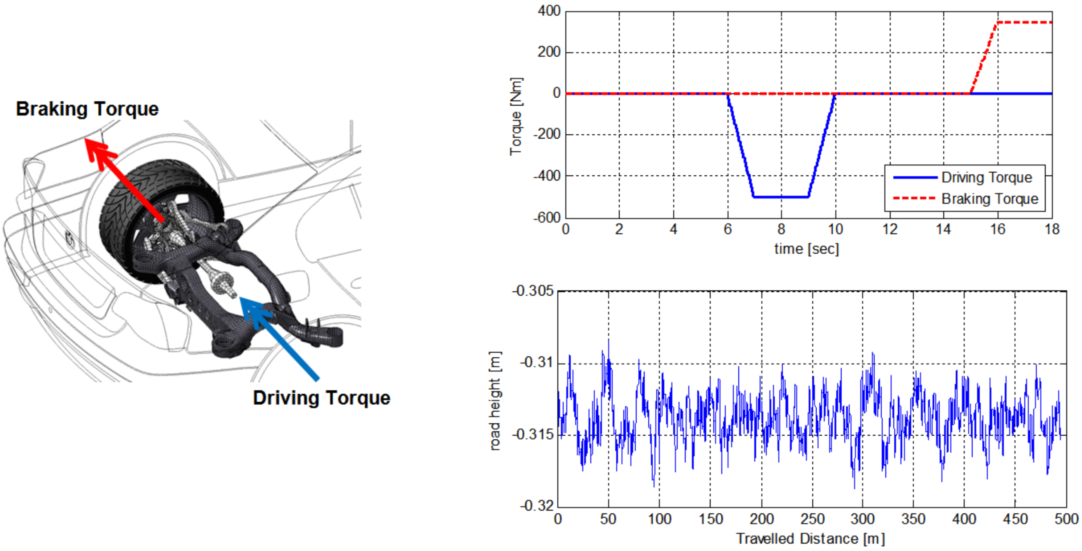


Figure 5.6: Inputs for the reference test manoeuvre

The driving and braking torques applied during the test manoeuvre are shown in Fig. 5.6 together with the B-type road profile. The test manoeuvre has been performed on the *ideal joints model* discussed in Chapter 4, and the correspondent dynamic response has been measured. In particular the vertical positions and velocities of the chassis CG have been selected to describe the dynamic response of the reference model. These quantities will be furnished in input to the identification algorithm and will represent the target dynamic response to which the concept model will be tuned.

Once the target dynamic response has been obtained, it is possible to start the parameter identification process. This latter has been carried out in Matlab by using the 'lsqnonlin' function. In particular a function called 'Quarter_trailing_arm_fun' has been implemented and given in input to the lsqnonlin routine. As shown in Fig. 5.7, this function receives in input the target dynamic response together with the current set of design parameters grouped in the vector **b** and performs the integration of the EOM of the concept model using the same driving and braking torques and the same road profile specified in the test manoeuvre.

The vertical displacements and velocities of the concept model's sprung mass obtained during the dynamic simulation are than compared with those coming from the test manoeuvre in order to obtain 2 cumulative errors, i.e. the sum of the errors between the reference and the concept responses throughout all the simulated test. The cumulative errors in the vertical positions and vertical velocities of the sprung mass are indicated respectively as $f_1(\mathbf{b})$ and $f_2(\mathbf{b})$ in Fig. 5.7 in order to highlight their dependency on the current set of design parameters grouped in the vector **b**.

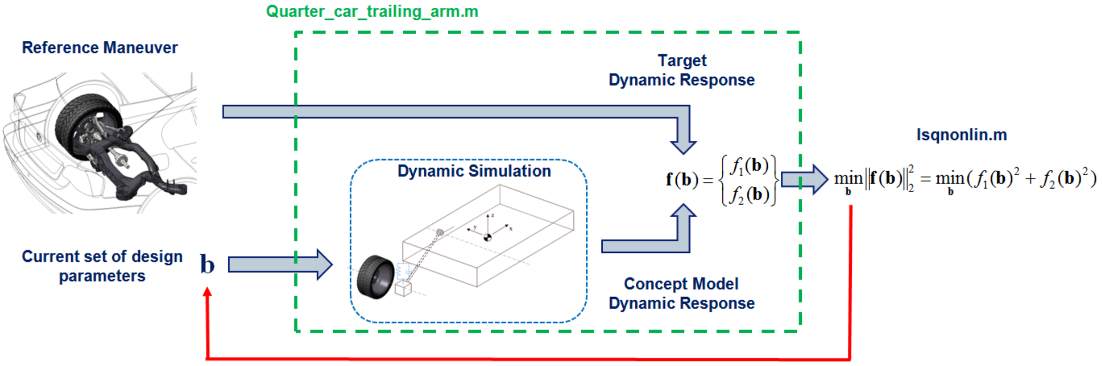

Quarter_car_trailing_arm.m
Reference Maneuver
Target
Dynamic Response
lsqnonlin.m
Dynamic Simulation
$\mathbf{f}(\mathbf{b}) = \begin{Bmatrix} f_1(\mathbf{b}) \\ f_2(\mathbf{b}) \end{Bmatrix}$
$\min_{\mathbf{b}} \|\mathbf{f}(\mathbf{b})\|_2^2 = \min_{\mathbf{b}}(f_1(\mathbf{b})^2 + f_2(\mathbf{b})^2)$
Current set of design parameters
b
Concept Model
Dynamic Response

FIGURE 5.7: Matlab parameters identification process

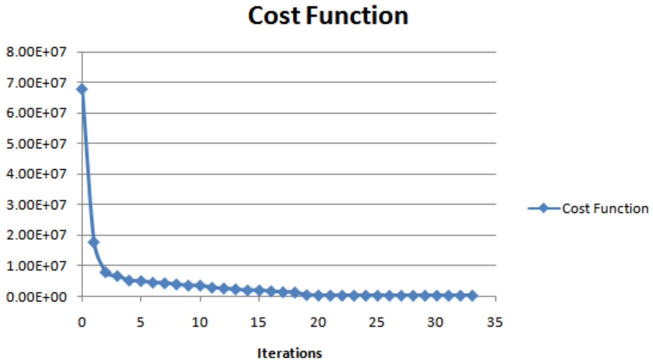


FIGURE 5.8: Cost function

The 'lsqnonlin' Matlab function finds the minimum to the following optimization problem :

$$\underset{\mathbf{b}}{min} \left\|\mathbf{f}(\mathbf{b})\right\|_2^2 = \underset{\mathbf{b}}{min}(f_1(\mathbf{b})^2 + f_2(\mathbf{b})^2) \tag{5.5}$$

In particular after 33 iterations the identification algorithms returns the vector of identified parameters which minimizes the cost function. The evolution of the cost function is reported in Fig. 5.8.

The target dynamic responses obtained with the *ideal joints model* have been compared with those of the identified trailing-arm model in Fig. 5.9. The identified trailing-arm concept model (blue curves) is able to correctly reproduce the anti-squat and anti-lift properties of the detailed suspension model (red curves). Indeed during the accelerations phase [6-10 seconds], the anti-squat force developed by the suspension links causes the the chassis to move upwards while the opposite phenomenon occurs during braking due to the anti-lift force. Moreover, the trailing-arm concept model reproduces the vertical ride behaviour of the suspensions with a satisfying degree of accuracy. In Fig. 5.9 also the responses of the trailing-arm concept model before the identification, i.e. with

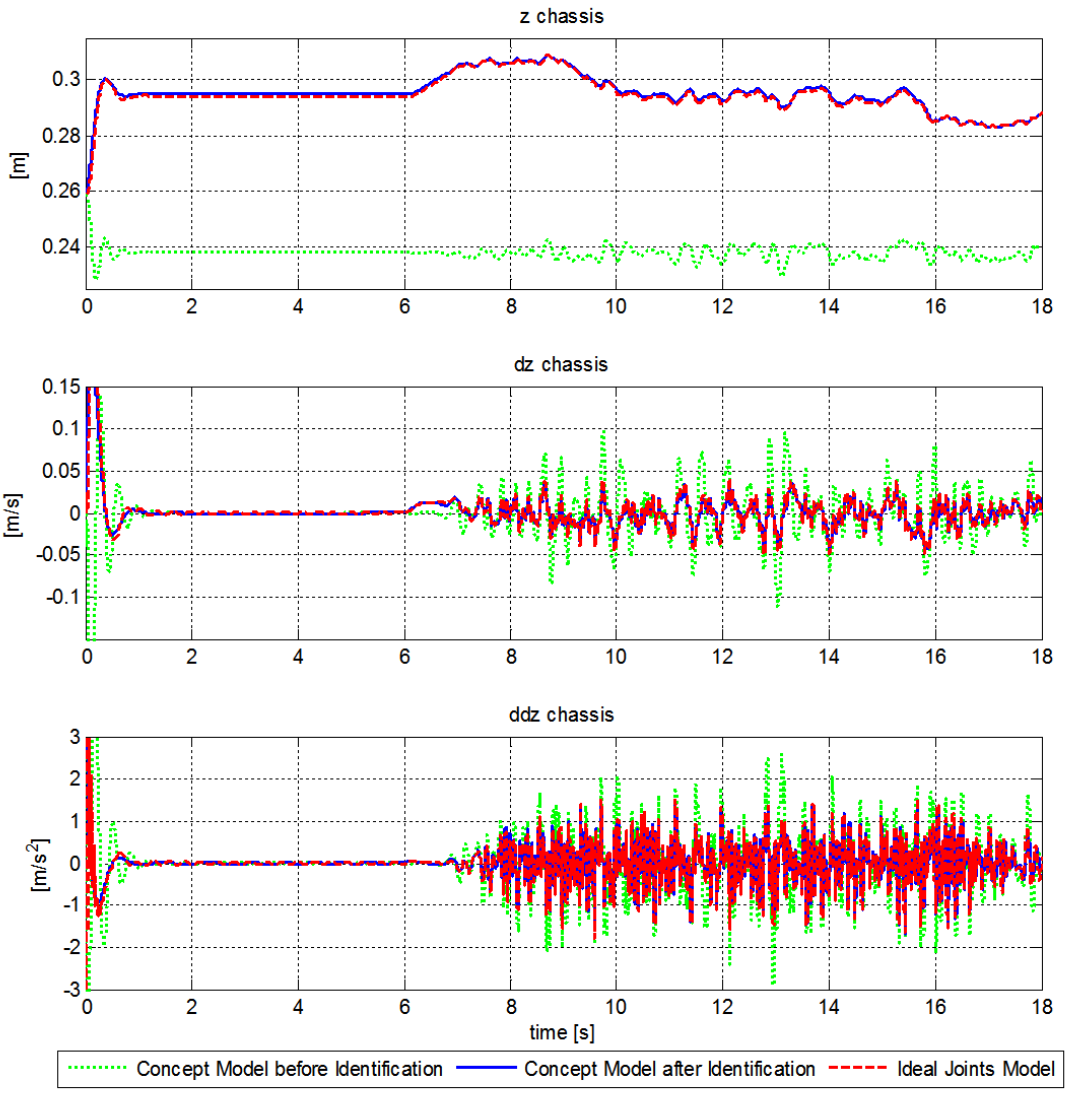


FIGURE 5.9: Dynamic responses during the test manoeuvre

an initial guess of the design parameters values, is reported (green curves) in order to highlight the potential errors which may be due to an incorrect selection of the design parameters. A zoom of the responses between 8 and 10 seconds is also reported in Fig. 5.10. In Tab. 5.1 the RMS (Root Mean Square) errors in the vertical positions, velocities and accelerations at the CG of the chassis are reported both before and after the identification process.

