State observers based on detailed multibody models applied to an automobile

Emilio Sanjurjo

Advisors: Miguel Ángel Naya Villaverde
Javier Cuadrado Aranda

Doctoral thesis

UNIVERSIDADE DA CORUÑA

Programa de Enxeñería Industrial

Ferrol, 2016

Dr. Miguel Ángel Naya Villaverde, Doctor by the University of A Coruña, and Dr. Javier Cuadrado Aranda, Doctor by the University of Navarra, certify that this doctoral dissertation, entitled State observers based on detailed multibody models applied to an automobile, has been developed by Emilio Sanjurjo under their supervision in order to obtain the International Doctor mention by the University of A Coruña.

Dr. Miguel Ángel Naya Villaverde, Doctor por la Universidad de A Coruña, y Dr. Javier Cuadrado Aranda, Doctor por la Universidad Navarra, certifican que la presente memoria, titulada State observers based on detailed multibody models applied to an automobile, ha sido desarrollada por Emilio Sanjurjo bajo su supervisión para optar al grado de Doctor con mención Internacional por la Universidad de A Coruña.

Ferrol, 2016.

Emilio Sanjurjo,	Dr. Miguel Á. Naya Villaverde	Dr. Javier Cuadrado Aranda
PhD. student	Advisor	Advisor
Doctorando	*Director*	*Director*

Agradecimientos

Aunque normalmente una tesis doctoral se ve como un trabajo personal, en realidad el autor suele recibir ayuda de varias personas, y esta tesis no es una excepción.

En primer lugar, me gustaría reconocer el gran trabajo de dirección que han realizado los directores de esta tesis, Javier Cuadrado y Miguel Ángel Naya. También quiero agradecer el buen ambiente de trabajo y la ayuda que me han proporcionado todos mis compañeros del Laboratorio de Ingeniería Mecánica de la Universidad de A Coruña: Florian, David, Amelia, Daniel, Alberto, Lolo, Urbano, Francisco González, Francisco Mouzo, Antonio, y Roland, que aunque ya no está en el laboratorio ha seguido prestándome ayuda. Me gustaría destacar la gran ayuda recibida de Alberto para solucionar problemas informáticos.

A lo largo de todo este tiempo también han ido pasando otros alumnos por el laboratorio que han realizado una gran contribución a esta tesis, especialmente Pasquale Gallo, Alfonso Varela, Alejandro Vázquez y Edoardo Sinigaglia.

También he recibido ayuda de gente de otras universidades, especialmente de mis compañeros de la Universidad de Almería, José Luis Torres y José Luis Blanco, y del supervisor de mi estancia en la Universidad Tecnológica de Delft, Arend Schwab, que contribuyeron enormemente al resultado final de esta tesis.

Por último, pero no menos importante, quiero agradecer todo el apoyo proporcionado por parte de mis padres, Luciano y Maruja, mis hermanos, José y Luco, y dos personas muy especiales para mí, Gon y Fátima.

Acknowledgments

Although a doctoral thesis is often seen as a personal work, actually the author usually receives help from several people, and this thesis is not an exception.

First, I would like to acknowledge the great work done by the advisors of this thesis, Javier Cuadrado and Miguel Ángel Naya. I also want to thank the good working environment and help provided to me by all my colleagues at the Laboratorio de Ingeniería Mecánica of the University of A Coruña: Florian, David, Amelia, Daniel, Alberto, Lolo, Urbano, Francisco González, Francisco Mouzo, Antonio, and Roland, who is no longer at the laboratory, but has still provided support to me. I would like to highlight the great help provided by Alberto to solve computer problems.

Other students have stayed at the laboratory throughout this time, and some of them have made a great contribution to this thesis, especially Pasquale Gallo, Alfonso Varela, Alejandro Vázquez and Edoardo Sinigaglia.

I have also received help from people from other universities, in particular from my colleagues of the University of Almería, José Luis Torres and José Luis Blanco, and the supervisor of my stay at the Delft University of Technology, Arend Schwab. All of them have greatly contributed to the final result of this thesis.

Last but not least, I want to thank all the support provided by my parents, Luciano and Maruja, my brothers, José and Luco, and two very special people for me, Gon and Fátima.

Resumo

Un sistema multicorpo é un conxunto de corpos ríxidos ou flexibles que están unidos a través de pares cinemáticos. A dinámica de sistemas multicorpo é unha disciplina na que se estudan métodos computacionais eficientes para resolver as ecuacións da dinámica destes sistemas. Se o cálculo da dinámica se executa máis rápido que o movemento do mecanismo, a simulación resultante pode ser sincronizada con elementos externos reais, abrindo un abanico de novas posibilidades, como o seu emprego en simuladores, ou en observadores de estados.

Un observador de estados é unha ferramenta matemática na que se combina o modelo dinámico dun sistema con información procedente de sensores para obter máis e/ou mellor información sobre o sistema. Empréganse cando se quere coñecer unha magnitude que non se pode medir directamente por razóns económicas ou técnicas.

Nesta tese, desenvólvense observadores de estados baseados en modelos multicorpo. Próbanse primeiro en simulación con modelos simples de mecanismos planos, estudando a precisión alcanzada e o custo computacional. Finalmente, o método que ten unha mellor relación entre a precisión e custo computacional aplícase ao caso dun vehículo automóbil.

Resumen

Un sistema multicuerpo es un conjunto de sólidos rígidos o flexibles que están unidos por medio de pares cinemáticos. La dinámica de sistemas multicuerpo es una disciplina en la que se estudian métodos computacionales eficientes para resolver las ecuaciones de la dinámica de estos sistemas. Si el cálculo de la dinámica se ejecuta más rápido que el movimiento del mecanismo, la simulación resultante se puede sincronizar con elementos externos reales, abriendo un abanico de nuevas posibilidades, como su empleo en simuladores, o en observadores de estados.

Un observador de estados es una herramienta matemática en la que se combina el modelo dinámico de un sistema con información procedente de sensores para obtener más y/o mejor información acerca del sistema. Se emplean cuando se desea conocer una magnitud que no puede ser medida directamente por motivos económicos o técnicos.

En esta tesis, se desarrollan y prueban observadores de estados basados en modelos multicuerpo. Se prueban primero mediante simulación con modelos sencillos de mecanismos planos, estudiando la precisión que alcanzan y su coste computacional. Finalmente, el método que presenta una mejor relación entre precisión y coste computacional se implementa para el caso de un vehículo automóvil.

Abstract

A multibody system is a set of rigid or flexible bodies that are linked via joints. Multibody dynamics is a discipline in which efficient computational methods are studied to solve the equations of the dynamics of these systems. If the calculation of dynamics runs faster than the motion of the mechanism, the resulting simulation can be synchronized with real external elements, opening a range of new possibilities, such as its use in simulators, or in state observers.

A state observer is a mathematical tool in which the dynamic model of a system is combined with information from sensors to obtain more and/or better information about the system. They are used when the desired magnitude cannot be directly measured due to economical or technical reasons.

In this thesis, multibody-based state observers are developed and tested. They are tested first in simulation with simple models of flat mechanisms, studying their achieved accuracy and computational cost. Finally, the method which has a better relationship between accuracy and computational cost is implemented for an automobile.

Contents

List of Figures

List of Tables

List of Tables

Acronyms

CEKF continuous-time extended Kalman filter. 8, 9, 25, 30, 48–51, 54, 59, 91, 149

DAE differential-algebraic equations. 26

DAP data acquisition processor. 60, 77, 78

DEKF discrete-time extended Kalman filter. 25, 31, 46, 48, 50–52, 54–57, 91

DIEKFpm discrete-time iterated extended Kalman filter with perfect measurements. 35, 91

ECEF Earth-centered Earth-fixed reference system. 62

EKF extended Kalman filter. 36, 38, 51, 55, 56, 86

ENU East-North-Up reference system. 62, 76, 77

errorEKF error-state extended Kalman filter. 25, 36, 44, 48, 50, 51, 54, 55, 59, 84, 85, 91, 92, 148–150

ETRS89 European Terrestrial Reference System 1989. 61

GNSS global navigation satellite system. 60

GPS Global Positioning System. 6

IGN Spanish National Geographic Institute. 61, 78

IMU inertial measurement unit. 81–83

KF Kalman filter. 11

LHS left-hand side. 28

MBS multibody simulation. 25, 27

MEMS microelectromechanical systems. 15, 42, 44

MSL mean see level. 63

NASA National Aeronautics and Space Administration. 5

PNOA Spanish National Plan of Aerial Ortophotography. 78

PSD power spectral density. 47

RMS root mean square. 49, 50, 52, 53

SBW steer-by-wire. 59, 60

SCKF smoothly constrained Kalman filter. 34, 91

UKF unscented Kalman filter. 25, 32, 36, 46, 50, 51, 54–58, 84, 91, 149, 150

UKF FE UKF with forward Euler integrator. 48, 52, 54

UKF TR UKF with trapezoidal rule integrator. 48

WGS84 World Geodetic System 1984. 61

WSS wheel speed sensor. 15, 16

List of symbols

The symbols used all along the document are defined here. The symbols employed in small parts of the document are defined locally.

Δt Time step. 16, 27, 31, 32, 34, 39

$\bar{\mathbf{M}}$ Mass matrix projected over the independent coordinates. 30, 31

$\bar{\mathbf{Q}}$ Vector of generalized forces projected over the independent coordinates. 30

$\mathbf{\Sigma}_c^P$ Covariance matrix of plan noise for continuous-time filters. 13, 31, 39

$\mathbf{\Sigma}^P$ Covariance matrix of plan noise for discrete-time filters. 12, 32–34, 36, 39

$\mathbf{\Sigma}^S$ Covariance matrix of sensor noise. 12, 13, 30–32, 34, 37, 86

$\mathbf{\Sigma}$ Innovation covariance matrix. 12, 32, 37, 86

$\boldsymbol{\alpha}$ Matrix of penalty coefficients. 26, 27

$\boldsymbol{\xi}$ Weakening matrix. 35

$\ddot{\mathbf{\Phi}}$ Second time derivative of the constraints vector. 26, 27

$\ddot{\mathbf{q}}$ Vector of dependent accelerations. 26–28, 34

$\ddot{\mathbf{z}}$ Vector of independent accelerations. 26, 28–32, 39

$\dot{\mathbf{\Phi}}$ Time derivative of the constraints vector. 26, 27, 35, 36, 38

$\dot{\mathbf{R}}$ Time derivative of velocity projection matrix R. 28–31

$\dot{\mathbf{q}}$ Vector of dependent velocities. 26–30, 34, 35, 37, 38, 44, 45, 66–68, 79, 80, 82–84

$\dot{\mathbf{z}}$ Vector of independent coordinates. 26, 28–32, 36, 38, 43

$\mathbf{0}$ Null matrix. 25–28, 30–32, 34, 36–39, 66–68, 80, 83, 84

$\mathbf{A}$ Continuous system transition matrix. 13

$\mathbf{B}$ Continuous input matrix. 13

$\mathbf{D}$ Projection matrix. 28, 38

List of symbols

$\mathbf{F}$ Transition matrix of a discrete linear system. 12

$\mathbf{G}$ Input matrix. 12

$\mathbf{H}$ Measurement sensitivity matrix. 12, 13

$\mathbf{I}$ Identity matrix. 12, 30–32, 34, 35, 37, 39, 66–68, 80, 84

$\mathbf{K}$ Kalman gain. 12, 13, 30–32, 34, 35, 37, 86

$\mathbf{M}$ Mass matrix. 26–31, 65–68

$\mathbf{P}$ Covariance matrix of state estimation uncertainty. 12, 13, 30–37, 50, 86

$\mathbf{Q}$ Vector of generalized forces. 26–30, 67, 68

$\mathbf{R_q}$ Jacobian of the matrix R wrt the dependent coordinates. 30

$\mathbf{R}$ Velocity projection matrix R, such as $\dot{\mathbf{q}} = \mathbf{R}\dot{\mathbf{z}}$. 28–31, 38

$\mathbf{S}$ Transformation matrix for rheonomous constraints. 28–30, 38

$\boldsymbol{\Psi}$ Rotation matrix. 14, 15, 62, 81–84

$\bar{\mathbf{C}}$ Damping matrix. 27, 30, 31

$\bar{\mathbf{K}}$ Stiffness matrix. 27, 30, 31

$\mathfrak{F}$ Jacobian of the continuous-time transition model (transition matrix if the system is linear). 30, 31

$\mathfrak{f}$ Continuous-time transition model. 30

$\mathbf{f}$ Transition function. 32–34

$\mathbf{h_x}$ Jacobian of measurement equation wrt states. 30–32, 34, 37, 38, 43–46, 55, 86

$\mathbf{h}$ Measurement equation. 30–34, 36–38, 43–45, 79, 80, 82–84, 86

$\mathbf{o}$ Vector of observations. 12, 30–34, 36, 37, 86

$\mathbf{q}$ Vector of dependent coordinates. 25–30, 34, 35, 37, 38, 42–45, 65–68, 79, 80, 82–84

$\mathbf{u}$ Vector of inputs. 12, 13

$\mathbf{v}$ Vector of measurement noise. 12, 13

$\mathbf{w}$ Vector of plant noise. 12, 13

$\mathbf{y}$ Vector of outputs. 12, 13

$\mathbf{z}$ Vector of independent coordinates. 25, 30–32, 36, 38, 43, 85

$\tilde{\mathbf{y}}$ Vector of innovation. 12, 32, 34, 36, 37, 86

g Number of degrees of freedom. 30–33, 39

l Size of the vector of states $\mathbf{x}$. 33, 55

n Size of the vector of dependent coordinates $\mathbf{q}$. 34, 39

$\mathbf{\Phi_q}$ Jacobian of the constraints vector wrt the coordinates. 25–28, 37, 38, 68

$\mathbf{\Phi}$ Vector of constraints. 25–30, 35, 36, 38, 42

$\boldsymbol{\lambda}$ Vector of Lagrange multipliers. 26–28, 68

$\dot{\mathbf{x}}$ Time derivative of the state. 13

$\hat{\dot{\mathbf{q}}}$ Estimated vector of dependent velocities. 34, 37

$\hat{\dot{\mathbf{x}}}$ Estimation of the time derivative of the state. 13, 30

$\hat{\dot{\mathbf{z}}}$ Estimated vector of independent velocities. 30–33, 37

$\hat{\mathbf{q}}$ Estimated vector of dependent coordinates. 34, 37

$\hat{\mathbf{x}}$ Estimation of the state. 12, 13, 30–37, 86

$\hat{\mathbf{z}}$ Estimated vector of independent coordinates. 31–33, 37, 85

$\mathbf{f_x}$ Jacobian of transition function wrt states. 32, 34, 36, 37, 39, 55

$\mathbf{h_q}$ Jacobian of measurement equation wrt dependent coordinates. 38, 45, 46, 80, 83, 84

$\mathbf{h_z}$ Jacobian of measurement equation wrt independent coordinates. 38, 46, 84

$\mathbf{h}_{\dot{\mathbf{q}}}$ Jacobian of measurement equation wrt dependent velocities. 38, 45, 46, 80, 83, 84

$\mathbf{h}_{\dot{\mathbf{z}}}$ Jacobian of measurement equation wrt independent velocities. 38, 46, 84

$\mathbf{x}$ State. 12, 13, 30–32, 34–36, 43, 44, 85

Chapter 1

Introduction

Multibody simulations are widespread tools employed to reduce development time and cost of new products in industry. Although they were initially used during the design phase of the products, over the last decades, their use has spread to many other applications, including real time interaction with humans (human-in-the-loop simulations), employed for example in driving simulators or virtual reality applications, and hardware-in-the-loop, mainly used to test electronic controllers, such as automotive electronic central units, before accomplishing tests with the final vehicle.

After real-time simulations were achieved, a new idea has emerged: on-board multibody simulations could be employed as virtual sensors to improve the control of machines and vehicles. Obviously, since a perfect model is impossible to achieve, some kind of feedback must be provided to correct the multibody model and obtain reliable measurements.

In this thesis, the Kalman filter is the algorithm employed to correct the state of the multibody model. However, due to the different mathematical structure of the Kalman filter and the multibody simulations, their integration is far from obvious, and several strategies can be employed.

This research studies several alternatives to combine both techniques, focusing both on accuracy and computational cost of the different techniques. Preliminary test were done with four-bar and five-bar planar linkages, to study the increase of computational cost with the size of the problem. Finally, the best candidate from all the explored possibilities was implemented with a multibody model of an automobile. The selection of this problem was motivated by the fact that current commercial vehicles employ several state observers to feed all the safety systems they incorporate. A single multibody-based state observer could be used to substitute all of them.

1.1. Motivation

The Laboratorio de Ingeniería Mecánica of the University of A Coruña has specialized in the development and application of multibody simulations in diverse fields, including biomechanics, virtual reality, simulation of fishing nets, heavy machinery, and automotive, among others. Vehicle models and vehicle simulators were developed before in this laboratory because they represent challenging problems to verify the performance of the multibody formulations developed in previous works. The field of multibody-based state observers was initiated at the Laboratorio de Ingeniería Mecánica with the works of Cuadrado et al. [1, 2] in which a continuous extended Kalman filter was applied to the multibody formulations of simple mechanisms, and a first attempt to apply this formulation to a vehicle model in [3]. The main problem of this method shown in these works was its computational cost, not allowing to run the state observer of a vehicle in real time.

After that, Pastorino completed his PhD. thesis [4], in which new state observers from the family of the sigma-point Kalman filters were taken into consideration. Moreover, in his thesis a multibody model of an automobile was implemented and validated. However, no state observer formulation was found suitable to be applied to such a complex model. Apart from the high computational cost of the methods presented until this time, the work of Pastorino showed that the continuous extended Kalman filter proposed in [1] did not work well if the measurements were not available at every time step of the multibody simulation.