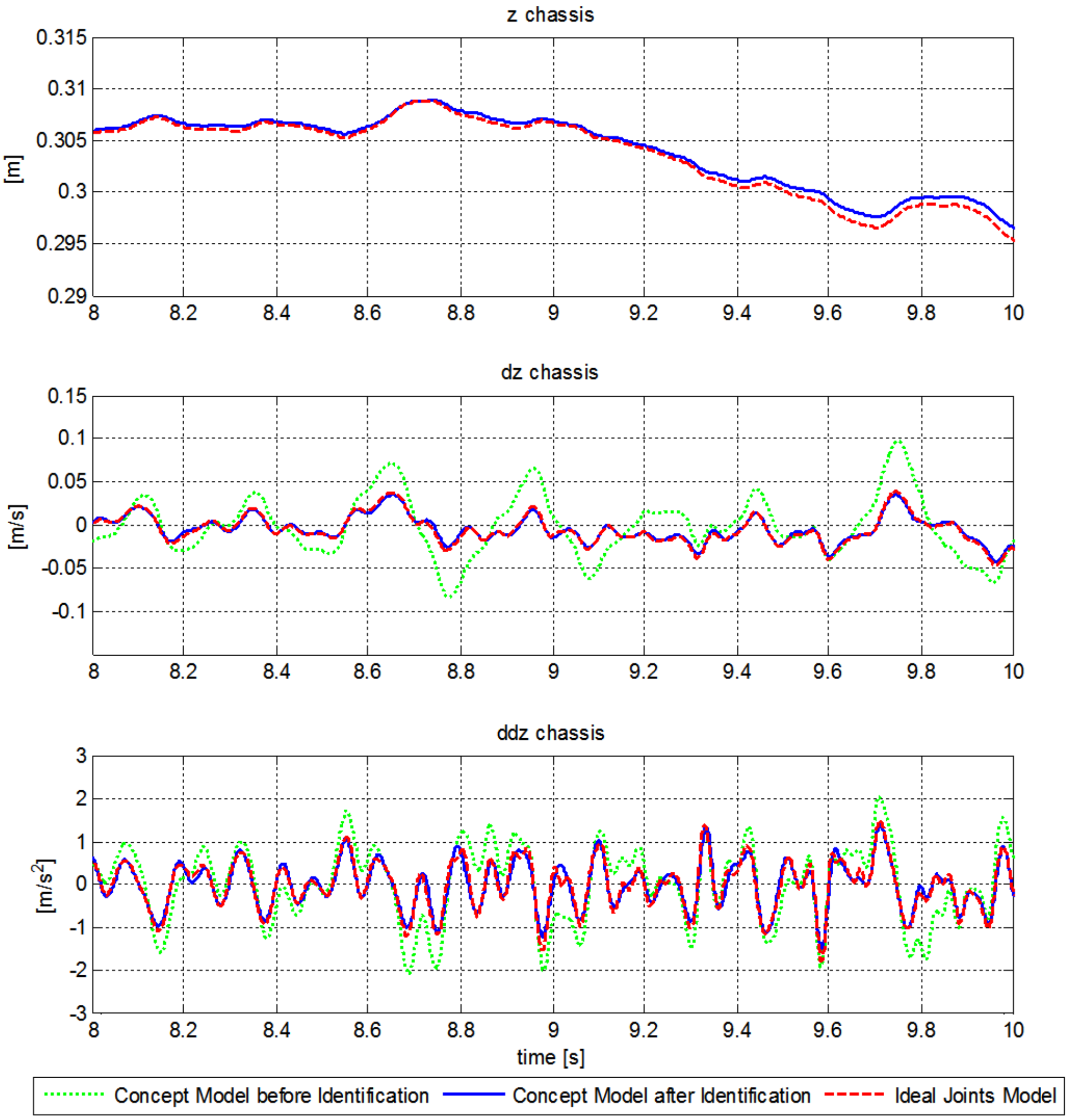


FIGURE 5.10: Zoom of the dynamic responses during the test manoeuvre [8s-10s]

TABLE 5.1: Test Manoeuvre:
RMS errors between the Concept and the Ideal Joints models responses

	RMS errors before identification	**RMS errors after identification**
chassis CG vertical positions	0.2370 $[m]$	0.0260 $[m]$
chassis CG vertical velocities	0.1145 $[m/s]$	0.0425 $[m/s]$
chassis CG vertical accelerations	0.5191 $[m/s^2]$	0.2583 $[m/s^2]$

5.1.4 Validation of the identified trailing-arm quarter-car model

In order to validate the identified concept model 2 validation manoeuvres have been selected in which different values for the driving and braking torques are assigned. Also the roughness of the road will be varied with respect to the reference test manoeuvre in order to assess the robustness of the proposed approach. The characteristics of the considered validation manoeuvres are the followings:

- **validation manoeuvre 1**: maximum driving torque 300 Nm - braking torque 150 Nm - road profile classified as A in the ISO 8608;
- **validation manoeuvre 2**: maximum driving torque 400 Nm - braking torque 250 Nm - road profile classified as C in the ISO 8608;

The dynamic responses of the *ideal joints model* and of the identified trailing-arm concept model are compared in Figs. 5.11 and 5.12 which correspond respectively to the validation manoeuvres 1 and 2. In both cases the accuracy of the identified concept model in reproducing the anti-features of the suspension and its vertical ride properties is acceptable. The RMS errors in the vertical responses of the chassis for the two validation manoeuvres are reported in Tab 5.2.

A comparison has been also done to assess the potential increase in simulation efficiency when using the identified trailing-arm model instead of its detailed counterpart, i.e. the *ideal joints model*. It is worth noticing at this point that the integration of the EOM of the trailing-arm concept model can be carried out by means of the explicit Euler integration scheme with the additional non-iterative projection step described in Chapter 3 to avoid the violation of the constraint equations, since no stiff force elements are contained in this simplified model. This greatly reduces the numerical burden during numerical integration since the evaluation of the Jacobians is completely avoid. Moreover the great reduction in the number of configuration parameters required to describe the trailing-arm concept model (i.e. n=26 against n=77 for the *ideal joints model*) as well as the reduction in the algebraic constraint equations in the model (i.e. m=16 against m=66 for the *ideal joints model*) has a crucial effect on the efficiency of the simulation. For these reasons the computation time associated to the explicit Euler method when solving the EOM of the trailing-arm concept model is only about the 30% of the time required by the linearly implicit Euler method in the computation of the EOM of the *ideal joints model* when they are expressed in a dependent coordinates formulation.

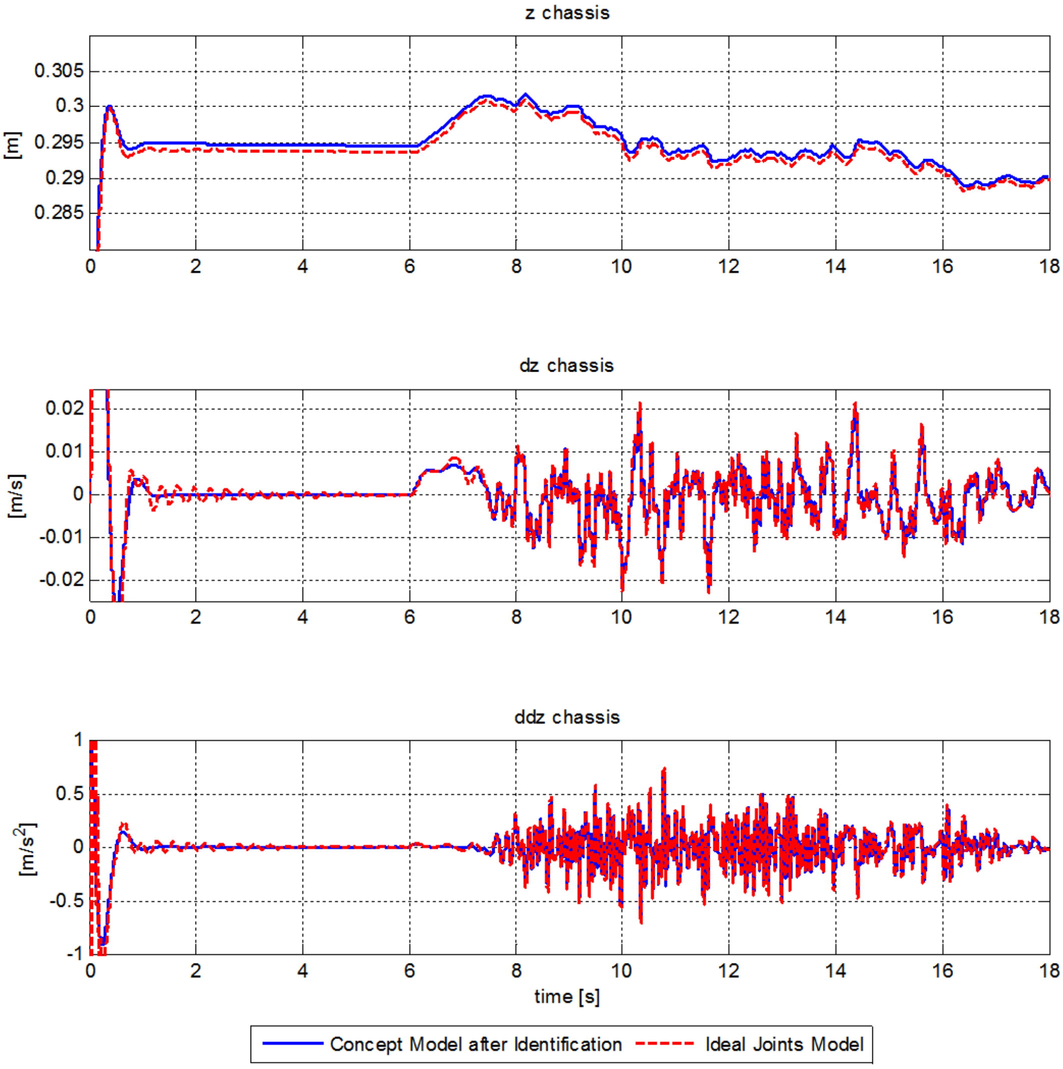


FIGURE 5.11: Validation Manoeuvre 1: dynamic responses

TABLE 5.2: Validation Manoeuvres:
RMS errors between the Concept and the Ideal Joints models responses

	RMS errors Validation I	**RMS errors Validation II**
chassis CG vertical positions	0.0285 $[m]$	0.0276 $[m]$
chassis CG vertical velocities	0.0330 $[m/s]$	0.0546 $[m/s]$
chassis CG vertical accelerations	0.1645 $[m/s^2]$	0.3067 $[m/s^2]$

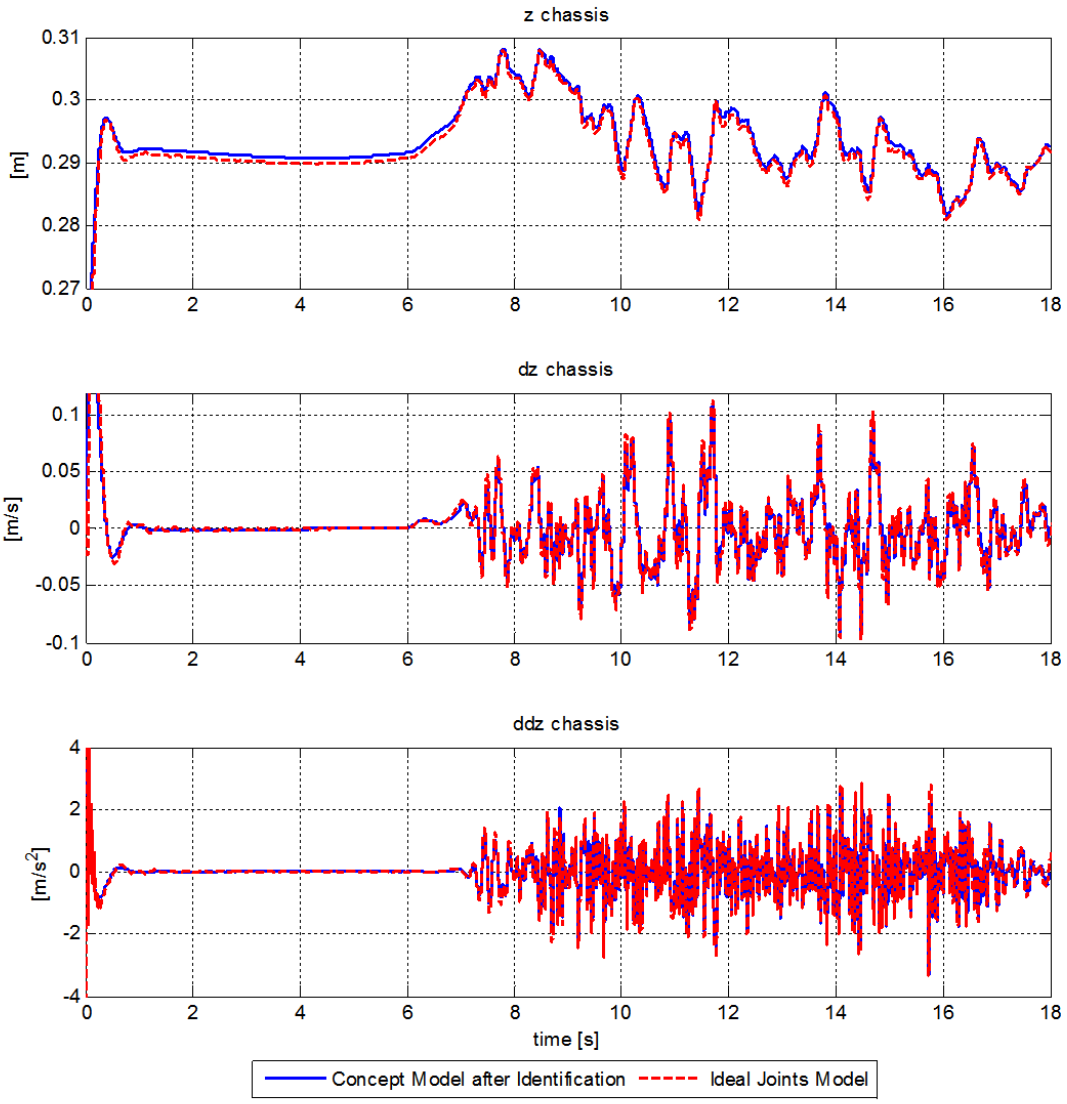


FIGURE 5.12: Validation Manoeuvre 2: dynamic responses

5.2 Concept modelling of the full vehicle

Having assessed the validity of the quarter-car trailing-arm concept model in the previous section the same modelling approach will be extended here to the whole vehicle. A 7 DOFs full-vehicle suspension system is generally used to analyse the ride behaviour of passenger cars with independent suspensions [63] where the unsprung masses are connected to the sprung mass by means of 4 prismatic joints. The bounce, roll and pitch movements of the sprung mass and the vertical displacement of each unsprung mass represent the 7 DOFs retained in this model. The vertical displacements at the 4 tire-road contact points can be given in input to the model in order to analyse the correspondent vertical dynamic response allowing a first assessment and optimization of the global ride properties. The design parameters of the 7 DOFs model are the mass and inertia properties of the sprung and unsprung masses and the stiffness and damping properties of suspensions and tires. However this model is not accurate in predicting the vehicle ride behaviour during longitudinal acceleration and braking manoeuvres.