The present work starts from the point reached by Pastorino's work at the end of his PhD. thesis, seeking to find more efficient state observers based on multibody models, and to improve the multibody model of the vehicle developed in his research.

1.2. Objectives

The main objective of this thesis is to obtain a method which allows to run a state observer based on a multibody simulation of a vehicle on board of such vehicle. In order to achieve this aim, several partial goals are set:

- To study different strategies to combine multibody models and Kalman filters. These methods are tested with simple mechanisms to understand how they work, and to evaluate the increase in computational cost with the size of the mechanism.

- To improve the multibody model of the vehicle already available. The model developed in [4] has to be improved in order to enhance its accuracy and reduce its computational cost.

- To install the multibody model on board of the vehicle. A computer program has to be developed to acquire data from the on-boar sensors and to run the multibody model on board of the vehicle in real time.

- To implement a state observer based on the multibody model of the vehicle, and evaluate its accuracy and its computational cost.

1.3. Thesis outline

The remainder of this thesis is organized as follows:

Chapter 2 covers the state of the art of the state observers, in general, and the fields of state observers applied to land vehicles and multibody-based state observers in particular.

Chapter 3 introduces the equations of the continuous and discrete Kalman filters, the multibody formulations used in this thesis, and the state observers based on multibody models which are employed in this research to build the transition and the measurement models.

Chapter 4 shows the applications of the methods previously developed to two planar mechanisms (four-bar and five-bar linkages), with several sensor configurations.

Chapter 5 covers the model improvements realized to the multibody models of the vehicle and the application of a state observer to this model, including the development of all the sensor models.

Chapter 6 draws the conclusions and indicates future research lines.

Chapter 2

State of the art

This thesis explores the fusion of two well established disciplines: on the one hand, multibody dynamics; on the other hand, state observers. Although the combination of both techniques is desirable, the success on achieving this aim has been limited.

In the subsequent sections, both techniques are roughly introduced and a bibliographic review is done, highlighting the use of automotive state observers employed mainly in driving aids, and the previous attempts to build state observers based on multibody simulations.

2.1. State observers

It seems that the earliest stimulus for the development of estimation theory was provided by astronomical studies in which planet and comet motion was studied using telescopic measurement data. To determine the parameters of the revolution of heavenly bodies, the method of least squares was invented by Karl Friedrich Gauss in 1795 [5].

From that time, other tools known as observers, estimators or filters have emerged. They are used to extract information about a system when a sensor cannot be used to do that due to technical or economical constraints. In [6], a survey of different state observers is presented.

From all the state observers, the most used is the Kalman filter, first presented in 1960 [7], and its extensions to nonlinear systems, mainly the extended Kalman filter and the unscented Kalman filter [8]. The Kalman filter is a recursive algorithm which provides optimal results when applied to a linear system affected by white noise, with measurements also affected by white noise. In practice these conditions are rarely met, and hence the extended Kalman filter or the unscented Kalman filter (or other less common nonlinear extensions of the method) have to be used. Then the filter is not optimal, but it has shown good results in numerous applications, including the first practical implementation of the Kalman filter, which was an extended Kalman filter used during the Apollo missions of the National Aeronautics and Space Administration (NASA).

The unscented Kalman filter arose later to solve some of the flaws of the extended Kalman filter: it deals better with highly nonlinear systems, and the explicit calculation of the linearized matrices of the system is avoided by using the unscented transform, which relies in some deterministically chosen samples to propagate the mean and the uncertainty of the estimation [8].

Since its inception, the Kalman filter played an important role in many fields, such as navigation (including the development of the Global Positioning System (GPS) system), radar tracking, and vehicle dynamics among many others [9].

2.2. State observers applied to road vehicles

Road accidents are one of the main causes of mortality around the world, in particular among young people [10]. This fact has motivated a great research effort which led to the development of driving aids, including ABS, electronic stability control, active body control, etc. With the advent of electric cars, which can have one motor per wheel, the torque applied to every wheel can be set independently, allowing to do more complex control strategies.

Other aids can also help to avoid accidents, such as the adaptive cruise control or the involuntary lane departure warning. Finally, the autonomous driving will prevent human-caused accidents.

All these technologies need from accurate dynamical information about the vehicle and its environment. However, some of the most important magnitudes for the vehicle control, such as the slip angle or tire-road friction coefficient cannot be measured by reliable and cost-effective sensors, and hence they must be estimated. Therefore, a number of observers to estimate these magnitudes have been proposed in the literature.

Usually, a simplified dynamical model is specifically developed to be used in the filter, looking for a trade-off between the accuracy and the computational cost of the model. Some of the most representative examples are mentioned hereafter.

In [11], a five degree-of-freedom vehicle model is combined with an extended Kalman filter to estimate the tire forces of a vehicle. This work was extended in [12] to estimate tire forces, motion and coefficient of friction of vehicles on asphalt surfaces.

In 1999, Venhovens et al. [13] presented three state observers used at BMW. The first of them considers the velocity and acceleration of two vehicles, and the distance between them, to be employed in an adaptive cruise control. The second observer uses a bicycle model aimed at estimating the lateral disturbances acting over the vehicle, employed in a lane keeping assistant. The third observer employs a two-track vehicle model and the ABS wheel-speed signals to determine yaw rate and tire slip angles.

In [14] an adaptive extended Kalman filter is presented. It is designed to estimate the yaw rate, sideslip velocity, forward velocity, roll rate, and roll angle, and then it is compared to an equivalent linear Kalman filter. Moreover, two different sets of sensors are considered.

In [15, 16] dual-antenna GPS and inertial sensors are combined to obtain body sideslip angle, wheel slip, and wheel slip angles. In [16] the roll angle and road grade are also estimated.

In [17] four state observers of vehicle sideslip angle and lateral forces are compared, being one of them linear, and the others nonlinear observers, including an extended Luenberg observer, an extended Kalman filter, and a sliding-mode observer. It was concluded that the observers were good enough for the linear region, but the models needed to be improved to obtain good estimations in the nonlinear region.

The design of a diagnosis system for a steer-by-wire system including vehicle dynamics modeling is addressed in [18]. The dynamics of the vehicle is represented by a planar bicycle model, whose states are the sideslip angle, the yaw rate, the steering angle, and its time derivative.

In [19] a dual extended Kalman filter is used to estimate both states and parameters of a vehicle model. The dual Kalman filter consists of two Kalman filters is parallel, one estimating the main parameters of the vehicle model, and the other estimating the states.

A 9 degree-of-freedom four wheel vehicle model is used in [20] in another extended Kalman filter. The degrees of freedom of the model are longitudinal and lateral displacements, the roll, pitch and yaw angles, and the four rotation angles of the wheels. The algorithm is aimed at estimating useful magnitudes to be used in automotive stability controllers. The state vector of the observer has 20 elements, including both states and parameters.

In [21] four observers for tire-road forces, sideslip angle and wheel cornering stiffness are evaluated. The estimation method is based in a two-step algorithm. The first part deals with the estimation of vehicle body dynamics, while the second part calculates the sideslip angle and the cornering stiffnesses. Both parts are based on the extended Kalman filter method, but two different models are considered in each block, resulting in the four methods finally used to compare with experimental data. An evolution of this approach is presented in [22], where the first Kalman filter is substituted for a sliding mode observer using the bicycle model.

A different approach is described in [23], where the estimation problem is split in four different state observers in a strategy called integrated Kalman filter. First, the heading angle is estimated. Then, this information is fed in the tire radius bias and longitudinal velocity estimator. The heading and velocity estimations are employed in other estimator to improve the longitudinal velocity, estimate the lateral velocity and accelerometer biases. Finally, these partial estimations are employed in another Kalman filter based on a bicycle model to estimate the steer bias as well as the sideslip and heading angle.

In 2012 a book about state observers for the estimation of vehicle dynamics was published [24]. The book deals with several techniques to estimate vertical and lateral tire forces. The observers are experimentally validated, and the details of the instrumented vehicle employed in such validation are also provided.

In [25] a more complex approach is used to estimate the sideslip angle: a dynamical Kalman filter using a bicycle model is used as a sensor for a kinematic Kalman filter. A fuzzy decision module determines the reliability of the dynamical filter to be used as a measurement by the kinematic filter.

The work presented in [26] is intended to provide a slip angle estimator in which the computational cost and the number of parameters parameters are reduced. To achieve that, the bicycle model is employed in an extended Kalman filter. An optimization technique is used to obtain the proper values of the plant noise and the tire-road friction coefficient. The method was tested in a real vehicle obtaining good results even without knowing the tire-road friction coefficient.

All the dynamical models employed in these works are based in closed-form models which are far in accuracy and complexity from a multibody model of the vehicle.

2.3. Multibody dynamics

Multibody dynamics is a discipline in which a mechanical system is considered as a finite number of bodies and their connections. Although the dynamics of rigid-body systems was studied before the generalized use of digital computers, the development of multibody dynamics as it is know today followed the early developments in computer science [27]. In 1977, the first practical solution methodology for large multibody systems was presented in [28], which culminated in the development of ADAMS, an acronym for Automatic Dynamic Analysis of Mechanical systems. Since then, the use of multibody dynamics was spread among several fields, including aerospace, automotive, robotics, biomechanics, etc. [29], and several multibody dynamics software appeared [30].

The increase of the available computational power, and the development of new formulations allowed the use of multibody simulations in real time [31]. Then, human-in-the-loop and hardware-in-the-loop become possible. They are used, for example, in driving simulators [32], or when testing new algorithms, such as a new stability controller, where the test vehicle can be simulated during the first development stages [33]. Recent developments have been done towards the implementation of real-time multibody models in low-cost hardware platforms [34].

2.4. State observers applied to multibody models

Once real-time multibody simulations are available, they become an attractive option to be used in controllers or in state observers, since they can be installed on board of vehicles, robots or machines as part of their control system. However, the inclusion of a multibody model inside a state observer is far from obvious. The reason is the mathematical structure of both problems: while state observers are usually formulated for first order, unconstrained, linear problems, the equations of a multibody model are, in general, second order, constrained, and non linear.

In this field, the non linear extensions of the Kalman filter have been explored. This research line started with the work of Cuadrado et al. [1, 2]. In [1], two multibody formulations, the matrix R and a penalty method, are combined with a continuous-time extended Kalman filter (CEKF), showing that the first method works better. In [2], different kinds of sensors are studied (position, velocity, and acceleration). In 2012, an improved version of the same method was applied to an automobile [3]. Although the algorithm was redesigned to achieve an enhanced efficiency, its computational cost prevented it from running in real time.

Several variants of the unscented Kalman filter were presented in [35, 36]. The methods presented in these works are much easier to implement than the former CEKF, and their accuracy is better in some circumstances, especially when the measurements from sensors have a lower update rate than the integration loop of the observer. However, the computational cost of the methods proposed in these papers is even higher than that of the CEKF.

In [37], the application of multibody-based state observers has been extended to the estimation of inputs, i.e, the forces driving the motion of the mechanism under study. In this work, simplified multibody models are employed in order to achieve real-time performance. With the sensor setup employed in this paper, the system is not observable, and hence thew method would finally diverge if position sensors are not added. If the method is to be applied to structures or mechanisms oscillating around a known equilibrium position, then dummy sensors at position level might be used to achieve the observability of the system [38].

Other force observer based in multibody models was proposed in [39]. This state observer is based on two steps: first, a Kalman filter estimates the kinematics of the mechanism, as shown in [40]. Then, this observer feeds a disturbance observer which estimates the unknown forces applied to the mechanism under study.

In this thesis, several multibody-based state observers were developed and tested. They are explained in chapter 3. Partial results of these developments have been published in [41, 42, 43, 44].

Chapter 3

State observers

State observers are tools which provide an estimation of the state of a system, combining information provided by a mathematical model with data supplied by some measurements, which are never perfect. Moreover, unmeasured internal states can be made accessible by using these techniques. State observers are recursive algorithms which include the new measurements when they are available, making them usable in real-time applications, such as providing information to controllers. Several state observers were designed throughout history [6], being the Kalman filter (and its variants) the most famous of them.

This chapter covers the application of Kalman filters to multibody models. To do that, a reminder of the equations of the Kalman filter, both in its discrete-time and continuous-time form are presented. A roll angle observer for a bicycle is presented as an application example of a discrete-time Kalman filter. Then, the multibody approach to kinematics is introduced. Kinematics is used to solve the position and velocity problems in filters in independent coordinates, to impose the constraints in methods in dependent coordinates, and to develop sensor models and their Jacobian matrices for the extended Kalman filters. After that, the two multibody formulations employed in this thesis are described. The final part of the chapter is focused on the description of the methods combining multibody formulations and Kalman filters employed in this thesis.

3.1. Kalman filter (KF)

A Kalman filter is an optimal recursive data processing algorithm, which propagates the conditional probability density of the desired quantities, conditioned on the knowledge of the data provided by the sensors [45]. Here, a reminder of the discrete-time Kalman filter and the continuous time Kalman filter equations will be done. The reader interested in a more in-depth discussion about the topic can refer to [46] and [47].

3.1.1. Discrete-time KF

First presented in 1960 [7], the discrete Kalman filter consist of two main steps: the prediction, and the correction. During the prediction, the state and the covariance of the error of the state matrix are propagated according to a linear model. Since this is the discrete version of the filter, the model used to propagate the state is a difference equation, obtained by integration from the equivalent differential equation. The model is then expressed as follows:

$$\begin{aligned} \mathbf{x}_k &= \mathbf{F}\mathbf{x}_{k-1} + \mathbf{G}\mathbf{u}_{k-1} + \mathbf{w}_k && (3.1)\\ \mathbf{y}_k &= \mathbf{H}\mathbf{x}_k + \mathbf{v}_k && (3.2)\\ \mathbf{w} &\sim N(0, \mathbf{\Sigma}^P) && (3.3)\\ \mathbf{v} &\sim N(0, \mathbf{\Sigma}^S) && (3.4) \end{aligned}$$

being $\mathbf{x}$ the state, $\mathbf{F}$ the transition matrix, $\mathbf{G}$ the input matrix, $\mathbf{u}$ the input vector, $\mathbf{y}$ the output vector, and $\mathbf{H}$ the measurement sensitivity matrix. Plant noise $\mathbf{w}$ and measurement noise $\mathbf{v}$ are additive Gaussian white noise with mean 0 and covariance matrices $\mathbf{\Sigma}^P$ and $\mathbf{\Sigma}^S$ respectively, and the subindex k represents the time step. The equations of the propagation stage are the next ones:

$$\begin{aligned} \hat{\mathbf{x}}_k^- &= \mathbf{F}\hat{\mathbf{x}}_{k-1}^+ + \mathbf{G}\mathbf{u}_{k-1} && (3.5)\\ \mathbf{P}_k^- &= \mathbf{F}_{k-1}\mathbf{P}_{k-1}^+\mathbf{F}_{k-1}^\top + \mathbf{\Sigma}^P && (3.6) \end{aligned}$$

where the hat ˆ represents estimated magnitudes, and $\mathbf{P}$ is the covariance matrix of the estimation error. The superindices '-' and '+' mean 'a priori' (i.e. before measurements of the current time step are applied) and 'a posteriori' magnitudes (after applying the measurements), respectively.

If measurements are available after the prediction stage, they are employed to improve the estimation of the magnitudes calculated in the prediction step. The first phase is calculating the mismatch $\tilde{\mathbf{y}}$ (often called innovation) between the sensors measurements (or observation, $\mathbf{o}$) and the virtual sensors according to the predicted states $\mathbf{H}\hat{\mathbf{x}}$:

$$\tilde{\mathbf{y}}_k = \mathbf{o}_k - \mathbf{H}\hat{\mathbf{x}}_k^- \qquad (3.7)$$

Then, a weighting matrix, usually called Kalman gain $\mathbf{K}$, is calculated as follows:

$$\begin{aligned} \mathbf{\Sigma}_k &= \mathbf{H}\mathbf{P}_k^-\mathbf{H}^\top + \mathbf{\Sigma}^S{}_k && (3.8)\\ \mathbf{K}_k &= \mathbf{P}_k^-\mathbf{H}^\top\mathbf{\Sigma}_k^{-1} && (3.9) \end{aligned}$$

The innovation covariance matrix $\mathbf{\Sigma}_k$ in Eq. (3.8) represents the uncertainty in the system state projected via the measurement sensitivity matrix $\mathbf{H}$ plus an additional covariance $\mathbf{\Sigma}^S$ due to the Gaussian noise originated in the sensor itself. Small values of $\mathbf{\Sigma}$ mean that the observation introduces useful information to constrain the estimation of the system state.

The estimation of the state $\hat{\mathbf{x}}$ and the covariance matrix of the estimation error are finally corrected employing the Kalman gain:

$$\begin{aligned} \hat{\mathbf{x}}_k^+ &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k\tilde{\mathbf{y}}_k && (3.10)\\ \mathbf{P}_k^+ &= (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_k^- && (3.11) \end{aligned}$$

If measurements are not available, the correction stage is omitted.

3.1.2. Continuous-time KF

In 1961, Kalman and Bucy presented the continuous time-Kalman filter, also known as the Kalman-Bucy filter [48]. Although this method is not usually employed in practice, it has demonstrated its value as a tool to tackle theoretical demonstrations. Moreover, it is included here because one of the multibody-based state observers presented subsequently is based on this formulation.

First, a linear system is considered, now in its differential form:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{w} && (3.12)\\ \mathbf{y} &= \mathbf{H}\mathbf{x} + \mathbf{v} && (3.13)\\ \mathbf{w} &\sim N(0, \mathbf{\Sigma}_c^P) && (3.14)\\ \mathbf{v} &\sim N(0, \mathbf{\Sigma}^S) && (3.15) \end{aligned}$$

being **A** the continuous system transition matrix, **B** the continuous input matrix. Notice that now the vector of plant noise **w** has the dimensions of the time derivative of the states $\dot{\mathbf{x}}$.

Then, the continuous-time Kalman filter equations are as follows:

$$\begin{aligned} \mathbf{K} &= \mathbf{P}\mathbf{H}^{\top}\left[\mathbf{\Sigma}^S\right]^{-1} && (3.16)\\ \dot{\hat{\mathbf{x}}} &= \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}\mathbf{u} + \mathbf{K}\left(\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}\right) && (3.17)\\ \dot{\mathbf{P}} &= \mathbf{A}\mathbf{P} + \mathbf{P}\mathbf{A}^{\top} - \mathbf{P}\mathbf{H}\left[\mathbf{\Sigma}^S\right]^{-1}\mathbf{H}\mathbf{P} + \mathbf{\Sigma}_c^P && (3.18) \end{aligned}$$

If this filter is implemented in a digital computer, then both the states **x** , and the covariance of the estimation error **P** have to be integrated from Eqs. (3.17) and (3.18).

3.2. Application: roll angle observer for a bicycle

In this section, an example of application of a conventional Kalman filter is provided, before starting with the description of the state observers based on multibody models. The method presented here was developed during a stay at the Bicycle Dynamics Laboratory of Delft University of Technology (The Netherlands). The objective of this research was to obtain a method to measure the lean angle (roll) of a single track vehicle (bicycle or motorcycle) with respect to gravity.

Single track vehicles can reach high roll angles during their normal operation, and this magnitude heavily affects the dynamics of the whole vehicle. Being able of measuring the lean angle with respect to the gravity is paramount in order to develop advanced ABS systems or stability controllers for single track vehicles, or to build autonomous vehicles of this kind. However, there are no sensors which can measure the roll angle directly, thus it has to be estimated from other measurements.

In this example, a wheel speed sensor, and a three axis angular rate sensor were used to implement the state observer on a bicycle. Optic sensors were installed on both sides of the bicycle to validate the method, although the roll measurements from the optic sensors are only valid when riding on flat roads.

3.2.1. Roll angle estimator

The attitude of a vehicle can be expressed by means of aircraft principal axes (as shown in Figure 3.1), using the roll (ϕ), pitch (θ), and yaw (ψ) angles. The order of the rotations is yaw-pitch-roll if they are expressed as z-y-x intrinsic rotations, or roll-pitch-yaw if they are expressed as x-y-z extrinsic rotations. The intrinsic rotations can be drawn using the so-called cans in series representation [49], as shown in Figure 3.2.

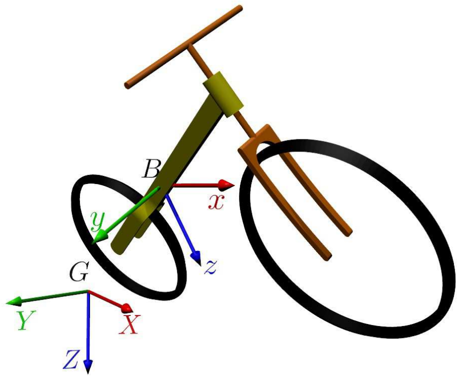


Figure 3.1: Single track vehicle with axes used: a global inertial frame G : (X, Y, Z) and a body fixed frame B : (x, y, z) attached to vehicle body, here the rear frame of the single track vehicle. The orientation of B with respect to G is described the aircraft Euler angles yaw (ψ), pitch(θ) and roll(ϕ).

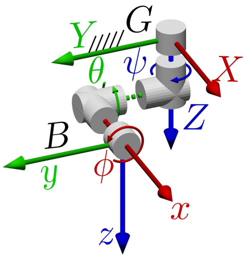


Figure 3.2: Order of yaw, pitch, roll rotations depicted by means of the cans in series [49].

A vector $\mathbf{u}^B$ expressed in the body reference system can be transformed into a vector $\mathbf{u}^G$ expressed in the global reference system using the rotation matrix:

$$\mathbf{u}^G = \mathbf{\Psi}_\psi \mathbf{\Psi}_\theta \mathbf{\Psi}_\phi \mathbf{u}^B = \mathbf{\Psi} \mathbf{u}^B \tag{3.19}$$

where,

$$\boldsymbol{\Psi}_\psi = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.20}$$

$$\boldsymbol{\Psi}_\theta = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \tag{3.21}$$

$$\boldsymbol{\Psi}_\phi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \tag{3.22}$$

This example is focused on the estimation of the roll angle of the rear frame of a single track vehicle with respect to gravity. In order to do that, it is assumed that a wheel speed sensor (WSS) and angular rate measurements are available. The angular rate sensors are assumed to be aligned with the axes in Figure 3.1, although if this condition cannot be met in an actual assembly, it can be solved by applying a rotation to the measurements. The selection of these sensors was conditioned by their cost and availability: WSS are already installed in every motorcycle equipped with ABS, while a three-axis MEMS angular rate sensor is a very affordable and small size device, hence it is suitable to be installed in any kind of vehicle.

Several data can be extracted from the selected sensors, but none of them is an accurate nor reliable measurement of the roll angle. For this reason, the different pieces of information obtained with this set of sensors have to be combined to get the most accurate information from them. This is achieved by means of a discrete Kalman filter in this example.

3.2.1.1. Dynamical model of the filter

The model employed in the filter has two states: the roll angle, and the bias of the angular rate sensor along the body fixed x-axis.

The angular velocity tensor $\boldsymbol{\Omega}^B$ of the rear frame in local coordinates can be expressed as follows (see section 5.6.3 for further details):

$$\boldsymbol{\Omega}^B = \boldsymbol{\Psi}^\top \dot{\boldsymbol{\Psi}} \tag{3.23}$$

Then, the relationship among the angular rates measured by the body-mounted angular rate sensors $(\omega_x^B, \omega_y^B, \omega_z^B)$ and the time derivative of the roll ($\dot{\phi}$), pitch ($\dot{\theta}$), and yaw ($\dot{\psi}$) angles of the vehicle body can be expressed as follows:

$$\dot{\phi} = \left(\omega_y^B \sin(\phi) + \omega_z^B \cos(\phi)\right) \tan(\theta) + \omega_x^B \tag{3.24}$$

$$\dot{\theta} = \omega_y^B \cos(\phi) - \omega_z^B \sin(\phi) \tag{3.25}$$

$$\dot{\psi} = \frac{\omega_y^B \sin(\phi) + \omega_z^B \cos(\phi)}{\cos(\theta)} \tag{3.26}$$

Assuming a small pitch angle, $|\theta| << 1$, Eq. (3.24) becomes

$$\dot{\phi} \approx \omega_x^B \tag{3.27}$$

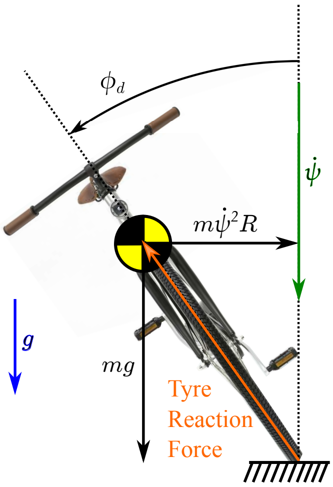


Figure 3.3: Dynamic force equilibrium during steady state cornering.

The bias of de x angular rate sensor (b_x) can be modelled as a random walk, i.e. assuming that it is constant and that the variations are produced by the plant noise. Once b_x is known, ω_x^B can be corrected. Therefore, after applying the forward Euler integration method, the dynamic model of the filter in Eq. (3.5) becomes:

$$\begin{bmatrix} \hat{\phi} \\ \hat{b}_x \end{bmatrix}_k^- = \begin{bmatrix} 1 & -\Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{\phi} \\ \hat{b}_x \end{bmatrix}_{k-1}^+ + \begin{bmatrix} \Delta t \\ 0 \end{bmatrix} \omega_{x,k-1} \tag{3.28}$$

3.2.1.2. Absolute ‘measurements’ of the roll angle

In the correction stage of the Kalman filter (in particular, in Eq. (3.7)), absolute measurements of the roll angle are needed. However, as actual measurements of the roll angle are not usually available, they are created by using data from a WSS and a three-axis angular rate sensor. In this example, different approaches are employed to obtain the absolute roll angle measurements, and then both are combined into a single measurement.

The first roll angle measurement has been used by many others (e.g. [50, 51, 52]). It assumes a steady state cornering motion and uses the lateral dynamics equilibrium to calculate the first estimation of the roll angle, ϕ_d (see Figure 3.3)

$$\phi_d = \arctan\left(\frac{\dot{\psi}^2 R}{g}\right) = \arctan\left(\frac{v^2}{Rg}\right) = \arctan\left(\frac{\dot{\psi} v}{g}\right) \tag{3.29}$$

where v is the forward speed of the vehicle and R is the radius of the trajectory, and g is the acceleration of gravity. Notice that this equation is only valid for stationary turns on level grounds. Although the three expressions in Eq. (3.29) are equivalent, the third one is more convenient for the implementation, since the radius of the trajectory is not involved in the equation. The speed of the vehicle is usually calculated from the angular rate of the wheels. The calculation of the yaw rate $\dot{\psi}$ is trickier, since its exact value is related to the body angular rates through Eq. (3.26), which involves the roll and pitch angles. However, in this work ω_z^B is used as an estimation of the yaw rate $\dot{\psi}$.

The main drawbacks of this approach are the following:

- The equations were developed with a level road in mind, so the bigger the slope, the larger error is expected.
- The cross section of the tires is not considered. However, if thick tires are employed (for example, in sport motorcycles), they are going to produce some effects when the vehicle is leaned, namely:
  - The contact point of the wheels is not in the symmetry plane of the vehicle.
  - The effective radius of the wheel decreases, thus the calculation of the speed of the vehicle is wrong.
- The gyroscopic effect of the wheels has an upright effect which has to be compensated by leaning more the vehicle.
- The roll angle is not properly estimated during transients (i.e., when $\dot{\phi} \neq 0$), even when using high pressure, thin tires while riding on a flat floor.
- An exact measurement of the yaw rate is not available if the roll or pitch of the vehicle are non-zero.
- This technique considers that the centre of mass of the vehicle is in the symmetry plane, but it is no longer true if the rider leans or displaces laterally his body.

All these effects become more important for big roll angles, but are almost negligible when riding straight.

For big roll angles, a new approach for measuring the roll angle was developed in this work. The method is based on the existent relationship between the angular rates measured by the angular rate sensor if a null pitch rate $\dot{\theta}$ is considered. This approximation relies on the fact that the long term mean of the pitch rate is null during riding. Introducing the null pitch assumption, Eq. (3.25) becomes:

$$0 = \omega_y^B \cos(\phi) - \omega_z^B \sin(\phi) \tag{3.30}$$

and the roll angle estimation ϕ_ω can be calculated as

$$\phi_\omega = \arctan\left(\frac{\omega_y^B}{\omega_z^B}\right) = \mathrm{sgn}(\omega_z^B) \arcsin\left(\frac{\omega_y^B}{\sqrt{(\omega_y^B)^2 + (\omega_z^B)^2}}\right) \tag{3.31}$$

where $\mathrm{sgn}(\omega_z^B)$ is the sign of the z angular rate. Both versions of this equation are mathematically equivalent, but the second one works better when used with noisy measurements taken from real sensors.

This approach has some advantages:

- It works well in roads with any slope, as long as the pitch of the vehicle doesn't change

- it is not affected by the tire thickness
- It is not affected by the rider's movements
- It is not affected by the gyroscopic effect of the wheels

And also some drawbacks:

- It does not work if the vehicle is not turning
- It is usually noisier than the method based on the lateral dynamics of the steady state cornering.

In order to take advantage of both methods for measuring the roll angle, they are combined using a weighted mean. The weighing function changes its value depending on the last available estimation of the roll angle $\hat{\phi}$, taken from the state observer:

$$W = \exp\left(-C\hat{\phi}^2\right) \tag{3.32}$$

where C is a constant value which can be used to adjust the behaviour of the weighting function. The roll angle measurement to be employed in Eq. (3.7), ϕ_m, is build as weighed combination of ϕ_d and ϕ_ω:

$$\phi_m = W\phi_d + (1 - W)\,\phi_\omega \tag{3.33}$$

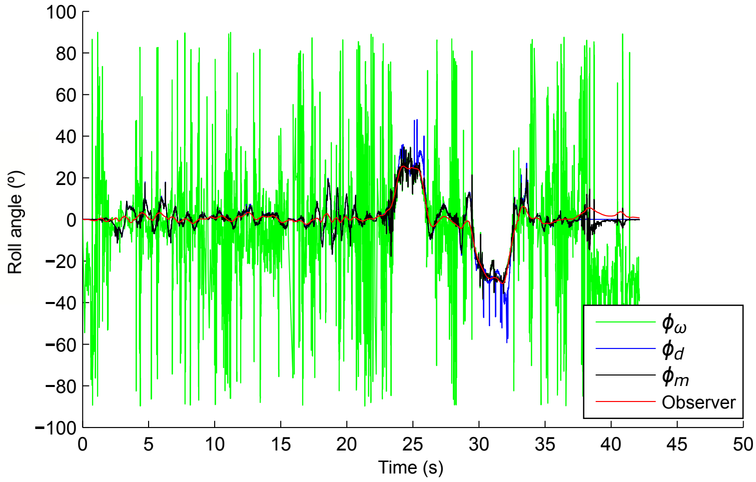


Figure 3.4: Comparison among the different roll angle estimations. The green line is roll angle estimation calculated from the relationship between y and z angular rates, ϕ_ω. The blue line is the roll angle estimation calculated from the steady state cornering dynamics, ϕ_d. The black line is the combination of both measurements ϕ_m as in Eq. (3.33). Finally, the red line is the estimation of the roll angle provided by the state observer.

The results from the various roll angle estimations during a maneuver can be seen in Figure 3.4. It can be appreciated that the method based on the relationship between angular velocities (green line) does not work when the vehicle goes straight, but it improves when the vehicle is leaned. The method based on the lateral dynamics of the steady state cornering (blue line) works well riding straight, but it performs bad at high roll angles, or during transients. However, combining both estimations according to Eq. (3.33) (black line) provides an improved approximation of the roll angle.

3.2.2. Experimental setup

In order to check the behaviour of the method in a real vehicle, a bicycle was instrumented, see Figure 3.5. The core of the system is an Arduino Due, a development board based on the Atmel SAM3X8E microcontroller (Figure 3.6). The Arduino acquires the data from the sensors, runs the roll angle observer, and logs the data to a SD card.

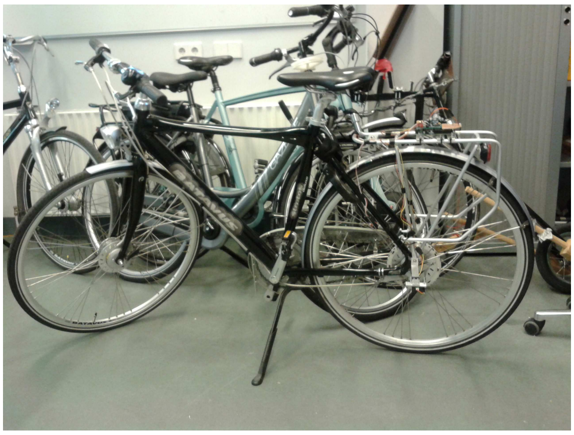


Figure 3.5: Instrumented bicycle.

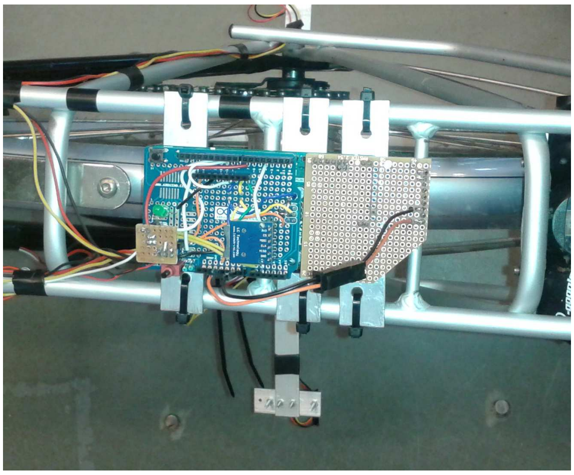


Figure 3.6: Arduino Due board with the angular rate sensors and the SD memory card added.

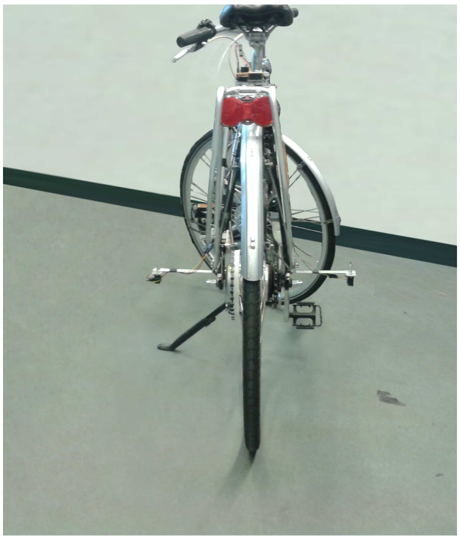


Figure 3.7: Bicycle rear view with left and right distance sensors for roll angle validation measurements.

In order to run the roll angle observer, the forward speed of the bicycle and the three angular rates of the rear frame are needed. The forward speed of the bicycle was measured with a toothed crown (Figure 3.8) fixed on the rear wheel and a through-beam infrared sensor (Omron EE-SX1041). The angular rates of the rear frame were measured with a three-axis angular rate sensor (STmicro L3GD20H) attached to the Arduino board.

For validation purposes, two infrared distance sensors (Sharp GP2Y0A02YK0F) were installed at both sides of the bicycle (Figure 3.7). The roll angle can be calculated from the measurements provided by these sensors using the following equation (see Figure 3.9):

$$\phi = \arctan\left(\frac{D_L - D_R}{D_H}\right) \tag{3.34}$$

where D_L and D_R are the distances measured by the sensors on the left and on the right respectively, and D_H is the distance between the sensors measured perpendicularly to the symmetry plane of the bicycle.