Indeed, even if the 7 DOFs model is modified in order to permit the analysis of longitudinal acceleration and braking manoeuvres, i.e. enabling the longitudinal motion by adding 4 rotating rims and 4 tire force elements, the prismatic connections are not able to correctly transfer the tire loads from the unsprung masses to the sprung mass.

This is due to the fact that the classical modelling of the unsprung-sprung masses connections via prismatic joints does not contain any information about the actual kinematics of the suspension's linkage which determines how the longitudinal tractive and braking forces developed at the tire-road contact patches are transmitted into the vehicle body.

5.2.1 Full vehicle model with trailing-arm suspension

To overcome the limitations imposed by modelling the unsprung-sprung masses connections by prismatic joints, a full vehicle model with trailing-arm suspensions has been developed as shown in Fig. 5.13. The model has a total of 12 DOFs which are the longitudinal and bounce movements of the sprung mass, its roll and pitch angles plus the 4 relative rotation of the trailing-arms with respect to the sprung mass and the 4 rotations of the rotating rims. This new full vehicle concept model has been proposed with the goal of providing a simple yet reliable model for the combined analysis of the vehicle ride performances and the dynamic phenomena occurring during longitudinal and braking manoeuvres. The main application areas where the proposed 12 DOFs model with trailing-arm suspensions could be employed are:

- **preliminary concept phase in the mechanical design of new vehicle**: at this preliminary stage the concept model could be effectively used to perform preliminary optimizations of the main design parameters such as sprung/unsprung masses and suspensions stiffness and damping properties as well as to optimize the anti-features of the front and rear suspensions in order to meet an initial set of performance attributes.
- **optimization of vertical ride and longitudinal acceleration & braking performances**: given its computational efficiency, the full trailing-arm model could be used in an iterative optimization procedure to tune the damping characteristics of the suspensions and their anti-features during the advanced design phase.
- **design process of active suspension systems**: due to its low complexity level and its ability to correctly capture the main vehicle ride properties in the low

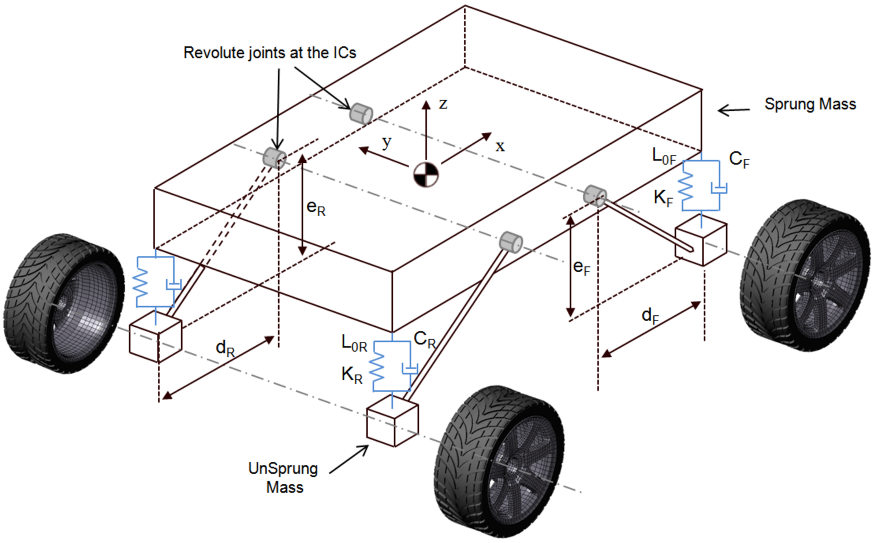


Figure 5.13: 12 DOFs full vehicle model with trailing-arm suspensions

frequency range as well as the pitch and bounce changes occurring during acceleration and braking, the proposed concept model is suitable to be used throughout the whole design process for the tuning of active suspension controllers.

5.2.2 Identification of the full trailing-arm model

Before the proposed full vehicle trailing-arm model can be used to study the vertical ride behaviour and the dynamic response during acceleration and braking, a proper set of design parameters must be assigned. Using the same approach described for the quarter-car trailing-arm model, an identification process will be employed to estimate the initial set of design parameters of the full vehicle concept model starting from a given reference model. However the methodology presented in the previous section will be enhanced here in order to be able to consider as the reference model, not only an available high fidelity multibody vehicle model but also a real vehicle opportunely instrumented (e.g. a predecessor vehicle or a competitor vehicle).

A number of design parameters can be directly measured on the reference model while there are several design parameters which cannot be measured in a straightforward manner. The assessed parameters which can be easily measured on the reference model are:

- sprung mass and its CG position
- Unsprung masses and their CG positions

Indeed the sprung and unsprung masses can be estimated by measuring the weight of the wheels and the vertical load acting on each wheel when the vehicle is in the rest condition. The positions of the wheel centres can also be measured and used to define the positions of the CGs of the unprung masses. Moreover a simple procedure can be applied to estimate the horizontal and vertical CG locations of the total vehicle as well as of the sprung mass as described in [64]. In the present work we assume that tire properties needed for the definition of the concept model are available. If this is not the case typical values of the vertical stiffness, vertical damping and rolling resistance coefficients can be assigned depending on the type of the tires under analysis.

The remaining design parameters i.e. the global suspension properties and the inertia moment of the sprung mass, are not easily obtainable by means of direct measurements on a predecessor model and a proper identification process needs to be set up in order to estimate them. The approach described in section 5.1.3 has to be modified in order to be able to use measured data coming from an objective test on a real vehicle. Indeed the inputs which have been used during the identification process to obtain the response of the quarter-car trailing-arm concept model, i.e. the road profile and the driving and braking torques at the wheels are not easily obtainable from standard measurements during a ride test on a real vehicle. A different strategy is thus required, which must extract the inputs to be applied to the concept model during the identification from available measured data. These inputs, which were the driving and braking torques and the road profile in the case of the identification of the trailing-arm model discussed in section 5.1.3, must reproduce on the concept model the same dynamic conditions experienced by the real vehicle during the test.

A suitable choice is to consider the wheel-centres vertical displacements and the longitudinal acceleration of the vehicle as the input for the concept model during the identification process. Indeed, as it will be described in the next section, these quantities are actually achievable from a measurements campaign performed on a real reference vehicle during a selected test manoeuvre. The vertical displacements of the rim centres can be used to kinematically drive the vertical positions of the unsprung masses in the concept model thus removing the need of modelling the tire's vertical behaviour. Moreover, the longitudinal acceleration of the reference model can be used to determine the traction and braking forces as well as the drive torques which must be applied to the concept model in order to reproduce the test manoeuvre under analysis. The definition of the multibody model of the full trailing-arm during the identification process must thus be changed, in particular the tire force elements are not required any more, since the vertical positions of the wheel centres are directly driven by means of vertical position drivers and because the longitudinal and braking forces are directly applied in input to the model (i.e. they are not generated by the tire force element). The elimination of tire force elements also removes the need for the rotating rim bodies.

As in the case of the quarter-car trailing-arm suspension, the trailing-arm model which wil be used during the identification process has been implemented in Matlab/Simulink. It is composed by 5 bodies, i.e. the sprung mass and the 4 unsprung masses and it is described by a total of n=35 configuration parameters. The unsprung masses are connected to the sprung mass in a trailing-arm configuration by using 4 revolute joints while the vertical positions of the wheel centres are imposed by 4 vertical position drivers. With the addition of the 5 constraint equations imposing the dependency among the Euler parameters related to each body, the total number of constraint equations raises to m=29 and the number of DOFs in the system is f=6 which correspond to the 3 translations and 3 rotations of the chassis. However, since we are not interested in the lateral dynamics we can also constraint the y position and the yaw angle of the chassis ending up with only 4 DOFs.

It will be described now how it is possible to obtain the longitudinal and braking forces to be applied at the tire-road contact patches starting from the measured longitudinal acceleration on the reference model. The traction force acting on each rear wheel during the acceleration phase can be estimated starting from the longitudinal acceleration of the reference model by imposing the dynamic equilibrium along the longitudinal direction on each track of the vehicle. For the rear left wheel the estimated traction force is given by:

$$F_{xRL} = \frac{1}{2} M_{tot} a_x + R_{xRL} + R_{xFL} \tag{5.6}$$

where R_{xRL} and R_{xFL} represent respectively the resistant forces at the rear left and front left wheels computed as:

$$R_x = f_r W_z + \frac{I_w \alpha_w}{r} \tag{5.7}$$

In Eq. 5.7 the parameters f_r and I_w are the rolling resistant coefficient of the tire and the inertia moment of the wheel around its spinning axes respectively while W_z is the vertical load on the wheel. The effective wheel radius r is considered as a constant while the angular acceleration of the wheel α_w can be approximated as the longitudinal acceleration of the vehicle divided by the wheel radius. The associated driving torque is obtained by multiplying the driving force in Eq. 5.6 by the effective wheel radius r:

$$T_{dRL} = r \cdot F_{xRL} \tag{5.8}$$

Since in the present work the full trailing-arm concept model will be used to analyse the dynamic behaviour of a rear-drive vehicle with independent suspensions, this driving torque must be applied from the vehicle body to the rear left trailing-arm as explained in section 5.1.1.

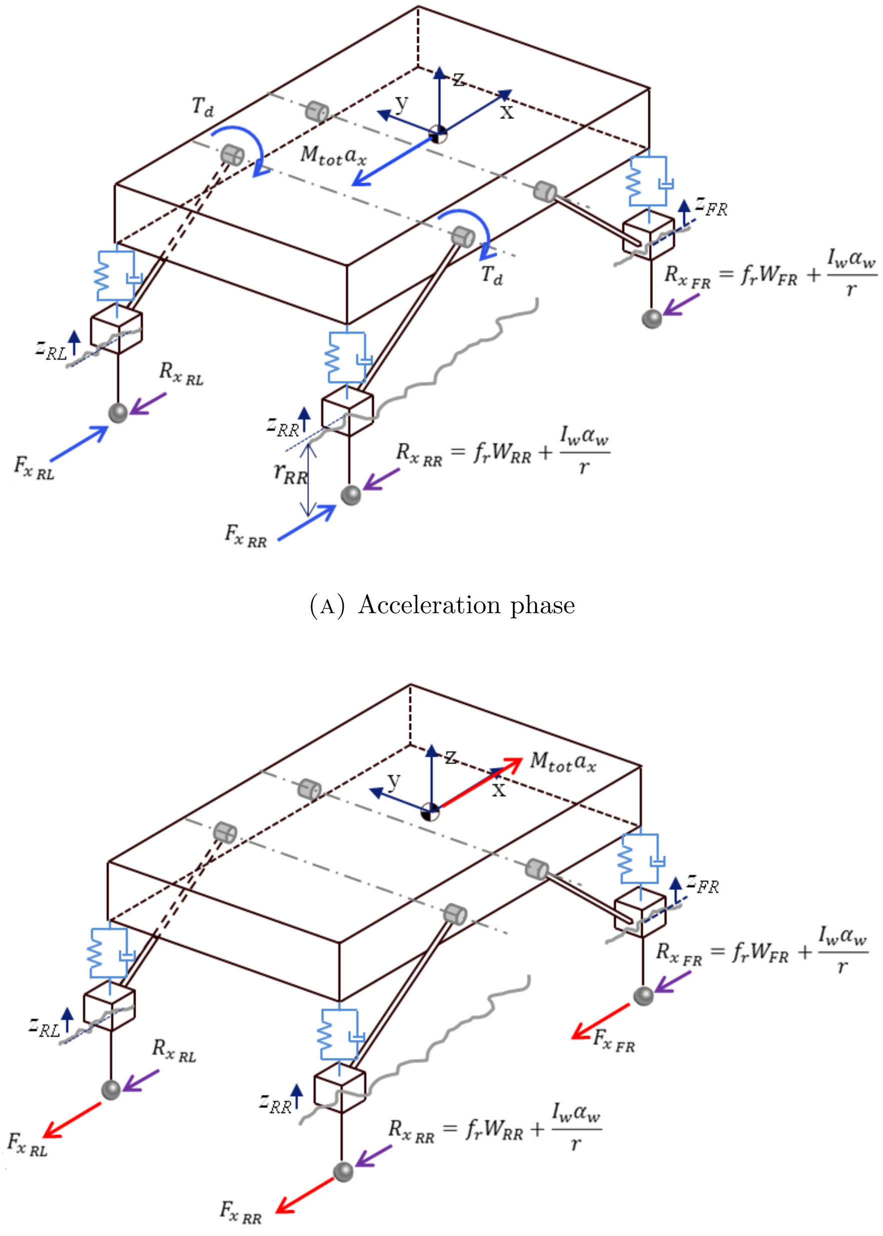
In the case of forward deceleration the braking forces acting on the 4 wheels can also be estimated by imposing the longitudinal dynamic equilibrium for each track of the vehicle provided that the distribution of the braking torques between front and rear axles is known. The braking forces at the rear left and front left wheels are given by:

$$F_{xRL} = \%rear\_braking \cdot (\frac{1}{2}M_{tot}a_x + R_{xRL} + R_{xFL}) \tag{5.9}$$

$$F_{xFL} = \%front\_braking \cdot (\frac{1}{2}M_{tot}a_x + R_{xRL} + R_{xFL}) \tag{5.10}$$

Eqs. 5.7, 5.9 and 5.10 are obtained assuming that the vehicle is running on a road with negligible slope. Moreover the aerodynamic forces acting on the vehicle body are, for sake of simplicity, not taken into account.