However, it was found that the distance measurements are quite noisy, and they are affected by the irregularities of the road: white lines, the different height of the pavement with respect to the road, etc. For this reason, the angle is calculated using a different algorithm:

- First, the roll angle is calculated according to the sensors of both sides, as follows:

$$\phi_L = \arctan\left(\frac{D_L - H_S}{\frac{D_H}{2}}\right) \tag{3.35a}$$

$$\phi_R = \arctan\left(\frac{H_S - D_R}{\frac{D_H}{2}}\right) \tag{3.35b}$$

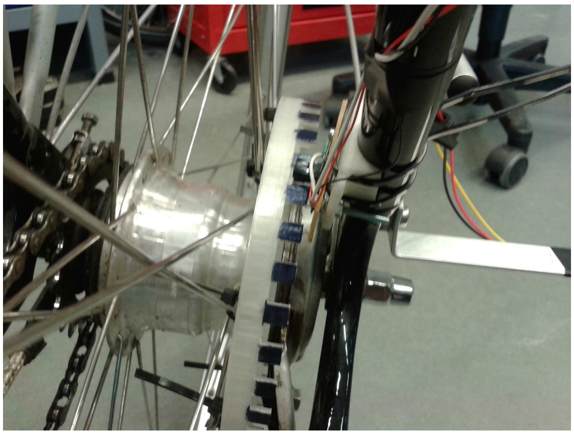


Figure 3.8: Rear wheel speed sensor with toothed crown and photoelectric sensor.

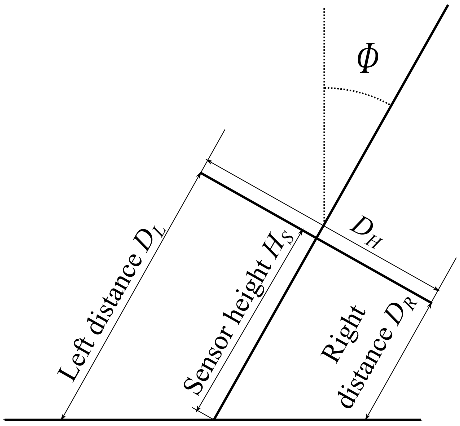


Figure 3.9: Distance sensors measurements (rear view).

where H_S is the sensor height at zero roll angle.

- Then, if both calculations of the roll angle are similar ($|\phi_L - \phi_R| < 0.3rad$), they are averaged. Otherwise, the one which has the larger difference with the previous roll angle calculation is discarded.
- Finally, a low pass filtering algorithm with a cutoff frequency of 2 Hz is applied.

After this process is done, the roll angle measurements from the optical sensors are quite consistent.

3.2.3. Results and discussion

In order to be able to validate the roll angle observer, the tests must be performed in a flat level ground, because the optical validation system measures the roll angle relative to the ground, whereas the roll angle observer measures the roll angle with respect to gravity. An image of the test track where the tests were conducted can be seen in Figure 3.10.

Figure 3.10: Test track employed for the tests.

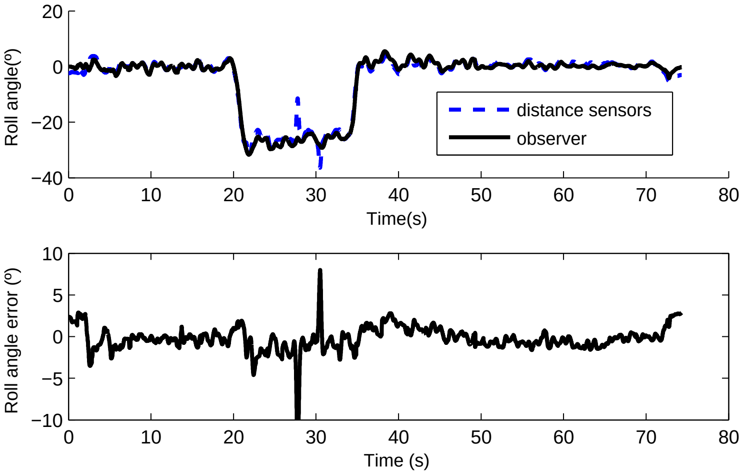


Figure 3.11: Roll angle from distance sensor (validation) and state observer during straight line/constant radius/straight line maneuver (counter-clockwise turn).

All the maneuvers started and finished with null velocity. In this situation, the rider steps on the floor to stabilize the bicycle, and the dynamics of a single track vehicle cannot be applied. For this reason, a settling time can be seen in all the tests at the beginning of the maneuvers, and also a drift at their ending.

The first test started with a straight section. After that, a constant radius turn was performed, and finally, the bicycle was ridden straight again. Results of this maneuver can be seen in Figure 3.11. There are two big errors around 30 seconds, but looking at the plot, it becomes clear that those errors were produced in the optical sensors, since the maneuver was a constant turn, and the signal with sudden variations is the one coming from the distance sensors. This maneuver demonstrates that the algorithm is stable in both straight and turning conditions, and that it also works well in the transitions between both situations. The RMSE for this maneuver was 1.5 degrees.

The second test conducted was a combination of obstacle avoidance and lane

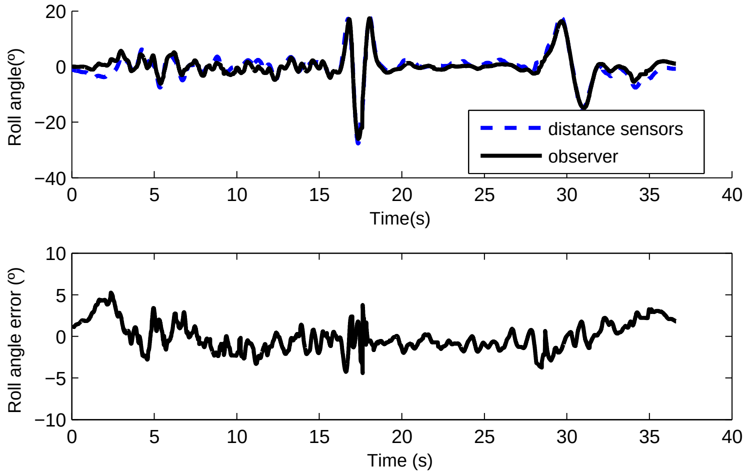


Figure 3.12: Roll angle from distance sensor (validation) and state observer during obstacle avoidance maneuver and lane change.

change (see Figure 3.12) . The maneuver started riding straight. After that, three consecutive turns were performed to avoid an obstacle and return to the original trajectory. Finally, a lane change was performed. This test shows that the roll angle observer performs well in fast series of alternative turns. The RMSE for this maneuver was 1.7 degrees.

The third test was a maneuver looking for the maximum roll angle, such that the bicycle was ridden at maximum speed in a sharp corner. An on-board picture of this test is shown in Figure 3.13, demonstrating the large roll angle. The results are shown in Figure 3.14. The RMSE for this test was 1.8 deg. The spike after 18 seconds is partially due to the delay produced by the low pass filter applied to the distance sensors.

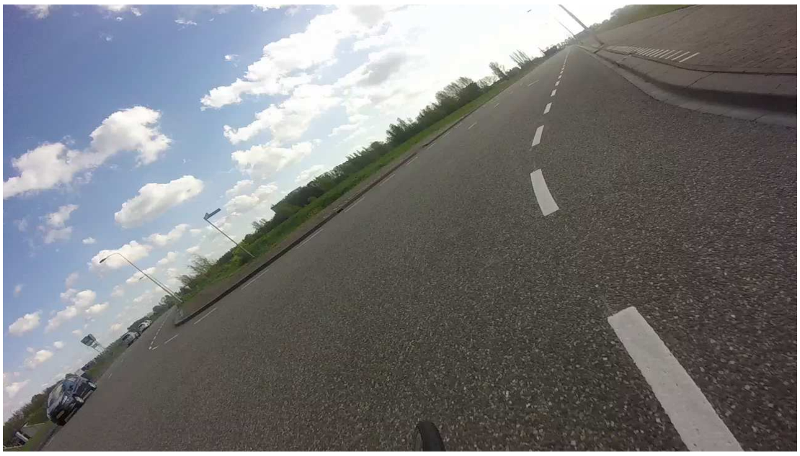


Figure 3.13: On board view of the aggressive cornering maneuver.

The last test was executed doing normal cycling in public roads, including potholes, slopes, bank angle, etc. Since the roads are usually not flat, the measurements from the distance sensors do not provide the correct roll angle. However, this test is useful to verify the stability of the observer during normal riding conditions. The

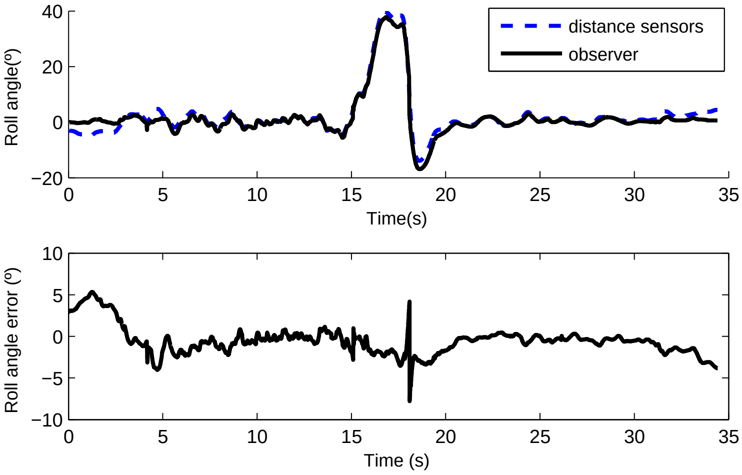


Figure 3.14: Roll angle from distance sensor (validation) and state observer during aggressive turning maneuver.

results can be seen in Figure 3.15. In the roll angle plot, it can be seen that the roll angle provided by the observer is always around zero, while the plot of the roll angle provided by the distance sensors has several sections in which the angle is not around zero, due to the non-zero bank angles in the several sections of the road employed for this test.

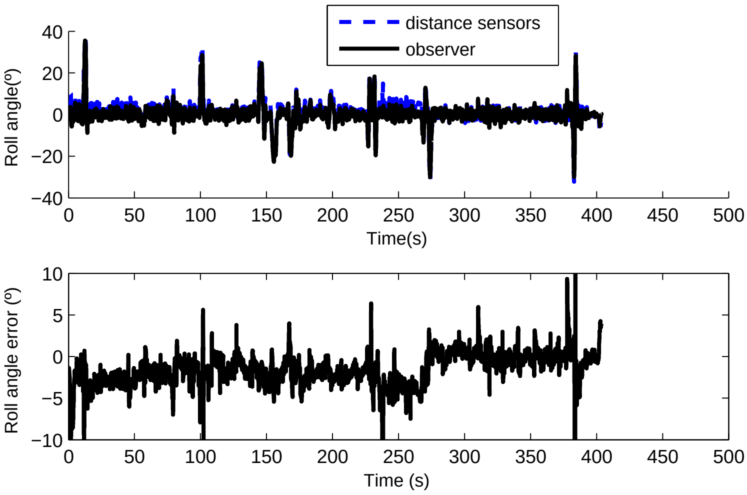


Figure 3.15: Roll angle from distance sensor and state observer during normal riding conditions.

This example shows that the Kalman filter can provide accurate results even when using a very simple model. The simplicity of the model in this case also keeps the computational cost low enough to allow the implementation of the method in a microcontroller. However, in applications which require more information, or more complex magnitudes, more detailed models would be needed. In these cases, it is worth considering the multibody approach to design the dynamical model of the filter.

3.3. Multibody simulation (MBS)

Several different options can be used for the parameterization of the motion of multibody systems but, in general, a set of dependent coordinates is needed to write the dynamic equations of complex systems. Classical parameterizations in multibody dynamics are reference point coordinates [53], natural coordinates [31], or relative (joint) coordinates [54, 55, 56]. In this work, mixed coordinates were used, employing mainly the natural coordinate approach, but adding some relative coordinates when it was convenient.

To implement multibody-based state observers, two main parts of the multibody theory must be known: the kinematics, and the forward dynamics.

The kinematics is employed to impose the constraints, and to develop the Jacobian matrices of the measurement models.

Regarding the forward dynamics, two formulations were used in this thesis. One of them is the matrix R formulation, which is used in the CEKF, described in section 3.4.1. Some parts of the matrix R formulation are also employed in the DEKF method, in section 3.4.2. Moreover, some kinematic expressions arisen from the development of this dynamical formulation are employed to build the Jacobian matrices of the measurement models, as shown in section 3.5. The other dynamical formulation is an an index 3 augmented Lagrangian formulation with velocity and acceleration projections. This formulation is not employed in the development of any filter, but its efficiency makes it a good option in methods in which the multibody formulation can be freely selected, such as the UKF (section 3.4.3) and the error-state extended Kalman filter (errorEKF), in section 3.4.6.

3.3.1. Kinematics

The multibody simulation (MBS) approach to the kinematic problems is introduced in this section. A more detailed discussion can be seen in [31]. The first step is solving the position problem. That is, obtaining the value of all the coordinates ($\mathbf{q}$) which fulfill the constraint equations once the values of the degrees of freedom ($\mathbf{z}$) are known. To do that, the constraint equations are written as follows:

$$\mathbf{\Phi} = \mathbf{0} \tag{3.36}$$

where $\mathbf{\Phi}$ is the vector of the constraint equations. The constraint equation are, in general, nonlinear equations, hence an iterative method is used to solve the problem. Usually, the Newton-Raphson method, which is based on a linearization, is employed:

$$\mathbf{\Phi}(\mathbf{q}, t) \approx \mathbf{\Phi}(\mathbf{q}_0, t) + \mathbf{\Phi}_{\mathbf{q}}(\mathbf{q}_0, t)(\mathbf{q} - \mathbf{q}_0) = \mathbf{0} \tag{3.37}$$

where $\mathbf{\Phi}_{\mathbf{q}}$ is the Jacobian matrix of the constraints with respect to the coordinates. Rearranging the terms of this equation, the next iterative expression is obtained:

$$\mathbf{\Phi}_{\mathbf{q}}(\mathbf{q}_i, t)(\mathbf{q}_{i+1} - \mathbf{q}_i) = -\mathbf{\Phi}(\mathbf{q}_i, t) \tag{3.38}$$

where the values correspondent to the degrees of freedom are not modified. This system must be solved until the position problem converges to the desired accuracy.

The next Kinematic problem to be solved is the velocity problem. Similarly to the position problem, the velocity problem consist in finding the vector of dependent velocities $\dot{\mathbf{q}}$ which fulfill the velocity constraints, given a position $\mathbf{q}$ of the system, and the velocity of the degrees of freedom, $\dot{\mathbf{z}}$. The velocity constraints $\dot{\boldsymbol{\Phi}}$ can be expressed as follows:

$$\dot{\boldsymbol{\Phi}} = \boldsymbol{\Phi}_{\mathbf{q}}\dot{\mathbf{q}} + \boldsymbol{\Phi}_t = \mathbf{0} \quad \Rightarrow \quad \boldsymbol{\Phi}_{\mathbf{q}}\dot{\mathbf{q}} = -\boldsymbol{\Phi}_t \tag{3.39}$$

where $\boldsymbol{\Phi}_t$ is the partial derivative of $\boldsymbol{\Phi}$ with respect to time t. This problem is linear, so the unknown values of $\dot{\mathbf{q}}$ are obtained in one step. Again, the values of the degrees of freedom are known, so they are not modified.

The last problem in kinematics is the acceleration problem: provided a position vector $\mathbf{q}$ and a velocity vector $\dot{\mathbf{q}}$ which fulfill the constraints at position and velocity respectively, and given the acceleration $\ddot{\mathbf{z}}$ of the degrees of freedom of the mechanism, the vector of dependent accelerations $\ddot{\mathbf{q}}$ which fulfill the acceleration constraints is obtained. The constraints at acceleration level can be written as:

$$\ddot{\boldsymbol{\Phi}} = \dot{\boldsymbol{\Phi}}_{\mathbf{q}}\dot{\mathbf{q}} + \boldsymbol{\Phi}_{\mathbf{q}}\ddot{\mathbf{q}} + \dot{\boldsymbol{\Phi}}_t = \mathbf{0} \quad \Rightarrow \quad \boldsymbol{\Phi}_{\mathbf{q}}\ddot{\mathbf{q}} = -\dot{\boldsymbol{\Phi}}_t - \dot{\boldsymbol{\Phi}}_{\mathbf{q}}\dot{\mathbf{q}} \tag{3.40}$$

Again, this problem is linear, so the unknown values of $\ddot{\mathbf{q}}$ are obtained from the previous equation by solving the linear system.

3.3.2. Index-3 augmented Lagrangian formulation with position and velocity projections

One efficient formulation to solve the dynamics of a multibody system is the Index-3 augmented Lagrangian with velocity and acceleration projections. This formulation, thoroughly explained in [57], has shown its efficiency and robustness in several previous works [58, 59, 60].

The equations of a multibody system in their most general form can be expressed as a system of index-3 differential-algebraic equations (DAE):

$$\mathbf{M}\ddot{\mathbf{q}} + \boldsymbol{\Phi}_{\mathbf{q}}{}^{\top}\boldsymbol{\lambda} = \mathbf{Q} \tag{3.41a}$$

$$\boldsymbol{\Phi} = \mathbf{0} \tag{3.41b}$$

where $\mathbf{M}$ is the mass matrix, $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers, and $\mathbf{Q}$ is the vector of external forces. The term $-\boldsymbol{\Phi}_{\mathbf{q}}{}^{\top}\boldsymbol{\lambda}$ represents the reaction forces.

To solve this system of equations, the augmented Lagrangian method is employed, leading to:

$$\mathbf{M}\ddot{\mathbf{q}} + \boldsymbol{\Phi}_{\mathbf{q}}{}^{\top}\boldsymbol{\alpha}\boldsymbol{\Phi} + \boldsymbol{\Phi}_{\mathbf{q}}{}^{\top}\boldsymbol{\lambda}^{*} = \mathbf{Q} \tag{3.42}$$

being $\boldsymbol{\alpha}$ a diagonal matrix with penalty coefficients. If can be substituted by an scalar if all the constraints have the same penalty coefficient. The Lagrange multipliers are updated following the next iterative process within every time step:

$$\boldsymbol{\lambda}^{*}_{k,i+1} = \boldsymbol{\lambda}^{*}_{k,i} + \boldsymbol{\alpha}\boldsymbol{\Phi}_{k,i+1} \tag{3.43}$$

where the subindex k refers to the time step, while the subidex i refers to the iteration within a given time step. This method is usually implemented with an implicit integrator, such as Newmark, HHT, and Generalized-α algorithms [61].

The integration scheme used in this work is the trapezoidal rule, which is a particular case of all the aforementioned integrators.

$$\dot{\mathbf{q}}_{k+1} = \frac{2}{\Delta t}\mathbf{q}_{k+1} + \dot{\mathbf{q}}_k^* \text{ , where } \dot{\mathbf{q}}_k^* = -\left(\frac{2}{\Delta t}\mathbf{q}_k + \dot{\mathbf{q}}_k\right) \tag{3.44}$$

$$\ddot{\mathbf{q}}_{k+1} = \frac{4}{\Delta t^2}\mathbf{q}_{k+1} + \ddot{\mathbf{q}}_k^* \text{ , where } \ddot{\mathbf{q}}_k^* = -\left(\frac{4}{\Delta t^2}\mathbf{q}_k + \frac{4}{\Delta t}\dot{\mathbf{q}}_k + \ddot{\mathbf{q}}_k\right) \tag{3.45}$$

Substituting Eqs. (3.44) and (3.45) into Eq. (3.42) and scaling by a factor of $\frac{\Delta t^2}{4}$, a nonlinear system of equations for $\mathbf{q}_{k+1}$ is obtained, thus yielding:

$$\mathbf{M}\mathbf{q}_{k+1} + \frac{\Delta t^2}{4}\boldsymbol{\Phi}_{\mathbf{q}_{k+1}}^{\top}\left(\boldsymbol{\alpha}\boldsymbol{\Phi}_{k+1} + \boldsymbol{\lambda}_{k+1}\right) - \mathbf{Q}_{k-1} + \frac{\Delta t^2}{4}\mathbf{M}\ddot{\mathbf{q}}_k^* = \mathbf{0} \tag{3.46}$$

If we denote this system as $\mathbf{f}\left(\mathbf{q}_{k+1}\right) = \mathbf{0}$, its solution by means of the Newton-Raphson iteration can be written as follows:

$$\left.\frac{\partial \mathbf{f}\left(\mathbf{q}\right)}{\partial \mathbf{q}}\right|_{\mathbf{q}=\mathbf{q}_{k+1,i}}\left(\mathbf{q}_{k+1,i+1} - \mathbf{q}_{k+1,i}\right) = -\mathbf{f}\left(\mathbf{q}_{k+1,i}\right) \tag{3.47}$$

where

$$\mathbf{f}\left(\mathbf{q}\right) = \frac{\Delta t^2}{4}\left(\mathbf{M}\ddot{\mathbf{q}} + \boldsymbol{\Phi}_{\mathbf{q}}{}^{\top}\boldsymbol{\alpha}\boldsymbol{\Phi} + \boldsymbol{\Phi}_{\mathbf{q}}{}^{\top}\boldsymbol{\lambda}^* - \mathbf{Q}\right) \tag{3.48}$$

and the approximated tangent matrix is:

$$\frac{\partial \mathbf{f}\left(\mathbf{q}\right)}{\partial \mathbf{q}} \simeq \mathbf{M} + \frac{\Delta t}{2}\bar{\mathbf{C}} + \frac{\Delta t^2}{4}\left(\boldsymbol{\Phi}_{\mathbf{q}}{}^{\top}\boldsymbol{\alpha}\boldsymbol{\Phi}_{\mathbf{q}} + \bar{\mathbf{K}}\right) \tag{3.49}$$

being $\bar{\mathbf{C}}$ and $\bar{\mathbf{K}}$ the damping and stiffness matrices, respectively.

The iteration to solve the integration is also employed to update the Lagrange multipliers in Eq. (3.43), thus saving computational cost.

Once the integration has converged, the constraints at position level are fulfilled, but the constraints at velocity and acceleration level are not expected to be satisfied, since the conditions $\dot{\boldsymbol{\Phi}} = \mathbf{0}$ and $\ddot{\boldsymbol{\Phi}} = \mathbf{0}$ were not imposed. Hence, velocity and acceleration projections are performed. Denoting $\tilde{\dot{\mathbf{q}}}$ and $\tilde{\ddot{\mathbf{q}}}$ to the velocities and accelerations obtained from the integration process, the projected velocities are obtained as:

$$\left[\mathbf{M} + \frac{\Delta t}{2}\bar{\mathbf{C}} + \frac{\Delta t^2}{4}\left(\boldsymbol{\Phi}_{\mathbf{q}}{}^{\top}\boldsymbol{\alpha}\boldsymbol{\Phi}_{\mathbf{q}} + \bar{\mathbf{K}}\right)\right]\dot{\mathbf{q}} = \left[\mathbf{M} + \frac{\Delta t}{2}\bar{\mathbf{C}} + \frac{\Delta t^2}{4}\bar{\mathbf{K}}\right]\tilde{\dot{\mathbf{q}}} - \frac{\Delta t^2}{4}\boldsymbol{\Phi}_{\mathbf{q}}{}^{\top}\boldsymbol{\alpha}\boldsymbol{\Phi}_t \tag{3.50}$$

The projected accelerations result:

$$\left[\mathbf{M} + \frac{\Delta t}{2}\bar{\mathbf{C}} + \frac{\Delta t^2}{4}\left(\boldsymbol{\Phi}_{\mathbf{q}}{}^{\top}\boldsymbol{\alpha}\boldsymbol{\Phi}_{\mathbf{q}} + \bar{\mathbf{K}}\right)\right]\ddot{\mathbf{q}} = \left[\mathbf{M} + \frac{\Delta t}{2}\bar{\mathbf{C}} + \frac{\Delta t^2}{4}\bar{\mathbf{K}}\right]\tilde{\ddot{\mathbf{q}}} - \frac{\Delta t^2}{4}\boldsymbol{\Phi}_{\mathbf{q}}{}^{\top}\boldsymbol{\alpha}\left(\dot{\boldsymbol{\Phi}}_{\mathbf{q}}\dot{\mathbf{q}} + \dot{\boldsymbol{\Phi}}_t\right) \tag{3.51}$$

The system matrix is the same in both velocity and acceleration projections, and hence it can be factorized only once to improve the efficiency of the method.

3.3.3. Matrix R formulation

This method, described in [31], is based on a projection matrix, called matrix R. First, a constant matrix projection matrix $\mathbf{D}$ is considered. The independent velocities $\dot{\mathbf{z}}$ are given by the projection of the dependent velocities $\dot{\mathbf{q}}$ over the rows of this matrix:

$$\dot{\mathbf{z}} = \mathbf{D}\dot{\mathbf{q}} \tag{3.52}$$

Writing Eqs. (3.39) and (3.52) together in matrix form yields:

$$\begin{bmatrix} \mathbf{\Phi}_{\mathbf{q}} \\ \mathbf{D} \end{bmatrix} \dot{\mathbf{q}} = \begin{bmatrix} -\mathbf{\Phi}_t \\ \dot{\mathbf{z}} \end{bmatrix} \tag{3.53}$$

The rows of the $\mathbf{D}$ matrix are linearly independent from the rows of the Jacobian matrix of the constraints $\mathbf{\Phi}_{\mathbf{q}}$. Hence, the matrix on the left-hand side (LHS) of the equation can be inverted, leading to:

$$\dot{\mathbf{q}} = \begin{bmatrix} \mathbf{\Phi}_{\mathbf{q}} \\ \mathbf{D} \end{bmatrix}^{-1} \begin{bmatrix} -\mathbf{\Phi}_t \\ \dot{\mathbf{z}} \end{bmatrix} \equiv \begin{bmatrix} \mathbf{S} & \mathbf{R} \end{bmatrix} \begin{bmatrix} -\mathbf{\Phi}_t \\ \dot{\mathbf{z}} \end{bmatrix} = -\mathbf{S}\mathbf{\Phi}_t + \mathbf{R}\dot{\mathbf{z}} \tag{3.54}$$

where the term $\mathbf{R}\dot{\mathbf{z}}$ represents the general solutions of the homogeneous velocity equation, and the term $-\mathbf{S}\mathbf{\Phi}_t$ represents a particular solution of the complete equation.

Differentiating Eq. (3.53) with respect to time, and rearranging the terms yields:

$$\begin{bmatrix} \mathbf{\Phi}_{\mathbf{q}} \\ \mathbf{D} \end{bmatrix} \ddot{\mathbf{q}} = \begin{bmatrix} -\dot{\mathbf{\Phi}}_t - \dot{\mathbf{\Phi}}_{\mathbf{q}}\dot{\mathbf{q}} \\ \ddot{\mathbf{z}} \end{bmatrix} \tag{3.55}$$

Solving for $\ddot{\mathbf{q}}$ and introducing the matrices $\mathbf{S}$ and $\mathbf{R}$ defined in Eq. (3.54),

$$\ddot{\mathbf{q}} = \begin{bmatrix} \mathbf{\Phi}_{\mathbf{q}} \\ \mathbf{D} \end{bmatrix}^{-1} \begin{bmatrix} -\dot{\mathbf{\Phi}}_t - \dot{\mathbf{\Phi}}_{\mathbf{q}}\dot{\mathbf{q}} \\ \ddot{\mathbf{z}} \end{bmatrix} = \mathbf{S}\left(-\dot{\mathbf{\Phi}}_t - \dot{\mathbf{\Phi}}_{\mathbf{q}}\dot{\mathbf{q}}\right) + \mathbf{R}\ddot{\mathbf{z}} \tag{3.56}$$

Substituting Eq. (3.56) in Eq. (3.41a), and premultiplying by $\mathbf{R}^\top$,

$$\mathbf{R}^\top \left\{\mathbf{M}\left[\mathbf{S}\left(-\dot{\mathbf{\Phi}}_t - \dot{\mathbf{\Phi}}_{\mathbf{q}}\dot{\mathbf{q}}\right) + \mathbf{R}\ddot{\mathbf{z}}\right] + \mathbf{\Phi}_{\mathbf{q}}{}^\top \boldsymbol{\lambda}\right\} = \mathbf{R}^\top\mathbf{Q} \tag{3.57}$$

Rearranging, and keeping in mind that the matrix $\mathbf{R}$ is the orthogonal complement of the Jacobian of the constraints $\mathbf{\Phi}_{\mathbf{q}}$, hence $\mathbf{\Phi}_{\mathbf{q}}\mathbf{R} = \mathbf{0}$:

$$\mathbf{R}^\top\mathbf{M}\mathbf{R}\ddot{\mathbf{z}} = \mathbf{R}^\top\mathbf{M}\mathbf{S}\left(\dot{\mathbf{\Phi}}_t + \dot{\mathbf{\Phi}}_{\mathbf{q}}\dot{\mathbf{q}}\right) + \mathbf{R}^\top\mathbf{Q} \tag{3.58}$$

$$\mathbf{R}^\top\mathbf{M}\mathbf{R}\ddot{\mathbf{z}} = \mathbf{R}^\top\left[\mathbf{Q} + \mathbf{M}\mathbf{S}\left(\dot{\mathbf{\Phi}}_t + \dot{\mathbf{\Phi}}_{\mathbf{q}}\dot{\mathbf{q}}\right)\right] \tag{3.59}$$

For scleronomous systems, $\mathbf{\Phi}_t = \mathbf{0}$, thus some simplifications can be introduced:

$$\dot{\mathbf{q}} = \mathbf{R}\dot{\mathbf{z}} \tag{3.60}$$

$$\ddot{\mathbf{q}} = \dot{\mathbf{R}}\dot{\mathbf{z}} + \mathbf{R}\ddot{\mathbf{z}} \tag{3.61}$$

leading to the following equation of motion:

$$\mathbf{R}^{\top}\mathbf{M}\mathbf{R}\ddot{\mathbf{z}} = \mathbf{R}^{\top}\left[\mathbf{Q} - \mathbf{M}\dot{\mathbf{R}}\dot{\mathbf{z}}\right] \tag{3.62}$$

The matrix $\mathbf{R}$ can be calculated symbolically in some simple cases, but it is usually calculated numerically at every time step from a velocity analysis: the i^{th} column of $\mathbf{R}$ is the vector of dependent velocities when the i^{th} degree of freedom has a unit velocity and all the others null velocity.

The term $-\mathbf{S}\left(\dot{\mathbf{\Phi}}_t + \dot{\mathbf{\Phi}}_{\mathbf{q}}\dot{\mathbf{q}}\right)$, which becomes $\dot{\mathbf{R}}\dot{\mathbf{z}}$ in scleronomous systems, represents the vector of velocity-dependent accelerations, hence it can be calculated numerically by doing an acceleration analysis in which all the degrees of freedom have a null acceleration.

3.4. Multibody-based state observers

In this section, the design of different multibody-based state observers is covered. The different options include continuous and discrete-time versions of the filter, methods in dependent and independent coordinates, and an indirect formulation. A summary of the methods studied in this thesis is presented in Table 3.1.

Method	Classification	State vector	Integrator
CEKF[a]	Continuous-time EKF	Indep. coord. and vel.	Trap. rule
DEKF[b]	Discrete-time EKF	Indep. coord. and vel.	Forward Euler
UKF[c]	Unscented KF	Indep. coord. and vel.	Any
SCKF[d]	Discrete-time EKF, iterated, with 'perfect' measurements	Dep. coord. and vel.	Forward Euler
DIEKFpm[d]	Discrete-time EKF, iterated, with 'perfect' measurements	Dep. coord. and vel.	Forward Euler
errorEKF[e]	Discrete-time EKF, indirect formulation	Error in indep. coord. and vel.	Any for the MBS, Forward Euler for the KF

[a] First presented in [1].
[b] First presented in [44].
[c] Similar to the UKF presented in [36], but with different states. First presented in [44] in its current form.
[d] First presented in [41].
[e] First presented in [43].

Table 3.1: Summary of the methods.

It is worth noticing that all the methods are implemented in a digital computer, thus all of them have to be discretized at some point. The difference between the continuous and the discrete-time methods relies in the moment to do the discretization: while in the former the equations of motion of the multibody model are combined with the equations of the continuous-time Kalman filter, and then integrated together, in the discrete-time methods, the equations of motion are integrated first, and then introduced into a discrete-time Kalman filter.

Multibody dynamics and Kalman filters are combined here aiming to systematize the design of state observers, which are usually based on simplified mathematical models expressly developed to be the plant of the state observers. Multibody dynamics provides a consistent framework to design both the transition and the measurement models of the state observers.

3.4.1. Continuous-time extended Kalman filter (CEKF)

This formulation is based on the one developed in [1] but adapted to deal with multirate (i.e. some or all the sensor information might not be available at every time step). The main idea under this formulation is to adapt the multibody equations in order to fit the Kalman filter structure. In its most basic form, the dynamics of a multibody system is described by the constrained Lagrangian equations, Eqs. (3.41a) and (3.41b).