The longitudinal forces and the driving torques which will be applied to the multibody concept model during the identification phase are shown in Fig. 5.14 for both the acceleration and the braking phase. By applying this forces and torques on the concept model and by imposing the vertical displacements at the wheel centres it is possible to run a multibody simulation reproducing the same test manoeuvre conducted on the reference model. The resulting dynamic response of the concept model can be then compared to that of the reference model in order to identify the unknown design parameters. In order to be able to compare the dynamic response of the full trailing-arm model with that of a real vehicle, the quantities describing its dynamic response must be actually obtainable from a real measurements campaign. For this reason the suspension's strokes were chosen as the representative dynamic quantities to be compared within the identification algorithm since they are actually measurable by means of wire potentiometers on a real reference model. The scheme of the parameters identification process of the full trailing-arm vehicle is reported in Fig. 5.15. In the figure it is highlighted that the wheel-centres vertical displacements and the longitudinal acceleration measured on the reference vehicle are the quantities given in input to the concept model to reproduce the test scenario. Moreover the dynamic response of the concept model is described by the


T_d
z
y
x
$M_{tot}a_x$
T_d
z_{FR}
$R_{x\,FR} = f_r W_{FR} + \frac{I_w \alpha_w}{r}$
z_{RL}
$R_{x\,RL}$
$F_{x\,RL}$
z_{RR}
r_{RR}
$R_{x\,RR} = f_r W_{RR} + \frac{I_w \alpha_w}{r}$
$F_{x\,RR}$

(A) Acceleration phase

z
$M_{tot}a_x$
y
x
z_{FR}
$R_{x\,FR} = f_r W_{FR} + \frac{I_w \alpha_w}{r}$
$F_{x\,FR}$
z_{RL}
$R_{x\,RL}$
$F_{x\,RL}$
z_{RR}
$R_{x\,RR} = f_r W_{RR} + \frac{I_w \alpha_w}{r}$
$F_{x\,RR}$

(B) Braking phase

Figure 5.14: Full trailing-arm model for the identification process

4 strokes of the suspensions which are given in input to the identification algorithm and compared with the actual ones measured on the reference model.

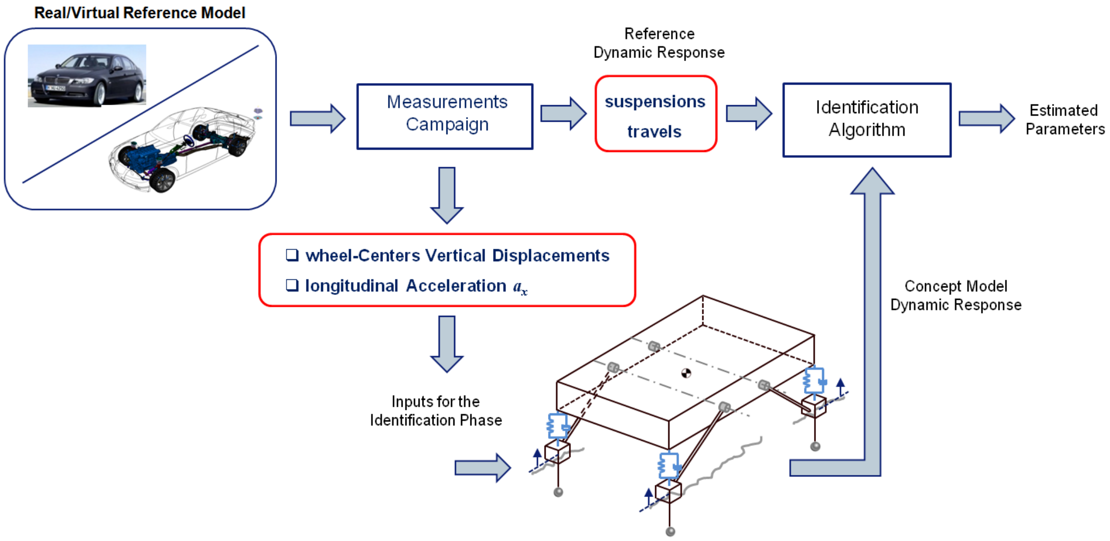

Real/Virtual Reference Model
Measurements Campaign
Reference Dynamic Response
suspensions travels
Identification Algorithm
Estimated Parameters
❑ wheel-Centers Vertical Displacements
❑ longitudinal Acceleration a_x
Inputs for the Identification Phase
Concept Model Dynamic Response

FIGURE 5.15: Parameters identification process

5.2.3 Virtual Measurements Campaign

In order to identify the design parameters needed for the definition of the concept model presented in the previous section a virtual measurements campaign has been carried on a detailed model of a passenger car defined in the multibody environment of Virtual.Lab Motion. The vehicle under analysis is a rear-wheel drive model mounting McPherson and multilink suspensions respectively at the front and rear axles. Geometric and inertia data such as the positions of the suspension's hard points and the mass and inertia properties of the various components have been derived from the industrial FE model of the full vehicle provided by a car manufacturer [65] and used for the definition of the model shown in in Fig. 5.16. Non-linear stiffness and damping properties of bushing connections are taken into account in the model as well as the non-linear characteristics of the shock absorbers which are provided with bump and rebound stops. A rack and pinion connection is used to model the steering system. Stabilizer bars are inserted on both the front and rear axles by means of concentrated rotational springs. The exhaust pipe elements are also included in order to provide the model with its complete mass and inertia properties. The drive torque is transmitted from the differential to the rear wheels by means of the half shafts while the braking torques are applied from the knuckles to the rim in order to simulate outboard brakes. The detailed multibody model is composed by a total of 81 bodies having 164 degrees of freedom. Tires behaviour is taken into account by means of the MF-Swift tire model by TNO Delft setted to 2D contact [59].

The high fidelity multibody model was employed to reproduce an experimental ride test in order to obtain the set of virtual measured quantities which will be used to define the

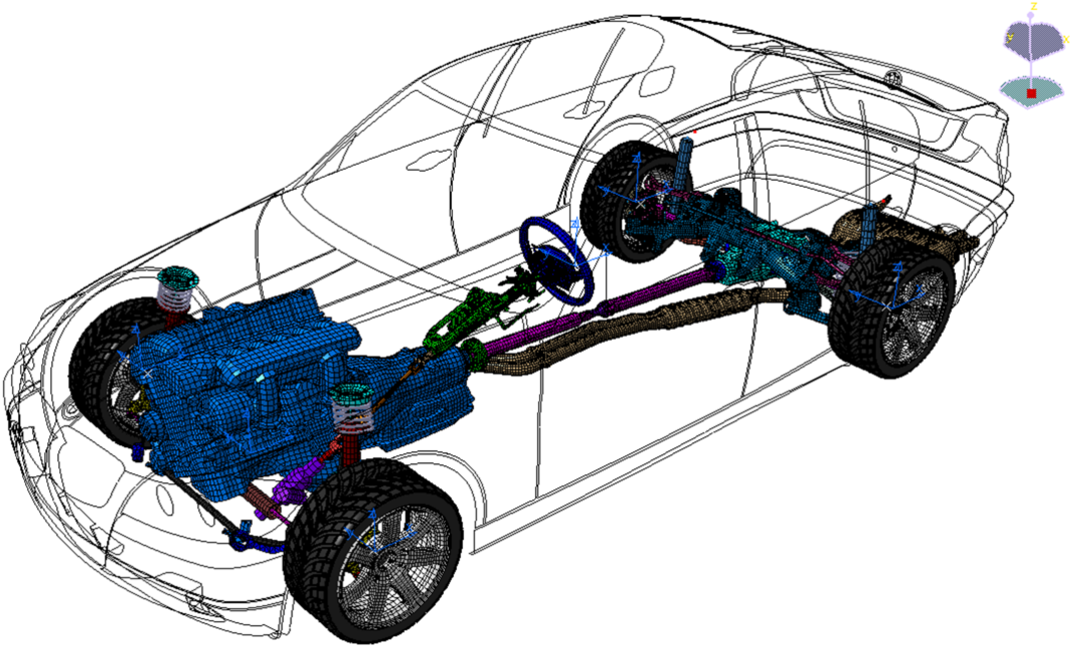


FIGURE 5.16: Full car detailed multibody model

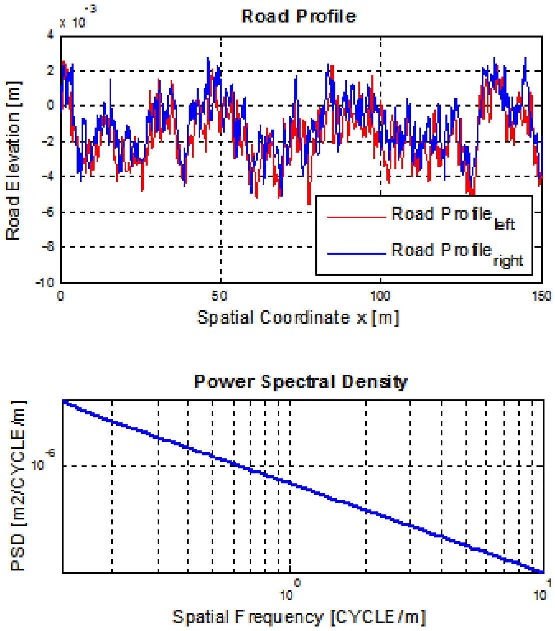
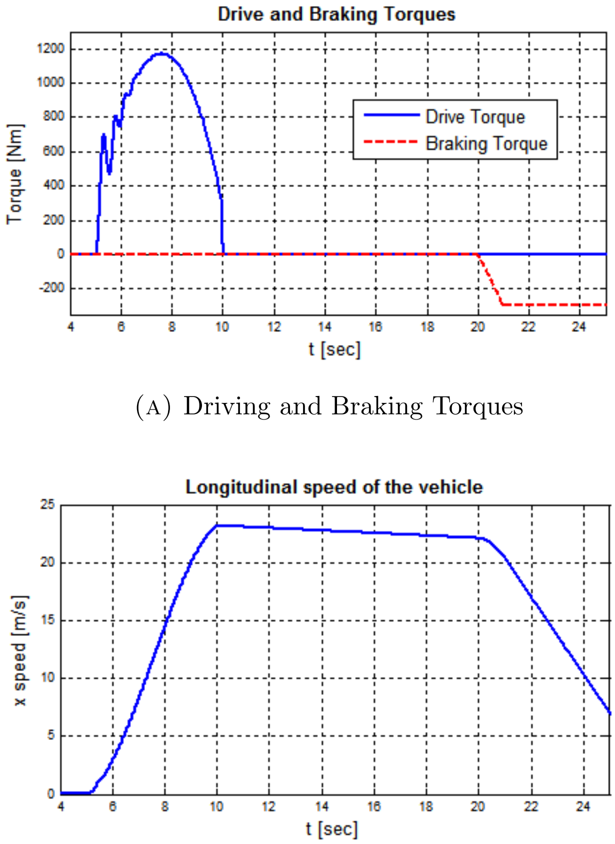


FIGURE 5.17: Portion of the road profile and its ISO 8608 classification by PSD

inputs of the full trailing-arm concept model during the parameter identification process. A proper manoeuvre has been selected in order to highlight both the ride properties of the vehicle and the effects of the vertical load transfer between front and rear axles which occurs during acceleration and braking. The vehicle model is settled on a rough road whose vertical profile has been obtained starting from the power spectral density related to a B-type road according to the ISO classification [62]. The B road left and right profiles, which are implemented in Virtual.Lab Motion by means of spline curves representing the road elevation at the tire contact patches, are depicted in Fig. 5.16 together with the corresponding PSD (Power Spectral Density).