As the multibody equations are expressed in the form of continuous-time differential equations, it seems natural to adopt the continuous-time version of the Kalman filter. The multibody formalism employed is the matrix R formulation [31], already described in Section 3.3.3. The acceleration of the independent coordinates can be calculated from Eq. (3.59):

$$\ddot{\mathbf{z}} = \left(\mathbf{R}^\top \mathbf{M}\mathbf{R}\right)^{-1} \left\{\mathbf{R}^\top \left[\mathbf{Q} + \mathbf{M}\mathbf{S}\left(\dot{\mathbf{\Phi}}_t + \dot{\mathbf{\Phi}}_{\mathbf{q}}\dot{\mathbf{q}}\right)\right]\right\} \equiv \bar{\mathbf{M}}^{-1}\bar{\mathbf{Q}} \tag{3.63}$$

If now the filter state is defined as the vector $\mathbf{x}^\top = \left[\mathbf{z}^\top, \dot{\mathbf{z}}^\top\right]$, containing the vectors of independent positions $\mathbf{z}$ and velocities $\dot{\mathbf{z}}$, it turns out that:

$$\begin{bmatrix} \hat{\dot{\mathbf{z}}} \\ \hat{\ddot{\mathbf{z}}} \end{bmatrix} = \begin{bmatrix} \hat{\dot{\mathbf{z}}} \\ \bar{\mathbf{M}}^{-1}\bar{\mathbf{Q}} \end{bmatrix} \quad \Rightarrow \quad \hat{\dot{\mathbf{x}}} = \mathfrak{f}\left(\hat{\mathbf{x}}\right) \tag{3.64}$$

where the hat ˆ denotes estimated magnitudes. These equations perfectly fit the continuous extended Kalman filter equation, so they can be straightforwardly applied. In particular, the state-space transition matrix is obtained as the linearization:

$$\mathfrak{F} = \frac{\partial \mathfrak{f}}{\partial \mathbf{x}} = \begin{bmatrix} \mathbf{0}_{g\times g} & \mathbf{I}_g \\ \dfrac{\partial\left(\bar{\mathbf{M}}^{-1}\bar{\mathbf{Q}}\right)}{\partial \mathbf{z}} & \dfrac{\partial\left(\bar{\mathbf{M}}^{-1}\bar{\mathbf{Q}}\right)}{\partial \dot{\mathbf{z}}} \end{bmatrix} \tag{3.65}$$

which can be approximated by:

$$\mathfrak{F} \simeq \begin{bmatrix} \mathbf{0}_{g\times g} & \mathbf{I}_g \\ \mathfrak{F}_{21} & \mathfrak{F}_{22} \end{bmatrix} \tag{3.66}$$

$$\mathfrak{F}_{21} = -\bar{\mathbf{M}}^{-1}\mathbf{R}^\top\left(\bar{\mathbf{K}}\mathbf{R} + 2\mathbf{R}_{\mathbf{q}}\mathbf{R}\ddot{\mathbf{z}}\right) \tag{3.67}$$

$$\mathfrak{F}_{22} = -\bar{\mathbf{M}}^{-1}\mathbf{R}^\top(\bar{\mathbf{C}}\mathbf{R} + \mathbf{M}\dot{\mathbf{R}}) \tag{3.68}$$

where $\bar{\mathbf{K}}$ and $\bar{\mathbf{C}}$ are the stiffness and damping matrices, respectively. In this case the size of the problem is $2g$, being g the number of degrees of freedom of the mechanism. Next, the CEKF correction stage is introduced [47], which fuses the sensor information into the filter, leading to:

$$\dot{\mathbf{z}} - \hat{\dot{\mathbf{z}}} + \mathbf{K}^{\mathbf{z}}(\mathbf{h}(\mathbf{x}) - \mathbf{o}) = \mathbf{0}_{g\times 1} \tag{3.69}$$

$$\bar{\mathbf{M}}\ddot{\mathbf{z}} - \bar{\mathbf{Q}} + \bar{\mathbf{M}}\mathbf{K}^{\dot{\mathbf{z}}}(\mathbf{h}(\mathbf{x}) - \mathbf{o}) = \mathbf{0}_{g\times 1} \tag{3.70}$$

where $\mathbf{K}^{\mathbf{z}}$ and $\mathbf{K}^{\dot{\mathbf{z}}}$ are the parts of the Kalman gain matrix correspondent to $\mathbf{z}$ and $\dot{\mathbf{z}}$, respectively, $\mathbf{h}(\mathbf{x})$ is the measurement model, and $\mathbf{o}$ is the vector of measurements from sensors. The Kalman gain $\mathbf{K}$ is calculated as follows:

$$\mathbf{K} = \mathbf{P}\mathbf{h_x}^\top\mathbf{\Sigma}^S \tag{3.71}$$

where $\mathbf{P}$ is the covariance matrix of the state estimation uncertainty, $\mathbf{h_x}$ is the Jacobian of the measurement model $\mathbf{h}(\mathbf{x})$ with respect to the states $\mathbf{x}$, and $\mathbf{\Sigma}^S$ is the covariance matrix of the measurement noise. The covariance matrix $\mathbf{P}$ evolves following the next equation when the measurements are available:

$$\dot{\mathbf{P}} = \mathfrak{F}\mathbf{P} + \mathbf{P}\mathfrak{F}^\top - \mathbf{P}\mathbf{h_x}{}^\top \mathbf{\Sigma}^S \mathbf{h_x}\mathbf{P} + \mathbf{\Sigma}_c^P \tag{3.72}$$

being $\mathbf{\Sigma}_c^P$ the continuous covariance matrix of the plant noise. But if the measurements are not available, the part related to the them must be eliminated, leading to:

$$\dot{\mathbf{P}} = \mathfrak{F}\mathbf{P} + \mathbf{P}\mathfrak{F}^\top + \mathbf{\Sigma}_c^P \tag{3.73}$$

Moreover, when the measurements are not available, the innovation $\mathbf{h}(\mathbf{x}) - \mathbf{o}$ should be set to $\mathbf{0}$ in Eq. (3.69).

In order to integrate the result of the filter numerically, the implicit single-step trapezoidal rule has been selected as integrator:

$$\begin{aligned}
\dot{\hat{\mathbf{z}}}_{k+1} &= \frac{2}{\Delta t}\hat{\mathbf{z}}_{k+1} - \left(\frac{2}{\Delta t}\hat{\mathbf{z}}_k + \dot{\hat{\mathbf{z}}}_k\right) && (3.74)\\
\ddot{\hat{\mathbf{z}}}_{k+1} &= \frac{2}{\Delta t}\dot{\hat{\mathbf{z}}}_{k+1} - \left(\frac{2}{\Delta t}\dot{\hat{\mathbf{z}}}_k + \ddot{\hat{\mathbf{z}}}_k\right) && (3.75)
\end{aligned}$$

Combining Eq. (3.69) and Eq. (3.74) leads to the following nonlinear system,

$$\left\{\begin{array}{l} \mathbf{g}_1(\hat{\mathbf{x}}_{k+1}) = \mathbf{0}_{g\times 1} \\ \mathbf{g}_2(\hat{\mathbf{x}}_{k+1}) = \mathbf{0}_{g\times 1} \end{array}\right. \quad \Rightarrow \quad \mathbf{g}(\hat{\mathbf{x}}_{k+1}) = \mathbf{0}_{2g\times 1} \tag{3.76}$$

This system can be iteratively solved, e.g. by means of the Newton-Raphson method, employing the following approximate Jacobian matrix:

$$\frac{\partial \mathbf{g}}{\partial \mathbf{x}} = \left[\begin{array}{cc} \frac{2}{\Delta t}\mathbf{I}_g & -\mathbf{I}_g \\ \mathbf{R}^\top \bar{\mathbf{K}}\mathbf{R} & \mathbf{R}^\top(\bar{\mathbf{C}}\mathbf{R} + \mathbf{M}\dot{\mathbf{R}}) + \frac{2}{\Delta t}\bar{\mathbf{M}} \end{array}\right] + \left[\begin{array}{cc} \mathbf{K}^{\mathbf{z}}\mathbf{h_z} & \mathbf{K}^{\mathbf{z}}\mathbf{h_{\dot{z}}} \\ \bar{\mathbf{M}}\mathbf{K}^{\dot{\mathbf{z}}}\mathbf{h_z} & \bar{\mathbf{M}}\mathbf{K}^{\dot{\mathbf{z}}}\mathbf{h_{\dot{z}}} \end{array}\right] \tag{3.77}$$

where the pair $\mathbf{h_z}$ and $\mathbf{h_{\dot{z}}}$ are the position and velocity parts of the Jacobian of the measurement model.

3.4.2. Discrete-time extended Kalman filter (DEKF)

This is the discrete-time version of CEKF described above. A key difference between CEKF and the rest of estimators described from now on, which work in discrete time steps, is that the filter formulation consists of two separated stages: state transition (also called prediction or time update) and state update (also called state correction or measurement update). The former relies on the transition model of the system (integration of dynamical equations) while the latter includes the information from sensors, or observations – this is in contrast to the CEKF, where both stages are seamlessly fused together.

Starting with the prediction stage, the EKF equations in their most generic form are:

$$\begin{aligned} \hat{\mathbf{x}}_k^- &= \mathbf{f}(\hat{\mathbf{x}}_{k-1}^+) && (3.78) \\ \mathbf{P}_k^- &= \mathbf{f}_{\mathbf{x}k-1}\mathbf{P}_{k-1}^+\mathbf{f}_{\mathbf{x}k-1}^\top + \mathbf{\Sigma}^P && (3.79) \end{aligned}$$

where $\mathbf{f}(\cdot)$ stands for the transition model of the system, and $\mathbf{f}_\mathbf{x}$ is its Jacobian matrix with respect to the state $\mathbf{x}$. By considering now the state vector of a MBS estimator in independent coordinates, $\mathbf{x}^\top = \left[\mathbf{z}^\top, \dot{\mathbf{z}}^\top\right]$, and assuming the usage of the Euler method for numerical integration of the multibody equations with time step Δt, the transition model $\mathbf{f}(\cdot)$ required by the EKF results:

$$\hat{\mathbf{x}}_k^- = \mathbf{f}(\hat{\mathbf{x}}_{k-1}^+) \quad \Rightarrow \quad \begin{bmatrix} \hat{\mathbf{z}}_k \\ \hat{\dot{\mathbf{z}}}_k \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{z}}_{k-1} + \Delta t \hat{\dot{\mathbf{z}}}_{k-1} \\ \hat{\dot{\mathbf{z}}}_{k-1} + \Delta t \ddot{\mathbf{z}}_{k-1} \end{bmatrix} \tag{3.80}$$

where the acceleration vector $\ddot{\mathbf{z}}_{k-1}$ is computed by solving the multibody equations of motions as in Eq. (3.63). Thus, it follows that the transition model Jacobian $\mathbf{f}_\mathbf{x}$ has a fairly simple structure:

$$\mathbf{f}_\mathbf{x} \equiv \frac{\partial \mathbf{f}}{\partial \hat{\mathbf{x}}} = \frac{\partial}{\partial \{\hat{\mathbf{z}}, \hat{\dot{\mathbf{z}}}\}} \begin{bmatrix} \hat{\mathbf{z}} + \Delta t \hat{\dot{\mathbf{z}}} \\ \hat{\dot{\mathbf{z}}} + \Delta t \ddot{\mathbf{z}} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_g & \Delta t \mathbf{I}_g \\ \mathbf{0}_{g \times g} & \mathbf{I}_g \end{bmatrix} \tag{3.81}$$

Regarding the discrete plant covariance matrix $\mathbf{\Sigma}^P$ appearing in Eq. (3.78), it stands for the additional uncertainty of the new state $\hat{\mathbf{x}}_k$, physically attributable to unmodeled forces and errors in the parametrization of the mechanism (e.g. bars lengths, inertia values, etc.).

The second stage of the DEKF method, the update, incorporates the sensor readings (when available) to improve the estimate:

$$\begin{aligned} \tilde{\mathbf{y}}_k &= \mathbf{o}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-) && (3.82) \\ \mathbf{\Sigma}_k &= \mathbf{h}_{\mathbf{x}k}\mathbf{P}_k^-\mathbf{h}_{\mathbf{x}k}^\top + \mathbf{\Sigma^S}_k && (3.83) \\ \mathbf{K}_k &= \mathbf{P}_k^-\mathbf{h}_{\mathbf{x}k}^\top\mathbf{\Sigma}_k^{-1} && (3.84) \\ \hat{\mathbf{x}}_k^+ &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k\tilde{\mathbf{y}}_k && (3.85) \\ \mathbf{P}_k^+ &= (\mathbf{I}_{2g} - \mathbf{K}_k\mathbf{h}_{\mathbf{x}k})\mathbf{P}_k^- && (3.86) \end{aligned}$$

where $\tilde{\mathbf{y}}_k$ in Eq. (3.82) is the error or mismatch (often called innovation) between the expected sensor readings and their actual values. The innovation covariance matrix $\mathbf{\Sigma}_k$ in Eq. (3.83) represents the uncertainty in the system state projected via the sensor function ($\mathbf{h}_{\mathbf{x}k}\mathbf{P}_k^-\mathbf{h}_{\mathbf{x}k}^\top$) plus an additional Gaussian noise originated at the sensor itself ($\mathbf{\Sigma^S}_k$). Small values of $\mathbf{\Sigma}_k$ mean that the observation introduces useful information to constrain the estimation of the system state. By evaluating the Kalman gain ($\mathbf{K}_k$) the estimation of the mean and covariance are updated in Eq. (3.85) and Eq. (3.86) respectively.

3.4.3. Unscented Kalman filter (UKF)

The Unscented Kalman Filter (UKF) [62] is an evolution of the family of Kalman filters that is better suited to cope with strong nonlinearities in the transition and observation models. Moreover it is easier to implement than an extended Kalman filter, since the Jacobians of the state transition and measurement equations are not required. This method, among other sigma-point Kalman filters, was previously applied to multibody models in [36] considering the independent accelerations of the multibody model as the states of the filter. The state vector of the UKF considered here, however, contains the independent coordinates and their velocities, that is, $\hat{\mathbf{x}}^\top = \left[\hat{\mathbf{z}}^\top, \dot{\hat{\mathbf{z}}}^\top\right]$.

The method comprises the same prediction and update stages than the DEKF. The differentiating feature of the UKF is the avoidance of the first order Taylor approximation in the propagation of Gaussian random variables through the transition and observation functions. Instead, the unscented transformation is employed: a set of $2l+1$ samples $\boldsymbol{\chi}(i)$ (usually called sigma-points) are deterministically chosen from the Gaussian distributions, being $l = 2g$ the length of the state vector:

$$\boldsymbol{\chi}_{k-1}(0) = \hat{\mathbf{x}}_{k-1}^{+} \tag{3.87}$$

$$\boldsymbol{\chi}_{k-1}(i) = \hat{\mathbf{x}}_{k-1}^{+} + \zeta\left(\sqrt{\mathbf{P}_{k-1}^{+}}\right)_i, \; i = 1, ..., l \tag{3.88}$$

$$\boldsymbol{\chi}_{k-1}(l+i) = \hat{\mathbf{x}}_{k-1}^{+} - \zeta\left(\sqrt{\mathbf{P}_{k-1}^{+}}\right)_i, \; i = 1, ..., l \tag{3.89}$$

where $\sqrt{\cdot}$ is the matrix square root using the lower triangular matrix of the Cholesky decomposition and $(.)_i$ stands for its i^{th} column, $\zeta = \sqrt{l+\lambda}$, $\lambda = \alpha^2(l+\kappa)$, α and κ are user-defined tuning parameters, with $0 < \alpha \leq 1$ and κ is usually set to 0. Then, these samples are transformed via the corresponding function (in this case, an integration step of the multibody simulation):

$$\boldsymbol{\chi}_k(i) = \mathbf{f}(\boldsymbol{\chi}_{k-1}(i)) \tag{3.90}$$

Both the forward Euler and the trapezoidal rule integration methods were considered in this work. Next, the mean and covariance of the resulting set are calculated:

$$\hat{\mathbf{x}}_k^{-} = \sum_{i=0}^{2l+1} W_i^m \boldsymbol{\chi}_k(i) \tag{3.91}$$

$$\mathbf{P}_k^{-} = \sum_{i=0}^{2l+1} W_i^c \left(\boldsymbol{\chi}_k(i) - \hat{\mathbf{x}}_k^{-}\right)\left(\boldsymbol{\chi}_k(i) - \hat{\mathbf{x}}_k^{-}\right)^\top + \boldsymbol{\Sigma}^P \tag{3.92}$$

where $W_0^m = \lambda/(l+\lambda)$, $W_0^c = W_0^m + (1-\alpha^2+\beta)$, $W_0^c = W_0^m = 1/[2(l+\lambda)]$, being β a secondary scaling factor used to emphasize the weighting on the zeroth sigma-point for the covariance calculation. The selection of the weights W_i^m and W_i^c has been done according to the rules proposed in [63].

After the prediction stage, the update is accomplished. A new set of sigma points can be generated, or the existent sigma points $\boldsymbol{\chi}_k(i)$ can be reused to save computational effort at the cost of sacrificing accuracy. This last option was used here. The measurement equation is applied to the samples, and the mean and covariance of the measurements are calculated with the same method applied in the prediction:

$$\boldsymbol{\mathcal{Y}}_k(i) = \mathbf{o}_k - \mathbf{h}\left(\boldsymbol{\chi}_k(i)\right) \tag{3.93}$$

$$\begin{aligned}\tilde{\mathbf{y}}_k &= \sum_{i=0}^{2l+1} W_i^m \boldsymbol{\mathcal{Y}}_k(i) && (3.94)\\ \mathbf{P}_{\tilde{\mathbf{y}}} &= \sum_{i=0}^{2l+1} W_i^c \boldsymbol{\mathcal{Y}}_k(i)\boldsymbol{\mathcal{Y}}_k(i)^\top + \boldsymbol{\Sigma}^S{}_k && (3.95)\end{aligned}$$

Next, the Kalman gain matrix is calculated and employed to correct state and covariance matrix:

$$\begin{aligned}\mathbf{P}_{\mathbf{x}\tilde{\mathbf{y}}} &= \sum_{i=0}^{2l+1} W_i^c \left(\boldsymbol{\chi}_k(i) - \hat{\mathbf{x}}_k^-\right)\left(-\boldsymbol{\mathcal{Y}}_k(i)\right)^\top + \boldsymbol{\Sigma}^S{}_k && (3.96)\\ \mathbf{K}_k &= \mathbf{P}_{\mathbf{x}\tilde{\mathbf{y}}}\mathbf{P}_{\tilde{\mathbf{y}}}^{-1} && (3.97)\\ \hat{\mathbf{x}}_k^+ &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k\tilde{\mathbf{y}}_k && (3.98)\\ \mathbf{P}_k^+ &= \mathbf{P}_k^- - \mathbf{h}_{\mathbf{x}k}\mathbf{P}_{\tilde{\mathbf{y}}}\mathbf{h}_{\mathbf{x}k} && (3.99)\end{aligned}$$

As shown in [62], this approach captures the correct posterior mean and covariance up to the third order of a Taylor series expansion, in contrast to the first order of DEKF and most other methods. In turn, its computational cost is in general higher than simpler methods.