From the rest condition, the steering wheel being kept locked, the vehicle is accelerated to the speed of about 80 Km/h in 5 seconds in order to point out the anti-squat properties of the rear suspensions. After the acceleration phase, the drive torque is removed simulating the engaging of the neutral gear and for 10 seconds the vehicle is let run on the rough road. A braking torque, equally distributed between front and rear axles, is then applied to the four wheels causing a sudden decrease of the vehicle speed. During the braking phase the anti-dive and anti-lift properties respectively of the front and rear suspensions can be appreciated. The drive torque at the rear wheels and the braking torque applied at the 4 wheels are reported in Fig. 5.18a together with the resulting longitudinal speed of the vehicle in Fig.5.18b. After the acceleration phase and the removal of the drive torque, the longitudinal deceleration of the vehicle reaches a typical value of about 0.01 g due to the rolling resistance forces at the four wheels [66].

During the simulation of the reference test manoeuvre several quantities have been virtually measured on the detailed full vehicle model. The extension/compression movements of each suspension have been first sensed. On real life measurements by mounting a wire potentiometer between the vehicle body and one of the suspension's links, it is possible to measure the variations in the suspension travel. In order to reproduce this measurement virtually, a distance sensor has been defined in the detailed vehicle multibody model to measure the stroke of each suspension during the test manoeuvre. The virtual quantities measured by the distance sensors have been filtered above 25 Hz with the aim of considering the same frequency content of signals coming from real instruments during an experimental test. Fig. 5.19 shows the virtual measured stroke of the front left suspension obtained in output from the test manoeuvre. During the acceleration phase (5-10 sec) and the braking phase (20-25 sec) the front suspension undergoes respectively an extension and a compression (brake-dive) due to the vertical load transfer between front and the rear axle. An opposed behaviour can be found if the deflections of one of the rear suspensions are considered.



(A) Driving and Braking Torques

(B) Longitudinal speed

FIGURE 5.18: Test scenario: input torques and resulting longitudinal speed

While rim center vertical displacements can be easily obtained in output from a multi-body simulation, they are not directly measurable during a real experimental campaign. For this reason these quantities, which are needed as input for the dynamic simulations during the identification process, must be reconstructed starting from quantities which are actually measurable in a real ride test. The method here proposed makes use of the vertical accelerations measured at the four corners of the vehicle body in addition

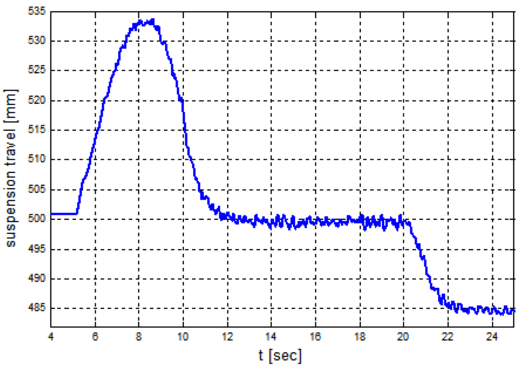


FIGURE 5.19: Front left suspension stroke

to the measures of the suspensions strokes in order to reconstruct the vertical absolute displacements of the four wheel centres. The proposed method can be better understood representing each suspension as a quarter-car model as reported in Fig. 5.2. The vertical acceleration at the body corner ddz_1 and the suspension deflections z_1-z_2 can be used to estimate the vertical displacements of the body corner z_1. By subtracting the suspension deflections from z_1 it is possible to obtain the vertical displacements of the unsprung mass z_2.

Four virtual sensors have been thus placed at the four corners of the vehicle body in order to measure the vertical accelerations in these locations. The low frequencies of the virtual acceleration signals have been filtered below 0.5 Hz in order to reproduce what is commonly done in the post processing of real accelerometers signals. Indeed, due to measurements noise, the low frequency content of the measured accelerations must be filtered and no information are generally available about the accelerations in the low frequency range. Fig. 5.20 shows the filtered vertical acceleration of the body at the FL (Front Left) corner measured during the test manoeuvre.

The obtained acceleration signals can be integrated two times to obtain the vertical displacements at the four body corners. However, since the low-frequency content of the vertical accelerations has been filtered, it is not possible to recover the low-frequency vertical movements at the four corners of the body by simply double integrating the acceleration signals. This is shown in Fig. 5.21 where the vertical position of the FL and RL (Rear Left) corners, obtained by double integrating the corresponding vertical accelerations, are compared to their actual values directly extracted in output from the multibody simulation. Information needed to reconstruct the low-frequency movements of the body corners can be extracted from the displacements signals measured by the virtual wire potentiometers. The changes in the suspension travels in the low frequency range (below 0.5 Hz) mainly result from the combination of the vertical movements of the body-corners and the variations in the wheels radius due to the vertical load transfer between front and rear axle. By neglecting these low frequency tires deflections, the

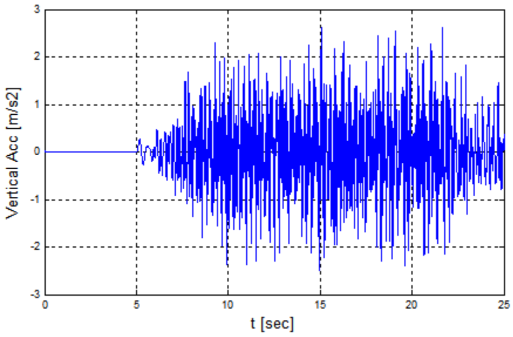


FIGURE 5.20: Body vertical acceleration at the FL corner

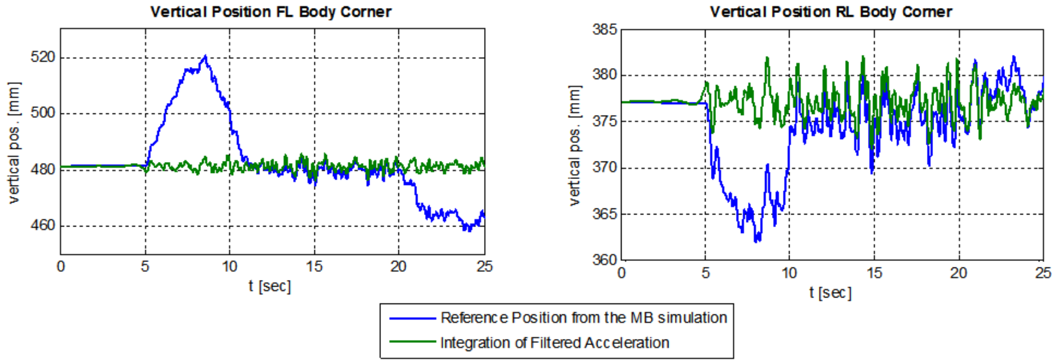


FIGURE 5.21: Chassis corners vertical displacements: step 1

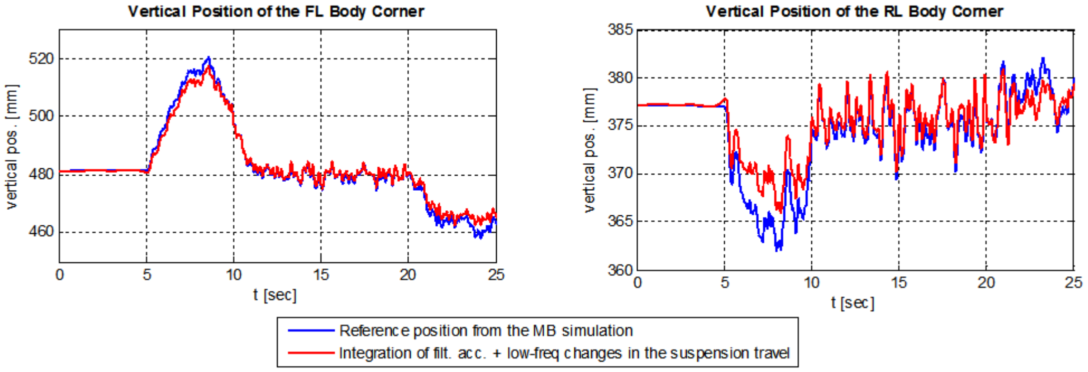


FIGURE 5.22: Chassis corners vertical displacements: step 2

global vertical position of the body corners can be reconstructed by summing the low frequency variations in the suspensions strokes to the vertical displacements obtained by double integrating the vertical accelerations at the body corners. As shown in Fig. 5.22 this leads to a better estimate of the low frequency vertical displacements of the chassis corners during acceleration and braking.

However by analysing Fig. 5.22 it is possible to notice that, since the tire deflections in the low frequency range have been neglected, the vertical position of the FL corner is slightly underestimated during the acceleration phase and, on the contrary, it is overestimated during the braking phase if compared to the reference. The opposite behaviour can be observed if the vertical position at the RL corner is taken into account. Tires deflections during the acceleration and braking phase can be assessed by means of the simple model shown in Fig. 5.23 once the longitudinal acceleration of the vehicle is known. This latter can be obtained by differentiating the longitudinal velocity signal coming from the tachometer during an experimental test.

For the definition of the model based observer 4 parameters are required which are the total weight of the vehicle W_{tot}, the height of the center of gravity h_{CG}, the wheelbase L and the vertical stiffness of the tires K_{tire}. Typical values of these parameters can be

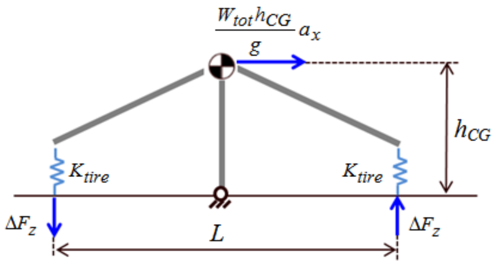


FIGURE 5.23: Model for tire deflections computation during acceleration & braking

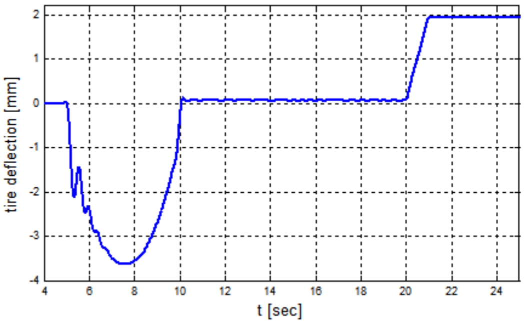


FIGURE 5.24: FL tire deflection due to the longitudinal transfer of vertical load

assigned depending on the class of the vehicle under analysis in case their actual values are unknown. By imposing the equilibrium for the moments it is possible to estimate the change in the vertical loads on the front and rear tires ΔF_z due to the longitudinal acceleration a_x as:

$$\Delta F_z = \frac{W_{tot} \cdot h_{CG}}{g \cdot L} \cdot a_x \tag{5.11}$$

Tire vertical deflections can then be obtained dividing ΔF_z by the tire stiffness. Fig. 5.24 shows the estimated deflections of the front left tire during the test manoeuvre. For the duration of the acceleration phase (5-10 sec) the vertical load on the front wheels decreases and the vertical deflection of the tire assumes negative values corresponding to an increase in the tire radius. The opposite phenomenon occurs during longitudinal deceleration (20-25 sec). The vertical deflections of the rear tires due to the longitudinal transfer of the vertical load are the same but with opposite sign. The obtained tire deflections can be used to correct the estimated vertical positions at the four corners of the body as shown in Fig. 5.25.