3.4.4. Smoothly constrained Kalman filter (SCKF)

This filter is the application of the algorithm described in [64] to a multibody model. In this method, the state $\mathbf{x}$ is build with the whole multibody coordinates and velocities vectors $\mathbf{q}$ and $\dot{\mathbf{q}}$, each one of size $n \times 1$. The SCKF transition function is built assuming the forward Euler integrator:

$$\hat{\mathbf{x}}_k^- = \mathbf{f}(\hat{\mathbf{x}}_{k-1}^+) \quad \Rightarrow \begin{bmatrix} \hat{\mathbf{q}}_k \\ \hat{\dot{\mathbf{q}}}_k \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{q}}_{k-1} + \Delta t \hat{\dot{\mathbf{q}}}_{k-1} \\ \hat{\dot{\mathbf{q}}}_{k-1} + \Delta t \ddot{\mathbf{q}}_{k-1} \end{bmatrix} \tag{3.100}$$

where the vector of dependent accelerations $\ddot{\mathbf{q}}_{k-1}$ is calculated from the equation of the dynamics of the system (Eqs. (3.41a) and (3.41b)). Thus, the transition model Jacobian is as follows:

$$\mathbf{f}_\mathbf{x} \equiv \frac{\partial \mathbf{f}}{\partial\{\hat{\mathbf{q}}, \hat{\dot{\mathbf{q}}}\}} = \begin{bmatrix} \mathbf{I}_n & \Delta t \mathbf{I}_n \\ \mathbf{0}_{n\times n} & \mathbf{I}_n \end{bmatrix} \tag{3.101}$$

and the covariance matrix is updated as:

$$\mathbf{P}_k^- = \mathbf{f}_\mathbf{x}\mathbf{P}_{k-1}^+\mathbf{f}_\mathbf{x}{}^\top + \boldsymbol{\Sigma}^P \tag{3.102}$$

After the time update, the measurement update is undertaken, but taking into account only the measurements coming from the sensors:

$$\begin{aligned}\mathbf{K}_{k,0} &= \mathbf{P}^-\mathbf{h}_{\mathbf{x}k}^\top(\mathbf{h}_{\mathbf{x}k}\mathbf{P}^-\mathbf{h}_{\mathbf{x}k}^\top + \boldsymbol{\Sigma}^S)^{-1} && (3.103)\\ \hat{\mathbf{x}}_{k,0}^+ &= \hat{\mathbf{x}}_k^- + \mathbf{K}_{k,0}(\mathbf{o}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-)) && (3.104)\\ \mathbf{P}_{k,0}^+ &= (\mathbf{I}_{2n} - \mathbf{K}_{k,0}\mathbf{h}_{\mathbf{x}k})\mathbf{P}_k^- && (3.105)\end{aligned}$$

Up to now, this is the algorithm of a conventional EKF, but at this moment the states are not expected to fulfill the constraints, so an iterative process is started to impose the position and velocity constraints as if they were additional measurements. Although the constraints are perfect measurements, virtual noise is added to them in order to ease the convergence of the problem. The covariance matrix of the virtual noise added is known as weakening matrix, and it is calculated as follows:

$$\boldsymbol{\xi}_0 = \tilde{\alpha} \begin{bmatrix} \boldsymbol{\Phi}_\mathbf{x} \\ \dot{\boldsymbol{\Phi}}_\mathbf{x} \end{bmatrix} \mathbf{P}^+_{k,0} \begin{bmatrix} \boldsymbol{\Phi}_\mathbf{x} \\ \dot{\boldsymbol{\Phi}}_\mathbf{x} \end{bmatrix}^\top \tag{3.106}$$

where $\tilde{\alpha}$ is a tuning parameter, and $\boldsymbol{\Phi}_\mathbf{x}$ and $\dot{\boldsymbol{\Phi}}_\mathbf{x}$ are the Jacobian matrices of the constraints at position and velocity levels with respect to the current states, respectively. This weakening matrix contains virtual noise to be added to the multibody constraints in order to ease the convergence of the problem. The iterative update is as follows:

$$\mathbf{K}_{k,i} = \mathbf{P}^+_{k,i-1} \begin{bmatrix} \boldsymbol{\Phi}_\mathbf{x} \\ \dot{\boldsymbol{\Phi}}_\mathbf{x} \end{bmatrix} + \left(\begin{bmatrix} \boldsymbol{\Phi}_\mathbf{x} \\ \dot{\boldsymbol{\Phi}}_\mathbf{x} \end{bmatrix} \mathbf{P}^+_{k,i-1} \begin{bmatrix} \boldsymbol{\Phi}_\mathbf{x} \\ \dot{\boldsymbol{\Phi}}_\mathbf{x} \end{bmatrix}^\top + \boldsymbol{\xi}_{i-1} \right)^{-1} \tag{3.107}$$

$$\hat{\mathbf{x}}^+_{k,i+1} = \hat{\mathbf{x}}^+_{k,i} - \mathbf{K}_{k,i} \begin{bmatrix} \boldsymbol{\Phi}(\hat{\mathbf{x}}^+_{k,i}) \\ \dot{\boldsymbol{\Phi}}(\hat{\mathbf{x}}^+_{k,i}) \end{bmatrix} \tag{3.108}$$

$$\mathbf{P}^+_{k,i+1} = \left(\mathbf{I}_{2n} - \mathbf{K}_{k,i} \begin{bmatrix} \boldsymbol{\Phi}_\mathbf{x} \\ \dot{\boldsymbol{\Phi}}_\mathbf{x} \end{bmatrix} \right) \mathbf{P}^+_{k,i} \left(\mathbf{I}_{2n} - \mathbf{K}_{k,i} \begin{bmatrix} \boldsymbol{\Phi}_\mathbf{x} \\ \dot{\boldsymbol{\Phi}}_\mathbf{x} \end{bmatrix} \right)^\top + \mathbf{K}_{k,i} \boldsymbol{\xi}_i \mathbf{K}^\top_{k,i} \tag{3.109}$$

$$\boldsymbol{\xi}_{i+1} = \boldsymbol{\xi}_i e^{-\tilde{\beta}} \tag{3.110}$$

being $\tilde{\beta}$ another tuning parameter. This iterative process is performed until the position constraints $\boldsymbol{\Phi}$, and velocity constraints $\dot{\boldsymbol{\Phi}}$ fit the desired tolerance.

One of the drawbacks of this method is that the measurement model is applied before the constraints are imposed. Hence, depending on the expression of the measurement model, additional errors may arise from its usage when the constraint equations are not fulfilled.

3.4.5. Discrete-time iterated extended Kalman filter with perfect measurements (DIEKFpm)

This method is a expansion of the standard DIEKF [47] to cope with constraints in its state space by employing so-called perfect measurements [65]. The key idea consists of augmenting the vector of observations to include virtual observations that reflect the fulfillment of the kinematics constraints in both position and velocities. The difference with the SCKF is that here both the actual measurements and the perfect measurements are applied simultaneously, and no weakening matrix is employed. Apart from these differences, the idea of this method is quite similar to the SCKF.

The state vector of this estimator comprises the multibody model coordinates and their derivatives, that is, $\mathbf{x}^\top = \left[\mathbf{q}^\top, \dot{\mathbf{q}}^\top\right]$. The augmented observation model

$\mathbf{h}'(\hat{\mathbf{x}})$ is defined as the concatenation of the real sensors $\mathbf{h}(\mathbf{x})$ and the kinematic constraints in position and velocity, such as $\mathbf{h}'(\mathbf{x})^\top = [\mathbf{h}(\mathbf{x})^\top \ \mathbf{\Phi}(\mathbf{x})^\top \ \dot{\mathbf{\Phi}}(\mathbf{x})^\top]$. This affects the calculation of the innovation (or "residual"), which must compare the actual sensor readings and current constraint errors with their predictions. For all time steps k and iteration index i, the predicted values of the constrains are always zero, i.e.

$$\tilde{\mathbf{y}}_{k,i} = \begin{bmatrix} \mathbf{o}_k \\ \mathbf{0}_{2m\times1} \end{bmatrix} - \mathbf{h}'(\hat{\mathbf{x}}_{k,i}) = \begin{bmatrix} \mathbf{o}_k - \mathbf{h}(\hat{\mathbf{x}}_{k,i}) \\ -\mathbf{\Phi}(\hat{\mathbf{x}}_{k,i}) \\ -\dot{\mathbf{\Phi}}(\hat{\mathbf{x}}_{k,i}) \end{bmatrix} \tag{3.111}$$

where m is the length of the constraints vector The adjective "perfect" that names this method comes from the assumption that there is no error source in the virtual observations. However, in practice some small noise is added to the extended sensor covariance matrix to improve the convergence of the filter.

3.4.6. Error-state extended Kalman filter (errorEKF)

A common problem of the previous extended Kalman filter (EKF) methods is that they require a tailor-made multibody algorithm, which is integrated with the Kalman filter algorithm. This problem is overcome by the UKF method, in which any multibody formulation or integrator can be used, but at the cost of running one multibody simulation for each sigma point. However, an error-state Kalman filter (also known as indirect Kalman filter) can combine the efficiency of the EKF while using the multibody simulation as a "black box". This kind of indirect formulation is commonly used in inertial navigation and absolute position sensors fusion algorithms [66].

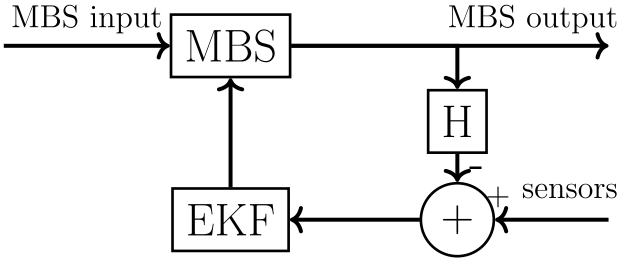


Figure 3.16: Simplified diagram of the error state Kalman filter applied to multibody simulations.

A simplified schematic of this method, coined as errorEKF, is shown in Figure 3.16. It works as follows: after one step of the multibody simulation is performed, the estimation of its error is launched. The state $\mathbf{x}^\top = \left[\Delta\mathbf{z}^\top, \Delta\dot{\mathbf{z}}^\top\right]$ consists of the error in position and velocity of the degrees of freedom of the mechanism, instead of the positions and velocities used in all the previous methods. The propagation phase is performed following the next equations:

$$\hat{\mathbf{x}}_k^- = \mathbf{0} \tag{3.112}$$

$$\mathbf{P}_k^- = \mathbf{f_x}_{k-1}\mathbf{P}_{k-1}^+\mathbf{f_x}_{k-1}^\top + \mathbf{\Sigma}^P \tag{3.113}$$

These equations are the conventional equations for the propagation of the Kalman state, and the transition matrix $\mathbf{f_x}$ is the same used in the DEKF method. However, as the estimated errors are fed back to the multibody simulation, the estimation of the error in the propagation phase is always null, as shown in Eq. (3.112).

The equations for the correction phase of the filter are also similar to the ones found in the DEKF.

$$\begin{aligned}
\tilde{\mathbf{y}}_k &= \mathbf{o}_k - \mathbf{h}(\mathbf{q}_k, \dot{\mathbf{q}}_k) && (3.114)\\
\mathbf{\Sigma}_k &= \mathbf{h_x}_k \mathbf{P}_k^- \mathbf{h_x}_k^\top + \mathbf{\Sigma}^S{}_k && (3.115)\\
\mathbf{K}_k &= \mathbf{P}_k^- \mathbf{h_x}_k^\top \mathbf{\Sigma}_k^{-1} && (3.116)\\
\hat{\mathbf{x}}_k^+ &= \mathbf{0} + \mathbf{K}_k \tilde{\mathbf{y}}_k && (3.117)\\
\mathbf{P}_k^+ &= (\mathbf{I}_{2g} - \mathbf{K}_k \mathbf{h_x}_k)\mathbf{P}_k^- && (3.118)
\end{aligned}$$

The differences can be found in Eq. (3.114), where the virtual measurements $\mathbf{h}(\mathbf{q}_k, \dot{\mathbf{q}}_k)$ are built by using the coordinates of the multibody model. The jacobian of the measurement model $\mathbf{h_x}$ has the same expression as in the DEKF, since the partial derivatives with respect to the errors in the states have the same value than the partial derivatives with respect to the states. Before the measurements are applied, the state is always null, so Eq. (3.117) is modified accordingly.

After the correction stage, the estimation of the position and velocity errors of the independent coordinates, $\Delta\hat{\mathbf{z}}$ and $\Delta\hat{\dot{\mathbf{z}}}$, are obtained. However, to correct the state of the multibody system, the errors for all the coordinates, $\Delta\hat{\mathbf{q}}$ and $\Delta\hat{\dot{\mathbf{q}}}$, must be obtained, so they must be projected over the constraints manifold, as explained hereafter.

An error in position means that an increment to the coordinates should be applied to get the position corrected. Such an increment must fulfill the velocity constraints. Therefore, the increments applied to the coordinates of the mechanism are calculated by solving the velocity problem explained in section 3.3.1, as follows:

$$\mathbf{\Phi_q}\Delta\hat{\mathbf{q}} = \mathbf{0} \quad \Rightarrow \quad \hat{\mathbf{q}} = \mathbf{q} + \Delta\hat{\mathbf{q}} \qquad (3.119)$$

where the values of the degrees of freedom, $\Delta\hat{\mathbf{z}}$, are introduced in their correspondent position in $\Delta\hat{\mathbf{q}}$. This method is an approximation, thus a perfect fulfillment of the constraints at position level is not expected. However, as the corrections are performed every time step, the errors are usually acceptable for most applications. The main advantage of this method is that the velocity problem is linear, so this process is much faster than solving the position problem. If the sensors employed have a low update rate, the position errors may become too big to employ this approximation, and hence the position problem should be solved instead.

The estimated velocities are obtained, again, by solving the velocity problem in which the velocities of the degrees of freedom are the velocities taken from the multibody model plus the velocity errors estimated.

After the corrections are applied to the model, the expected error is $\hat{\mathbf{x}}_k^+ = \mathbf{0}$.

3.5. Sensor models in multibody dynamics

One of the advantages of including a multibody model inside a Kalman filter is that it provides a means to develop in a systematic fashion the measurement sensitivity matrices $\mathbf{h}_{\mathbf{x}}$ needed by the EKF. As all the dependent variables of the multibody model are available to build the model of the sensors, the expressions of such models are usually simple, and their derivatives with respect to the dependent coordinates $\mathbf{q}$ and velocities $\dot{\mathbf{q}}$ are straightforward. However, the filters in independent coordinates require the derivatives with respect with the independent coordinates $\mathbf{z}$ and velocities $\dot{\mathbf{z}}$, which are obtained by applying the chain rule, as shown in Eqs. (3.120) and (3.121). Acceleration sensors are not considered here, since the states of the filters studied in this thesis do not contain accelerations.

$$\mathbf{h}_{\mathbf{z}} \equiv \frac{\partial \mathbf{h}(\mathbf{q},\dot{\mathbf{q}})}{\partial \mathbf{z}} = \mathbf{h}_{\mathbf{q}}\frac{\partial \mathbf{q}}{\partial \mathbf{z}} + \mathbf{h}_{\dot{\mathbf{q}}}\frac{\partial \dot{\mathbf{q}}}{\partial \mathbf{z}} \tag{3.120}$$

$$\mathbf{h}_{\dot{\mathbf{z}}} \equiv \frac{\partial \mathbf{h}(\mathbf{q},\dot{\mathbf{q}})}{\partial \dot{\mathbf{z}}} = \mathbf{h}_{\mathbf{q}}\cancelto{\mathbf{0}}{\frac{\partial \mathbf{q}}{\partial \dot{\mathbf{z}}}} + \mathbf{h}_{\dot{\mathbf{q}}}\frac{\partial \dot{\mathbf{q}}}{\partial \dot{\mathbf{z}}} \tag{3.121}$$

where $\frac{\partial \mathbf{q}}{\partial \mathbf{z}} = \frac{\partial \dot{\mathbf{q}}}{\partial \dot{\mathbf{z}}} = \mathbf{R}$, the projection matrix explained in section 3.3.3. The term $\frac{\partial \dot{\mathbf{q}}}{\partial \mathbf{z}}$ is developed in [67], and reproduced hereafter for the convenience of the reader. Taking the partial derivative of Eq. (3.53) with respect to $\mathbf{q}$ yields:

$$\begin{bmatrix} \mathbf{\Phi}_{\mathbf{q}} \\ \mathbf{D} \end{bmatrix} \frac{\partial \dot{\mathbf{q}}}{\partial \mathbf{q}} + \begin{bmatrix} \mathbf{\Phi}_{\mathbf{qq}} \\ \mathbf{0} \end{bmatrix} \dot{\mathbf{q}} = \begin{bmatrix} -\mathbf{\Phi}_{t\mathbf{q}} \\ \mathbf{0} \end{bmatrix} \tag{3.122}$$

Recalling from Eq. (3.54):

$$\begin{bmatrix} \mathbf{\Phi}_{\mathbf{q}} \\ \mathbf{D} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{S} & \mathbf{R} \end{bmatrix} \tag{3.123}$$

Then, Eq. (3.122) can be rewritten as:

$$\frac{\partial \dot{\mathbf{q}}}{\partial \mathbf{q}} = \begin{bmatrix} \mathbf{\Phi}_{\mathbf{q}} \\ \mathbf{D} \end{bmatrix}^{-1} \begin{bmatrix} -\mathbf{\Phi}_{t\mathbf{q}} - \mathbf{\Phi}_{\mathbf{qq}}\dot{\mathbf{q}} \\ \mathbf{0} \end{bmatrix} \Rightarrow \frac{\partial \dot{\mathbf{q}}}{\partial \mathbf{q}} = -\mathbf{S}\dot{\mathbf{\Phi}}_{\mathbf{q}} \tag{3.124}$$

From this result, the desired term can be obtained:

$$\frac{\partial \dot{\mathbf{q}}}{\partial \mathbf{z}} = \frac{\partial \dot{\mathbf{q}}}{\partial \mathbf{q}}\frac{\partial \mathbf{q}}{\partial \mathbf{z}} = -\mathbf{S}\dot{\mathbf{\Phi}}_{\mathbf{q}}\mathbf{R} \tag{3.125}$$

3.6. Covariance matrices of plant and measurement noise

It is known that, when applying Kalman filters, the tuning of the parameters of the algorithm (covariance matrices of plant and measurement noise) is paramount. Even if everything else is correct in the algorithm, it can become unstable if the covariance matrices of the noise are not properly set.

When working with a simulated plant, the signals from the sensors are built from a multibody model, playing the role of actual mechanism, and then white Gaussian noise is generated and added to the sensors signal. Hence their properties are perfectly known.