As previously anticipated, the vertical displacements at the wheel centres can then be computed by subtracting the suspension deflections from the global vertical positions of the body corners. In Fig. 5.26 the wheel centres vertical displacements of the FL and the RL wheels estimated using the proposed methodology are compared with the

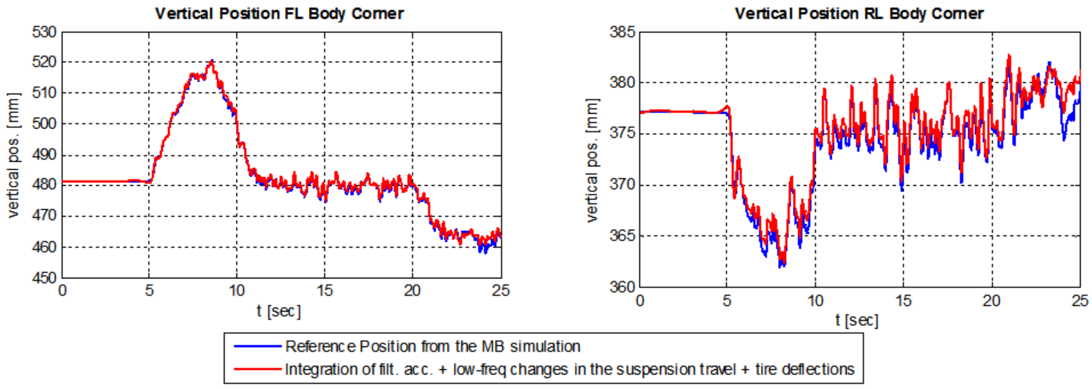


FIGURE 5.25: Chassis corners vertical displacements: step 3

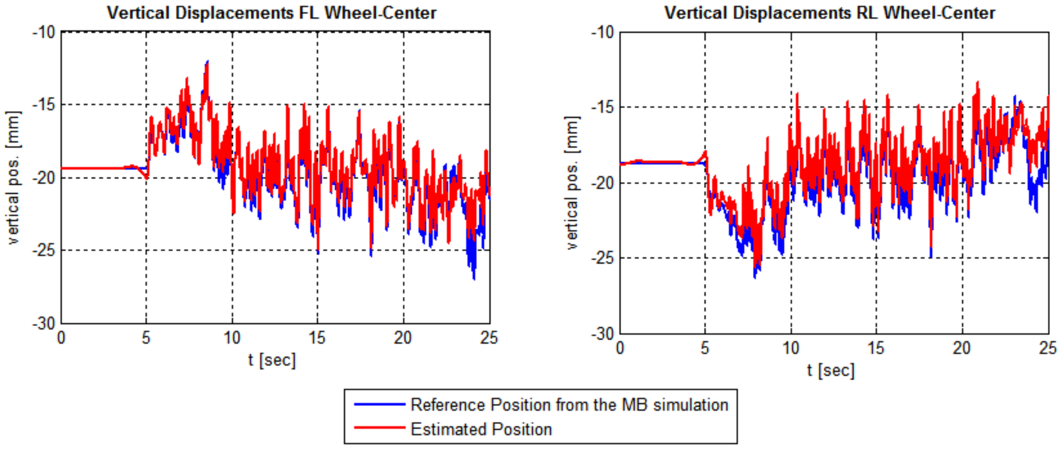


FIGURE 5.26: Estimated vertical displacements of the wheel centres

actual vertical displacements coming from the multibody simulation. The good matching between the estimated displacements and the numeric reference values confirms the validity of the proposed approach. In Fig. 5.26 it is also possible to appreciate the increase and the decrease in the front wheel radius respectively during the acceleration and the braking phase due to the longitudinal transfer of vertical load between the two axles.

5.2.4 Identification: Results and Validation

Once the wheel-centre vertical displacements have been obtained, together with the longitudinal acceleration of the vehicle and the suspension strokes, it is possible to start the parameter identification process in Matlab by using the 'lsqnonlin' function. The numerical identification approach is exactly the same as the one described in section 5.1.3. In this case a function called 'Concept_Vehicle_fun' has been implemented and given in input to the lsqnonlin routine. As shown in Fig. 5.27, this function receives in input the current set of design parameters **b** and performs the integration of the

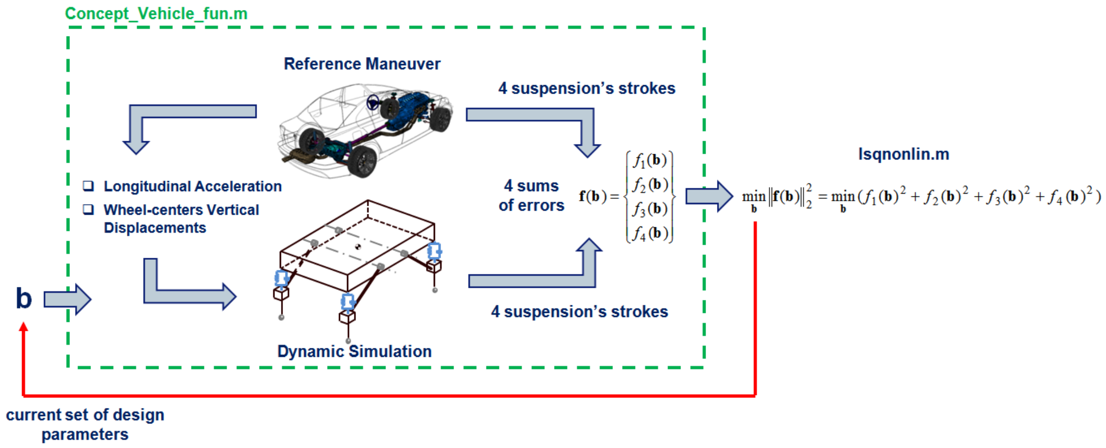

Concept_Vehicle_fun.m
Reference Maneuver
4 suspension's strokes
❑ Longitudinal Acceleration
❑ Wheel-centers Vertical Displacements
4 sums of errors
$\mathbf{f}(\mathbf{b}) = \begin{Bmatrix} f_1(\mathbf{b}) \\ f_2(\mathbf{b}) \\ f_3(\mathbf{b}) \\ f_4(\mathbf{b}) \end{Bmatrix}$
lsqnonlin.m
$\min_{\mathbf{b}} \|\mathbf{f}(\mathbf{b})\|_2^2 = \min_{\mathbf{b}} (f_1(\mathbf{b})^2 + f_2(\mathbf{b})^2 + f_3(\mathbf{b})^2 + f_4(\mathbf{b})^2)$
b
4 suspension's strokes
Dynamic Simulation
current set of design parameters

FIGURE 5.27: Numerical identification algorithm in Matlab

EOM of the concept model considering in input the longitudinal acceleration and the wheel-centres vertical displacements coming from the measurements phase. The multibody simulations within the iterative identification process have been carried out using the explicit Euler method in conjunction with the non-iterative projection step for the stabilization of the constraint equations as described in Chapter 3. The 4 suspension's travel obtained during the dynamic simulation are compared with those coming from the test manoeuvre in order to obtain 4 cumulative errors which constitute the input for the lsqnonlin function in Matlab. After 21 iterations the identification algorithms returns the vector of identified parameters which minimizes the cost function. The 14 unknown parameters estimated during the identification process are:

- the inertia moments of the unsprung mass I_{xx} and I_{yy};
- the 4 suspensions pre-loads (i.e. the springs free lengths) L_{0RL}, L_{0RR}, L_{0FL} and L_{0FR};
- the stiffness and damping properties of the front and rear suspensions K_F, K_R, C_F and C_R;
- the ICs locations in the side view plane of the front and rear suspensions d_F, e_F, d_R and e_R;

The set of identified parameters can be used at this point to define the full trailing-arm concept model shown in Fig. 5.13 which has been implemented in VL.Motion. By assigning a road profile and the driving and braking torques at the four wheels this model can be used to study the ride performances and the handling behaviour of the vehicle during acceleration and braking. In particular, once the parameters have been identified and loaded into the full trailing-arm concept model in VL.Motion, the same direct dynamic analysis can be performed on both the concept and the high-fidelity

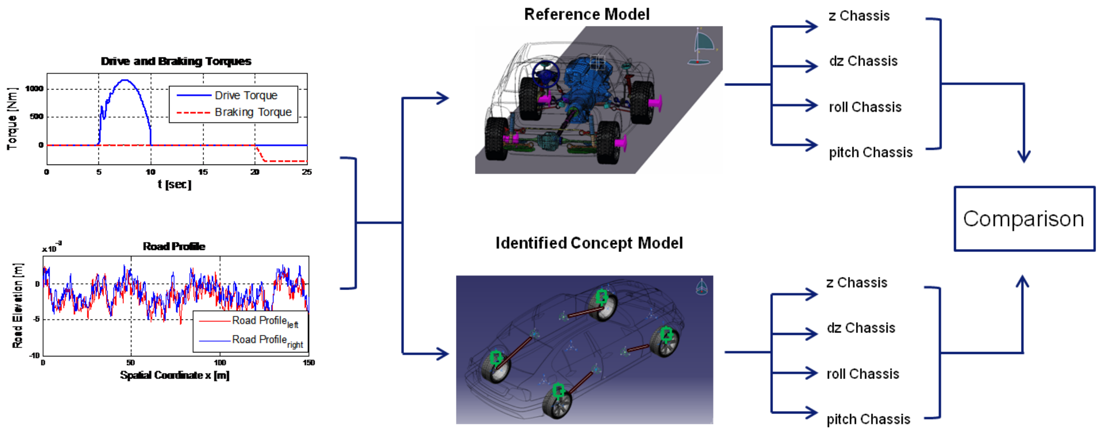


FIGURE 5.28: Validation process

TABLE 5.3: Validation manoeuvres

	Manoeuvre I	**Manoeuvre II**	**Manoeuvre III**
Acceleration Phase 5-10 sec	0-80 Km/h	0-80 Km/h	0-54 Km/h in 5 sec
Braking Phase 20-25 sec	200 Nm all wheels	300 Nm all wheels	310 Nm Front 200 Nm Rear
Road Type (ISO Classification)	A	C	A/B

model by specifying in input the same road profile and the same driving and braking torques. The obtained responses can then be compared to test the effectiveness of the identification process as described in Fig. 5.28.

Three validation manoeuvres have been performed to reproduce 3 different driving scenarios as specified in Table 5.3. In the first manoeuvre the vehicle runs on a A-type road and it undergoes an acceleration from 0 to 80 Km/h in 5 seconds while during the braking phase a braking torque of 200 Nm is applied at each wheel. Fig. 5.29 shows a comparison between the concept model and the high-fidelity model in terms of vertical displacement and vertical velocity at the CG of the chassis, and in terms of pitch and roll angle of the chassis. Results show that the concept model is able to accurately predict the dynamic response of the reference high fidelity multibody model thus confirming the validity of the proposed identification process. In particular the bounce and pitch movements of the vehicle body during the acceleration phase (5-10 sec) and the braking phase (20-25) are correctly reproduced by the concept model proving that the trailing-arm representation of the suspensions is effective in reproducing the dynamic phenomena occurring during acceleration and braking manoeuvres. In the second manoeuvre the vehicle undergoes again an acceleration from 0 to 80 Km/h in 5 seconds while during the braking phase a braking torque of 300 Nm is applied to each wheel.

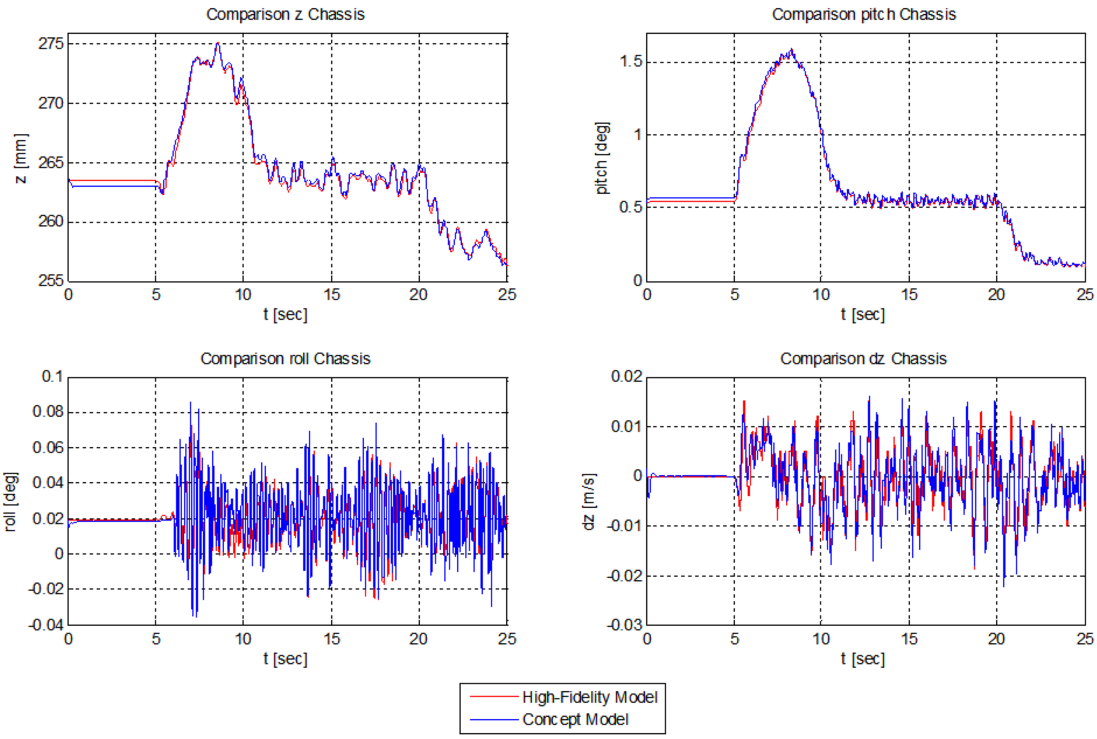


FIGURE 5.29: Results validation manoeuvre I

During the manoeuvre the vehicle runs on a C-type road. Also in this case the identified concept model is able to correctly reproduce both the low frequency pitch variations due to the longitudinal acceleration/braking and the vertical ride behaviour of the suspensions. Finally the same level of accuracy has been obtained in the third manoeuvre where the vehicle starts moving on a A-type road and then incurs in a B-type road. In this manoeuvre a different braking distribution has been considered, in particular during the braking phase a braking torque of 310 and 200 Nm has been applied respectively to the front and rear wheels. The identified model proves to be accurate also in this case.

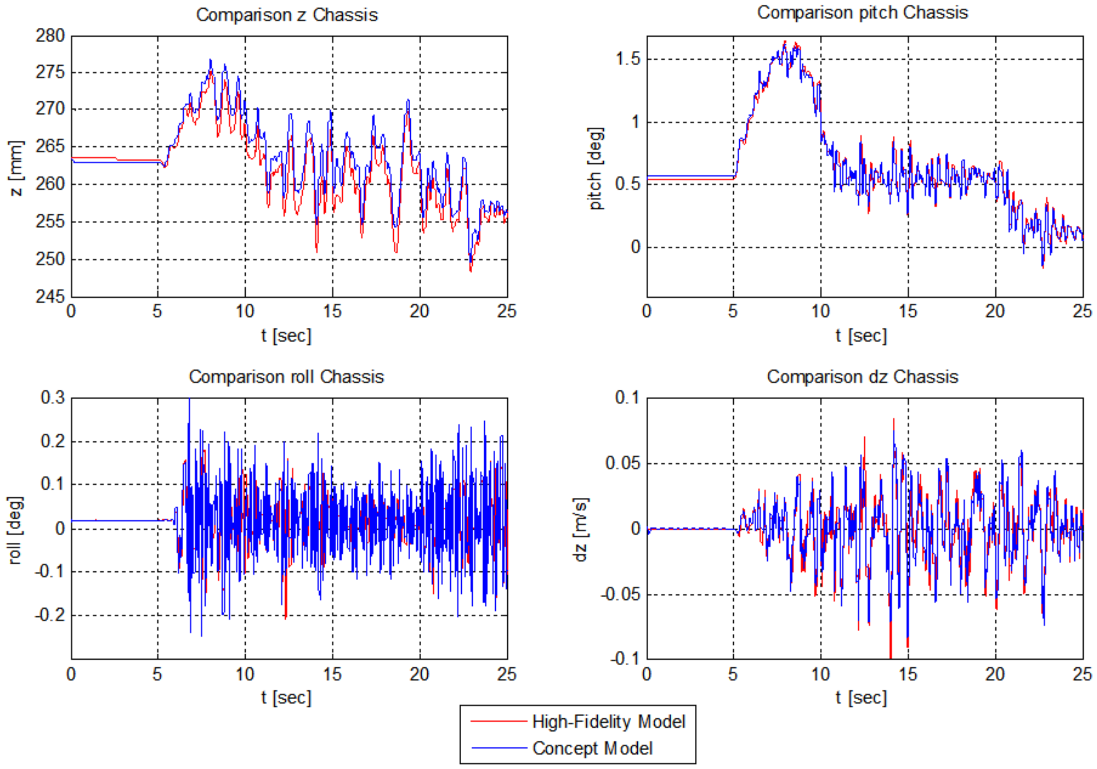


FIGURE 5.30: Results validation manoeuvre II

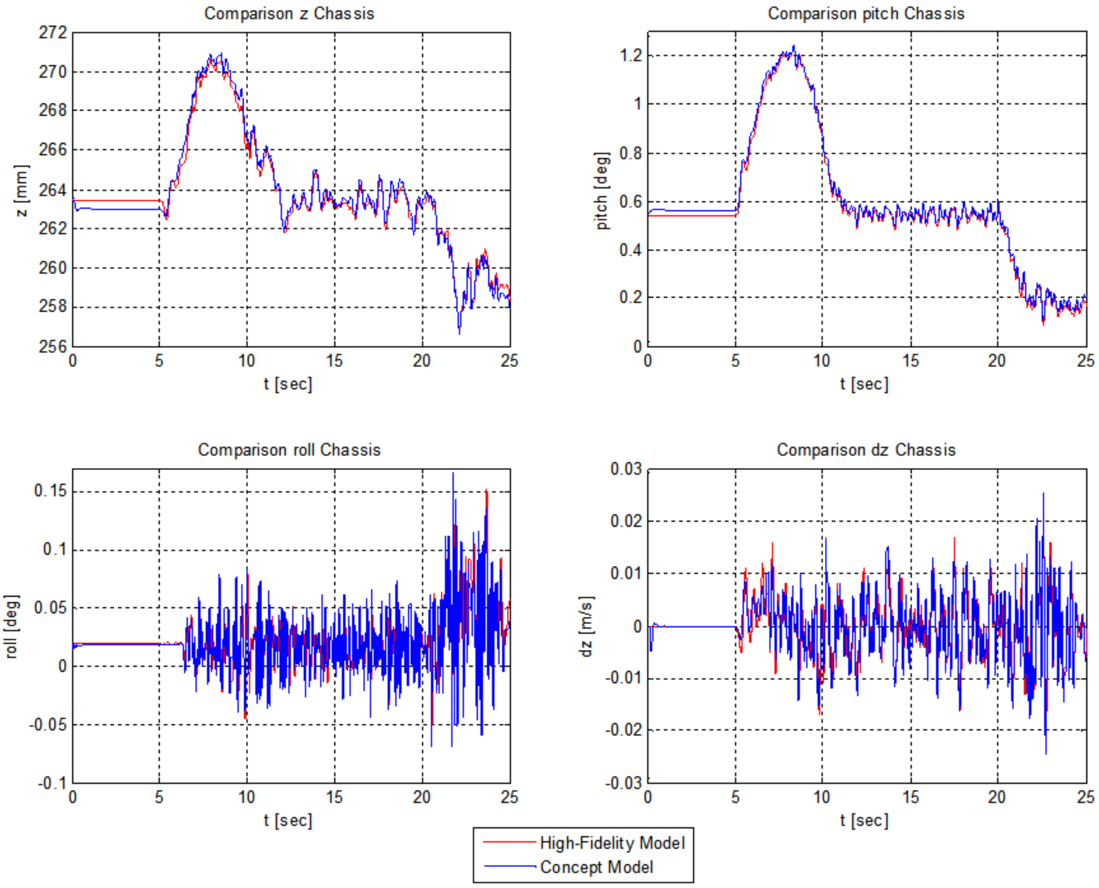


FIGURE 5.31: Results validation manoeuvre III

5.3 Summary

This Chapter has been dedicated to the development of a model reduction process which allows to enclose the characteristics of a detailed suspension system within a simplified trailing-arm concept model. The trailing-arm representation of the suspension has been adopted to include the anti-lift, anti-dive and anti-squat properties of the suspension which influence the pitch and vertical movements of the chassis during longitudinal acceleration and braking manoeuvres. A parameters identification process has been also proposed which allows to identify the design parameters of the concept model in order to reproduce the same dynamic responses measured on the reference model during a selected test manoeuvre.

The proposed approach has been first presented by considering a quarter-car suspension problem. The detailed multibody model of the rear multi-link suspension analysed in Chapter 4 has been selected as the reference model to be mapped into an equivalent trailing-arm concept model. A test scenario has been simulated, in order to highlight the vertical ride properties and the anti-features of the reference rear multi-link suspension model. During the identification process, which has been carried out in Matlab, the same test scenario has been simulated on the quarter-car trailing-arm concept model, and its design parameters have been automatically tuned in order to obtain the same ride and anti properties of the reference model. The effectiveness of the identified quarter-car trailing-arm concept model has been validated in two different driving scenarios showing that the ride characteristics and the anti-squat and anti-lift properties of the reference model are accurately reproduced.

The reduction process has been extended to the full vehicle in the second part of the Chapter by introducing a 12 DOFs full trailing-arm concept model. The correspondent parameters identification process has been enhanced in order to be able to consider a real car as the target vehicle. For this reason all the quantities needed to reproduce the test manoeuvre scenario on the full trailing-arm concept model and to compare its dynamic response against the reference one are derived from physical quantities which are actually measurable during a real experimental test, i.e. the vertical accelerations at the chassis corners, the suspensions travels and the longitudinal acceleration of the chassis. The identified 12 DOFs full trailing-arm concept model has been also validated in several tests scenario showing its ability to capture the vertical ride properties and the pitch and bounce movements of the chassis during longitudinal acceleration/braking manoeuvres.

The employment of the proposed trailing-arm models is suitable at the early stage of the design process when detailed geometrical and component data are not yet available, in order to perform preliminary optimization studies on the global properties of the vehicle such as sprung and unsprung masses and suspensions rates/damping/anti-features properties. Moreover, due to their low complexity level and their computational efficiency, they can be used in iterative processes requiring a high number of simulations, such as the optimization of damping and anti-feature properties of the suspensions. Finally they could be a useful tool for control engineers in order to set up and tune the control logics of active suspension systems for the enhancement of ride performances and for the control of chassis pitch and bounce movements during acceleration and braking.

The proposed identification process can be used at the beginning of the concept design phase of new projects, in order to obtain a starting set of design parameters from a virtual/real predecessor vehicle. Moreover it can be employed to obtain the design parameters of competitors vehicles for benchmarking analysis.

Since it allows a straightforward mapping of the characteristics of detailed multibody suspension models into equivalent trailing-arm concepts models, the proposed identification strategy represents also a useful tool to link the mechanical and the control sides of the design process. Indeed the components modifications required to obtain the desired performances from detailed suspension multibody models, i.e. changes in the hard points locations, components geometries, spring rates, damping properties etc., can be automatically reflected into the simplified trailing-arm models used to create and tune the control logics of active suspension systems thus leading to an integrated vehicle mechatronic design.

Chapter 6

Conclusions and Outlooks

In this research work an investigation has been conducted on the improvements in the efficiency of the LI Euler method coming from an independent coordinates representation of the EOM associated to mechanical systems containing kinematic closed loops with ideal joints and stiff force elements. The automatic transformation of the EOM from a dependent to an independent coordinates formulation at each time step relies on the method based on the matrix **R** whose columns represent a basis of the nullspace of the constraint Jacobian matrix. Proper stabilization of the constraint equations has been carried out by means of a non-iterative projection method which guarantees the drift off effect to remain bounded for arbitrarily long simulations. The obtained numerical results show that the proposed implementation of the LI Euler method may enhance the efficiency of the integration process when dealing with highly constrained multibody systems described by a considerable number of configuration parameters. The accuracy level of the numerical solution has been proved to be equivalent to that achieved with the classical implementation of the LI Euler method. Therefore the proposed algorithm seems particularly attractive for RT automotive applications where an automated switching from a dependent to an independent coordinates representation of the EOM translates in a considerable reduction in the dimensions of the linear system to be inverted at each time step in order to proceed with the integration process. Future research works could quantify the benefits, in terms of integration performances, of the proposed implementation of the LI Euler method when dealing with large multibody systems, such as detailed models of the full vehicle composed by a high number of bodies. Moreover the proposed algorithm could be compiled in a low-level language such as C or Fortran, which are commonly use to set up RT simulations due to their efficiency. This could allow to study the performances of the proposed algorithm in terms of effective turnaround time, allowing a better understanding of its performances and a more effective comparison against the standard implementation of the LI Euler method.

In the final part of the manuscript a model reduction process has been also proposed which exploits a trailing-arm representation of the suspension in order to correctly reproduce both the ride and the anti-dive/lift/squat properties of a reference suspension system. Both a quarter-car and a full vehicle models with trailing arm suspensions have been presented. An identification process has been also set up in order to embed the dynamic characteristics of selected reference suspension systems into the design parameters of the trailing-arm concept models. In particular the identification process has been first carried out using a detailed quarter-car multibody model of a rear multilink suspension as the reference model. The procedure has been then enhanced in order to be able to identify the design parameters of a full trailing arm model using data coming from an objective test conducted on a real target vehicle. Due to their low complexity level, the identified trailing arm models are suitable to be employed in iterative optimization tasks or for the design and tuning of active suspension controllers monitoring the ride properties as well as the pitch and bounce responses of the chassis during straight acceleration and braking manoeuvres. They could also be employed in the early stages of the design phase, when detailed geometrical and components data are not yet available, in order to perform preliminary benchmarking analysis on the main vehicle design parameters such as sprung and unsprung masses, suspensions damping and stiffness and anti-dive/lift/squat properties. A fundamental enhancement of the proposed model reduction approach, which could be addressed by future research works, is the extension of the capability of the trailing-arm models in order to be able to analyse also the lateral dynamics of the vehicle. To achieve this goal the IC position of the suspension in the front view plane should be also considered by reorienting the axis of the revolute joints connecting the unsprung masses to the sprung mass, i.e. by proper positioning of the instant rotation axis of the suspension. Also the contribution of the anti-roll bars to the total roll stiffness of the vehicle should be taken into account and, finally, a proper simplified representation of the steering system should be employed.

Bibliography

[1] http://www.fkfs.de/english/automotive-mechatronics/leistungen/driving-simulators/

[2] Cuadrado, J., Cardenal, J.,Modeling and Solution Methods for Efficient Real-Time Simulation of Multibody Dynamics, Multibody Syst. Dyn. **1**, 259-280 (1997).

[3] Nikravesh, P. E. (2005). An overview of several formulations for multibody dynamics. In Product Engineering (pp. 189-226). Springer Netherlands.

[4] De Cuyper, J., Furmann, M., Kading, D., Gubitosa, M. (2007). Vehicle dynamics with LMS virtual. lab motion. Vehicle System Dynamics, 45(S1), 199-206.

[5] Ryan, R. R. (1990). ADAMS—Multibody system analysis software. In Multibody systems handbook (pp. 361-402). Springer Berlin Heidelberg.

[6] Haug, E. J. (1989). Computer aided kinematics and dynamics of mechanical systems (Vol. 1, pp. 48-104). Boston: Allyn and Bacon.

[7] García de Jalón, J., Bayo, E., Kinematic and Dynamic Simulation of Multibody Systems: The Real-time Challenge, Springer-Verlag, New York (1994).

[8] Shabana A. Dynamics of Multibody Systems, Cambridge 2005.

[9] Bae, D. S., Haug, E. J. (1987). A recursive formulation for constrained mechanical system dynamics: Part i. open loop systems. Journal of Structural Mechanics, 15(3), 359-382.

[10] Kim, S. S., Haug, E. J. (1988). A recursive formulation for flexible multibody dynamics, Part I: open-loop systems. Computer Methods in Applied Mechanics and Engineering, 71(3), 293-314.

[11] Changizi, K., Shabana, A. A. (1988). A recursive formulation for the dynamic analysis of open loop deformable multibody systems. Journal of applied mechanics, 55(3), 687-693.

[12] Nikravesh, P. E., Affifi, H. A. (1994). Construction of the equations of motion for multibody dynamics using point and joint coordinates. In Computer-aided analysis of rigid and flexible mechanical systems (pp. 31-60). Springer Netherlands.

[13] Jalón, D., García, J., Unda, J., Avello, A. (1986). Natural coordinates for the computer analysis of multibody systems. Computer Methods in Applied Mechanics and Engineering, 56(3), 309-327.

[14] Bayo, E., Garcia de Jalon, J., Serna, M. A. (1988). A modified Lagrangian formulation for the dynamic analysis of constrained mechanical systems. Computer methods in applied mechanics and engineering, 71(2), 183-195.

[15] Wehage, R.A. and Haug, E.J., Generalized co-ordinate partitioning for dimension reduction in analysis of constrained dynamic systems, ASME Journal of Mechanical Design 104, 1982, 247–255.

[16] Kane, T.R. and Levinson, D.A., Dynamics: Theory and Applications, McGraw-Hill, New York, 1985.

[17] Serna, M. A., Avilés, R., García de Jalón, J. (1982). Dynamic analysis of plane mechanisms with lower pairs in basic coordinates. Mechanism and Machine Theory, 17(6), 397-403.

[18] Jerkovsky, W. (1978). The structure of multibody dynamics equations. Journal of Guidance, Control, and Dynamics, 1(3), 173-182.

[19] Kim, S. S., Vanderploeg, M. J. (1986). A general and efficient method for dynamic analysis of mechanical systems using velocity transformations. Journal of Mechanical Design, 108(2), 176-182.

[20] Nikravesh, P. E., Gim, G. (1993). Systematic construction of the equations of motion for multibody systems containing closed kinematic loops. Journal of Mechanical Design, 115(1), 143-149.

[21] de Jalón, J. G., Jiménez, J. M., Avello, A., Martín, F., Cuadrado, J. (1991). Real time simulation of complex 3-D multibody systems with realistic graphics. In Real-Time Integration Methods for Mechanical System Simulation (pp. 265-292). Springer Berlin Heidelberg.

[22] Paul, B. (1975). Analytical dynamics of mechanisms — A computer oriented overview. Mechanism and Machine Theory, 10(6), 481-507.

[23] Kim S.S., Jeong W.H., Jung D.H., Choi H.J., HIL-simulation for Evaluation of Intelligent Chassis Controller using Real-time Multibody Vehicle Dynamics Model.

The 1st Joint International Conference on Multibody System Dynamics, May 25-27, 2010, Lappeenranta, Finland.

[24] Ohba, T., Takema, I., Minami, Y., Yokoyama, H. (2002). Application of HIL Simulations for the Development of Vehicle Stability Assist System (No. 2002-01-0816). SAE Technical Paper.

[25] Yan, Q. Z., Thompson, F. C., Paul, R. E., Bielenda, J. J. (2004). Hardware in the Loop for Dynamic Chassis Control Algorithms Test and Validation (No. 2004-01-2059). SAE Technical Paper.

[26] Dahlberg T., An optimised speed controlled suspension of a 2-DOF vehicle travelling on a randomly profiled road, Journal of Sound and Vibration 64 (4) (1979) 541–546.

[27] Wang F., Design and Synthesis of Active and Passive Vehicle Suspensions, PhD Thesis, Queens' College, 2001.

[28] Burgermeister, B., Arnold, M., Eichberger, A., Smooth velocity approximation for constrained systems in real-time simulation, Multibody Syst. Dyn. **26**, 1-14 (2011).

[29] Prescott, B., Heirman, G., Furman, M., De Cuyper, J., Using High-Fidelity Multibody Vehicle Models in Real-Time Simulations, SAE Technical Paper 2012-01-0927, (2012).

[30] Arnold, M., Burgermeister, B., Linearly Implicit Time Integration Methods in Real-Time Applications: DAEs and stiff ODEs, Multibody Syst. Dyn. **17**, 99-107 (2007).

[31] Burgermeister, B., Arnold, M., Esterl, B., DAE time integration for real-time applications in multi-body dynamics, Z. Angew. Math. Mech. **86**, 759-771 (2006).

[32] Weber, S., Arnold, M., Model reduction via quasi static approximations, Proc. Appl. Math. Mech. **10**, 655-656 (2010).

[33] Eichberger, A., Rulka, W., Process Save Reduction by Macro Joint Approach: The Key to Real Time and Efficient Vehicle Simulation, Vehicle Syst. Dyn., **41**, 401-413 (2010).

[34] Rill, G., A modified implicit Euler algorithm for solving vehicle dynamic equations, Multibody Syst. Dyn. **15**, 1-24 (2006).

[35] Schiela, A., Bornemann, F., Sparsing in real time simulation, Z. Angew. Math. Mech. **83**, 637–647 (2003).

[36] Hall A., Uchida T., Loh F., Schmitke C., McPhee J., Reduction of a Vehicle Multibody Dynamic Model using Homotopy Optimization. The Archive of Mechanical Engineering, Vol. LX, Num.1, 2013.

[37] Saglam F., Identification of low order vehicle handling models from multibody vehicle dynamics models. Mechatronics (ICM), IEEE International Conference, 13-15 April 2011, Instanbul.

[38] Serban, R., Freeman, J. S. (2001). Identification and identifiability of unknown parameters in multibody dynamic systems. Multibody System Dynamics, 5(4), 335-350.

[39] Eberhard, P., Piram, U., Bestle, D. (1999). Optimization of damping characteristics in vehicle dynamics. Engineering optimization, 31(4), 435-455.

[40] Thoresson M.J., Uys P.E., Els P.S., Snyman J.A., Efficient optimisation of a vehicle suspension system, using a gradient-based approximation method, Part 1: Mathematical modelling. Mathematical and Computer Modelling 50 (2009) 1421–1436, Elsevier.

[41] Etman L.F.P., Vermeulen R.C.N., van Heck J.G.A.M., Schoofs A.J.G., van Campen D.H., Design of a stroke dependent damper for the front axle suspension of a truck using multibody system dynamics and numerical optimisation, Vehicle System Dynamics 38 (2002) 85–101.

[42] Eichberger, A. (2003). Generating multibody real-time models for hardware-in-the-loop applications. ATA-TORINO-, 56(5/6), 164-167.

[43] Kim, S. and Jung, H., Functinal Suspension Model for Real-Time Vehicle Dynamic Analysis. Proceeding on International Conference on Automotive Engineering, Dec. 10–13, 2001, Koogmin University, Seul Korea.

[44] Rulka, W., Pankiewicz, E. (2005). MBS approach to generate equations of motions for HiL-simulations in vehicle dynamics. Multibody system dynamics, 14(3-4), 367-386.

[45] Hong-Kyu Jung, Sang-Sup Kim, Jeong-Soo Shim, Chae-Won Kim, Development of Vehicle Dynamics Model for Real-time ECU Evaluation System.

[46] Baumgarte, J., Stabilization of constraints and integrals of motion in dynamical systems, Comput. Methods Appl. Mech. Eng. **1**, 1-16 (1972).

[47] Petzold L.R., A description of DASSL: A differential/algebraic system solver. In R.S. Stepleman et al., editors, IMACS Trans. Scient. Comp. Vol. 1 , pages 65-68. North-Holland, Amsterdam, 1983.

[48] Haug E.J., Negrut D., Engstler C., Implicit Runge-Kutta Integration of the Equations of Multibody Dynamics in Descriptor Form. Mechanics Based Design of Structures and Machines - MECH BASED DES STRUCT MECH 01/1999; 27(3):337-364.

[49] Haug, E., J., Iancu, M., Negrut, D., Implicit Integration of the Equations of Multibody Dynamics in Descriptor Form, in Advances in Design Automation-Proceedings of the ASME Design Automation Conference, Sacramento, 1997.

[50] Haug E., J., Negrut, D., Iancu, M., A State-Space Based Implicit Integration Algorithm for Differential-Algebraic Equations of Multibody Dynamics, Mech. Struct.&Mech., vol. 25(3), pp. 311-334, 1997.

[51] Negrut, D., On the Implicit Integration of Differential-Algebraic Equations of Multibody Dynamics, Ph.D. Thesis, The University of Iowa, 1998.

[52] O. Bruls, M.Arnold, The generalized-alpha scheme as a linear multistep integrator: Toward a general mechatronic simulator, Journal Of Computational And Nonlinear Dynamics, 3, 041007.

[53] Hairer, E.,Wanner, G., Solving Ordinary Differential Equations. II. Stiff and Differential-Algebraic Problems, 2nd edn. Springer–Verlag,Berlin Heidelberg New York (1996).

[54] Pennestrì, E., Valentini, P.P., Coordinate Reduction Strategies in Multibody Dynamics: A Review, Conference on Multibody System Dynamics, Pitesti, Romania, ISBN 1582-9561, 25-26 October (2007).

[55] Lubich, Ch., Engstler, Ch.,Nowak, U., Pöhle, U.,title, Numerical integration of constrained mechanical systems using MEXX, Mech. Struct. Mach. **23**, 473-495 (1995).

[56] Pacejka, H. B., Tire and Vehicle Dynamics, 2nd edn. Elsevier, Oxford (2006).

[57] Rulka, W. (1990). SIMPACK—A computer program for simulation of large-motion multibody systems. In Multibody systems handbook (pp. 265-284). Springer Berlin Heidelberg.

[58] Lot, R., Da Lio, M.,A Symbolic Approach for Automatic Generation of the Equations of Motion of Multibody Systems, Multibody Syst. Dyn. **12**, 147-172 (2004).

[59] TNO Automotive: MF-Tyre/MF-Swift 6.1.1 Help Manual, TNO Automotive, The Netherlands, 2009.

[60] C. Kim, P. I. Ro. Reduced-order modeling and parameter estimation for a quarter-car suspension system. Proc. Instn Mech. Engrs, Part D, Journal of Automobile Engineering, 2000, 214 D, 851-864.

[61] C. Kim, P. I. Ro, H. Kim. Effect of suspension structure on equivalent suspension parameters. Proc. Instn Mech. Engrs, Part D, Journal of Automobile Engineering, 1999, 213(D5), 457-470.

[62] ISO 8608. Mechanical vibration - Road surface profiles - Reporting of measured data, First Edition, 1995.

[63] S. Abramov, S. Mannan, O. Durieux. Semi-active suspension system simulation using Simulink. International Journal of Engineering Systems Modelling and Simulation 2009 - Vol.1.

[64] W.F. Milliken, D.L. Milliken, and L.D. Metz. Race car vehicle dynamics. SAE International Warrendale, PA, 1995, pag.666-673.

[65] T. Tamarozzi, G. Stigliano, M. Gubitosa, S. Donders, W. Desmet. Investigating the use of reduction techniques in concept modeling for vehicle body design optimization, ISMA 2010.

[66] Gillespie T D. Fundamentals of Vehicle Dynamics. Society of Automotive Engineers, Inc., 1992.