The plant noise properties, however, are not known even in this case, since the errors introduced in the simulation are not additive white noise, but errors in the multibody model which produce deviations from the ideal behavior. In a real mechanism is even worst, because it is not know where the errors are.

When the ground truth is not know, the only way to check that the filter is working well is by checking the innovation sequence. It should behave like white noise. Although this criterion does not allow to find the correct absolute value of the covariance matrices, it allows to find the correct relation between plant and measurement covariance matrices.

3.6.1. Structure of plant noise

When dealing with multibody models, usually the geometry is known accurately enough to model it properly. However, getting precise models of the forces and obtaining the actual distribution of mass is usually more complex in practice. Both these effects appear as errors in the acceleration. Then, the integration process and the multibody formulation may introduce more errors, but they are usually negligible compared to the previous ones. For this reason, only the variance of acceleration terms $\sigma_{\ddot{\mathbf{z}}}^2$ is considered in the plant noise. Although each acceleration considered in the plant can have a different variance, they are expressed here as if they were the same to ease the notation. The values must be tuned as indicated in section 3.6.

In the CEKF, which is the only method in continuous time studied in this thesis, the acceleration noise is straightforwardly introduced:

$$\mathbf{\Sigma}_c^P = \begin{bmatrix} \mathbf{0}_{g\times g} & \mathbf{0}_{g\times g} \\ \mathbf{0}_{g\times g} & \sigma_{\ddot{\mathbf{z}}}^2 \mathbf{I}_g \end{bmatrix} \tag{3.126}$$

However, in the discrete-time methods, the matrix of the covariance of the plant noise must be calculated from its continuous counterpart by integration:

$$\mathbf{\Sigma}^P = \mathbf{f}_{\mathbf{x}}(t_k, t_{k-1}) \left[\int_{t_{k-1}}^{t_k} \mathbf{f}_{\mathbf{x}}{}^{-1}(\tau, t_{k-1}) \mathbf{\Sigma}_c^P \mathbf{f}_{\mathbf{x}}{}^{-\top}(\tau, t_{k-1}) d\tau \right] \mathbf{f}_{\mathbf{x}}{}^{\top}(t_k, t_{k-1}) \tag{3.127}$$

This integration can be done using Van Loan's method [68], obtaining a covariance matrix of process noise with the following structure for the methods in independent coordinates:

$$\mathbf{\Sigma}^P = \begin{bmatrix} \sigma_{\ddot{\mathbf{z}}}^2 \frac{\Delta t^3}{3} \mathbf{I}_g & \sigma_{\ddot{\mathbf{z}}}^2 \frac{\Delta t^2}{2} \mathbf{I}_g \\ \sigma_{\ddot{\mathbf{z}}}^2 \frac{\Delta t^2}{2} \mathbf{I}_g & \sigma_{\ddot{\mathbf{z}}}^2 \Delta t \mathbf{I}_g \end{bmatrix} \tag{3.128}$$

In the methods in dependent coordinates, the same structure is applied, but using blocks of size n instead of g.

Chapter 4

Application to planar linkages

The methods presented in Chapter 3 have been implemented in the case of two planar mechanisms: a four-bar and a five-bar linkages. The aim was to discover the level of accuracy and the computational cost of the different methods, as well as the increase of the computational cost with the size of the mechanism, and the performance of the methods with different levels of modeling error, and with different sets of sensors. Both mechanisms are closed loop to highlight all the advantages of combining a multibody simulation with a Kalman filter, although this methodology can be applied also to open loop mechanisms.

4.1. Methodology

All results presented on this chapter rely on simulations, in order to have a ground truth suitable for a fair comparison among the methods. For every test, a multibody simulation was run using a model which was considered as the real mechanism, thus providing the ground truth to verify the results delivered by the observers. This simulation was also employed to build the signals from the sensors. To do this, perfect sensors were modeled, and then pseudo-random noise was added to their measurements. The sequence of the pseudo-random values of the noise was the same for all the tests carried out, enabling a fair comparison among the different methods.

Then, a second multibody model was built, but modifying some of the properties, to simulate modeling error. Usually, the geometry of any machine or vehicle can be known with great accuracy. However, the level of accuracy in the determination of forces and mass distribution is often not so good, both producing acceleration errors. Therefore, the parameter which is intentionally modified is the acceleration of gravity, leading to an erroneous acceleration. Moreover, the initial position of the mechanism is also modified to simulate situations in which it is not exactly known. Finally, the state observer was built using the latter multibody model (the imperfect one), and corrected with the information provided by the noisy sensors built from the simulation of the first multibody model.

All the multibody simulations were run with a time step of 5e-3 s. The sensors considered in the test were position and velocity sensors in different configurations, and their sampling rates were considered from 200 Hz (the frequency of the multibody simulations) to 10 Hz. This range of frequencies has been selected to cover the sampling rate of the most common sensors, from the 200 Hz of many microelectromechanical systems (MEMS) to the 10 Hz of the modern low-cost GPS receivers.

4.2. Multibody model of four-bar linkage

Both mechanisms have been modeled in mixed coordinates, i.e., defining the whole mechanism with natural coordinates, and adding extra coordinates when it was convenient. The four-bar linkage is shown in Figure 4.1. The lengths and weights of the bars can be seen in Table 4.1. The full set of coordinates employed in this work for this mechanism is $\mathbf{q} = \begin{bmatrix} x_1 & y_1 & x_2 & y_2 & \theta \end{bmatrix}^\top$, where the points 1 and 2 are the left and the right ends of the coupler, respectively.

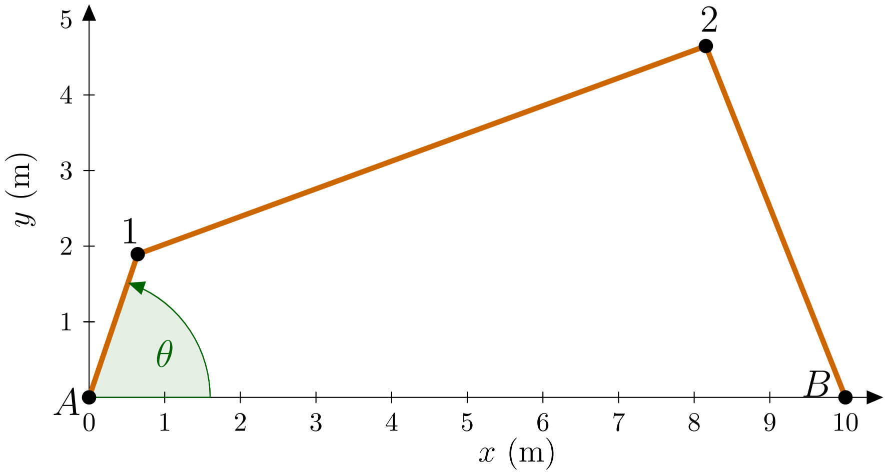


Figure 4.1: Four-bar linkage employed in this thesis.

	Crank	Coupler	Rocker	Ground element
Mass (kg)	2	8	5	—
Length (m)	2	8	5	10

Table 4.1: Properties of the four-bar linkage.

The vector of constraints consists of three constraints imposing the length of the bars and one more to define the angle θ. There are two options for the last constraint, one using the sine, and other using the cosine. Depending on the value of θ, one or the other is used. The vector of constraints $\mathbf{\Phi}$ is as follows:

$$\mathbf{\Phi} = \begin{bmatrix} (x_1 - x_A)^2 + (y_1 - y_A)^2 - L_{A,1}^2 \\ (x_2 - x_1)^2 + (y_2 - y_1)^2 - L_{1,2}^2 \\ (x_B - x_2)^2 + (y_B - y_2)^2 - L_{2,B}^2 \\ f(\theta) \end{bmatrix} \tag{4.1}$$

where $L_{i,j}$ is the length of the bar from point i to point j, and $f(\theta)$ is defined as follows:

$$f(\theta) = \begin{cases} y_1 - y_A - L_{A,1}\sin(\theta) & \text{if } |\sin(\theta)| < 0.7 \\ x_1 - x_A - L_{A,1}\cos(\theta) & \text{if } |\sin(\theta)| \geq 0.7 \end{cases} \tag{4.2}$$

The angle θ was selected as the degree of freedom because it represents the position of the mechanism in all the range of its motion, thus $\mathbf{z} = \begin{bmatrix}\theta\end{bmatrix}$.

4.3. Multibody model of five-bar linkage

The four-bar linkage is shown in Figure 4.2. The lengths and weights of the bars can be seen in Table 4.2. The full set of coordinates employed in this work for this mechanism is $\mathbf{q} = \begin{bmatrix}x_1 & y_1 & x_2 & y_2 & x_3 & y_3 & \theta_1 & \theta_2\end{bmatrix}^\top$. The vector of constraints is similar to that of the four-bar linkage, but with four equations of constant length, and two constraining the value of the angles θ_1 and θ_2. Again, the angles were selected as the degrees of freedom, hence, $\mathbf{z} = \begin{bmatrix}\theta_1 & \theta_2\end{bmatrix}^\top$.

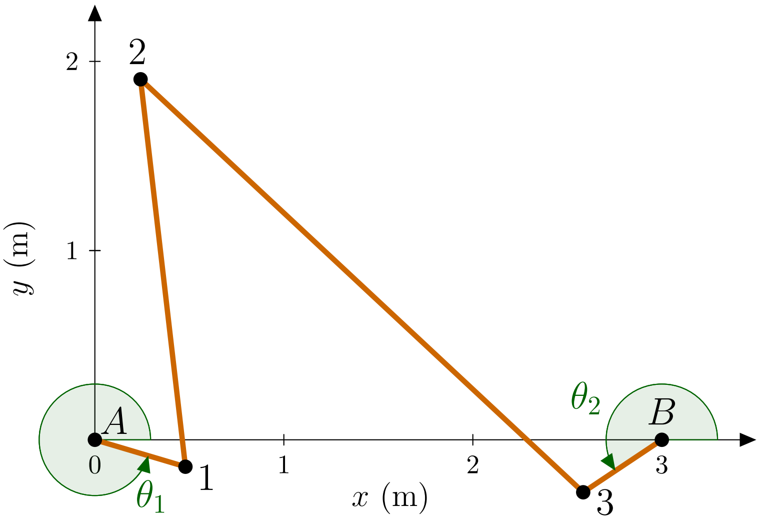


Figure 4.2: Five-bar linkage employed in this thesis.

	Left crank	Left coupler	Right coupler	Right crank	Ground element
Mass (kg)	3	1	2	3	—
Length (m)	0.5	2.062	3.202	0.5	3

Table 4.2: Properties of the five-bar linkage.

4.4. Position sensors

Encoders are measurement devices for measuring angular position. They are commonly used in all kinds of machines when an angular magnitude has to be monitored. Therefore, encoders were selected as position sensors in these mechanism. In the four-bar mechanism, an encoder was considered measuring the angle θ, such that the measurement model is $\mathbf{h}(\mathbf{x}) = [\theta]$. In filters in independent coordinates, whose states are $\mathbf{x} = \begin{bmatrix}\mathbf{z}^\top & \dot{\mathbf{z}}^\top\end{bmatrix}^\top$, the Jacobian of the measurement model with respect the states results as follows:

$$\mathbf{h}_\mathbf{x} = \begin{bmatrix}1 & 0\end{bmatrix} \tag{4.3}$$

This expression is also valid for the errorEKF.

In the methods using the dependent coordinates as states, such that $\mathbf{x} = \begin{bmatrix}\mathbf{q}^\top & \dot{\mathbf{q}}^\top\end{bmatrix}^\top$, the Jacobian of the measurement model with respect to the states is:

$$\mathbf{h}_\mathbf{x} = \begin{bmatrix}0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0\end{bmatrix} \tag{4.4}$$

For the five-bar linkage, the expressions are analogous. The measurement model is $\mathbf{h}(\mathbf{x}) = \begin{bmatrix}\theta_1 & \theta_2\end{bmatrix}^\top$. Its Jacobian matrix in methods in independent coordinates yields:

$$\mathbf{h}_\mathbf{x} = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\end{bmatrix} \tag{4.5}$$

while in dependent coordinates it results:

$$\mathbf{h}_\mathbf{x} = \begin{bmatrix}0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix} \tag{4.6}$$

4.5. Velocity sensors

The state observers developed in this thesis were also tested with sensors providing velocities. In particular, angular rate measurements were considered, since they are also quite common measuring devices. Especially, MEMS gyroscopes are used in a high variety of applications, including cell phones, safety systems in vehicles, autonomous vehicles, robots, etc.

Regarding the tests performed with the planar linkages, two configurations were tested in each mechanism: in one configuration the angular rate sensors were installed on the cranks, while in the other the gyroscopes were installed on the couplers. The measurement models of both systems are detailed hereafter.

The measurement equations with MEMS gyroscopes installed on the cranks, and their Jacobian matrices are similar to those ones resulting from the use of encoders. In the four-bar linkage, the measurement equation is $\mathbf{h}(\mathbf{x}) = \begin{bmatrix}\dot{\theta}\end{bmatrix}$. Its Jacobian matrix with respect to the states in methods in independent coordinates is as follows:

$$\mathbf{h}_\mathbf{x} = \begin{bmatrix}0 & 1\end{bmatrix} \tag{4.7}$$

while in independent coordinates the Jacobian matrix yields:

$$\mathbf{h}_\mathbf{x} = \begin{bmatrix}0 & 0 & 0 & 0 & 1\end{bmatrix} \tag{4.8}$$

In the five-bar linkage the measurement model is $\mathbf{h}(\mathbf{x}) = \begin{bmatrix}\dot{\theta}_1 & \dot{\theta}_2\end{bmatrix}^\top$. The Jacobian in independent coordinates results:

$$\mathbf{h}_\mathbf{x} = \begin{bmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix} \tag{4.9}$$

while its counterpart in dependent coordinates yields:

$$\mathbf{h}_\mathbf{x} = \begin{bmatrix}0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1\end{bmatrix} \tag{4.10}$$

If the gyroscope is installed on the coupler rods, the measurement models become more complex. In Section 5.6 a model of a generic angular velocity sensor is covered. However, as the mechanisms considered in this section are planar, a simpler model can be used here.

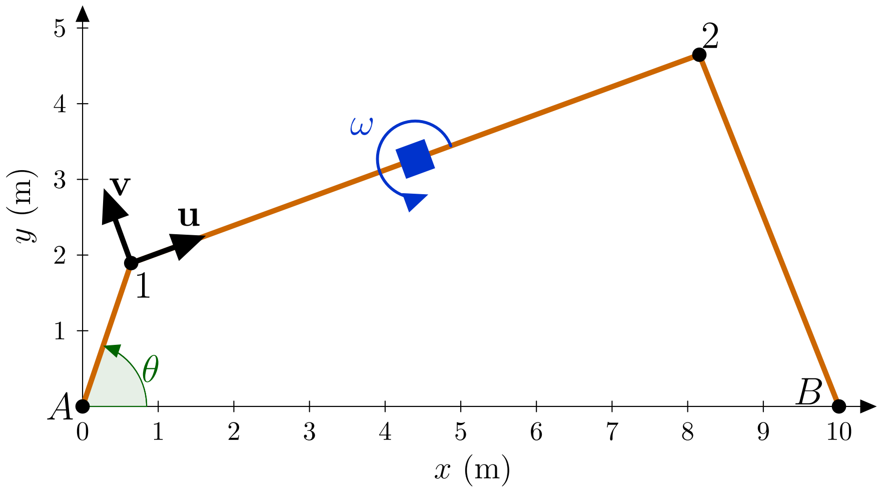


Figure 4.3: Gyroscope installed on the coupler bar of the four-bar linkage.

To illustrate this configuration, the four-bar linkage with a gyroscope installed on the coupler bar is represented in Figure 4.3. Let $\mathbf{r}_i$ and $\dot{\mathbf{r}}_i$ be the position and velocity of the point i. Then, unit vectors $\mathbf{u}$ and $\mathbf{v}$ can be defined as follows:

$$\mathbf{u} \equiv \begin{bmatrix} u_x & u_y \end{bmatrix}^\top = \frac{\mathbf{r}_2 - \mathbf{r}_1}{L_{1,2}} \tag{4.11}$$

$$\mathbf{v} \equiv \begin{bmatrix} -u_y & u_x \end{bmatrix}^\top \tag{4.12}$$

The angular velocity ω is as follows:

$$\omega = \frac{(\dot{\mathbf{r}}_2 - \dot{\mathbf{r}}_1)}{L_{1,2}}\mathbf{v} \tag{4.13}$$

The measurement model is obtained from Eq. (4.13) by writing $\dot{\mathbf{r}}_2$, $\dot{\mathbf{r}}_1$ and $\mathbf{v}$ as function of the coordinates $\mathbf{q}$ and velocities $\dot{\mathbf{q}}$, yielding:

$$\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \omega = \frac{(\dot{x}_2 - \dot{x}_1)(y_1 - y_2) + (\dot{y}_2 - \dot{y}_1)(x_2 - x_1)}{(L_{1,2})^2} \tag{4.14}$$

Thus, the Jacobian matrices of the measurement model with respect to the dependent coordinates $\mathbf{q}$ and velocities $\dot{\mathbf{q}}$ yield:

$$\mathbf{h}_{\mathbf{q}} = \frac{1}{(L_{1,2})^2}\begin{bmatrix} (\dot{y}_1 - \dot{y}_2) & (\dot{x}_2 - \dot{x}_1) & (\dot{y}_2 - \dot{y}_1) & (\dot{x}_1 - \dot{x}_2) & 0 \end{bmatrix} \tag{4.15}$$

$$\mathbf{h}_{\dot{\mathbf{q}}} = \frac{1}{(L_{1,2})^2}\begin{bmatrix} (y_2 - y_1) & (x_1 - x_2) & (y_1 - y_2) & (x_2 - x_1) & 0 \end{bmatrix} \tag{4.16}$$

Then, the Jacobian of the measurement model for methods in dependent coordinates results as follows:

$$\mathbf{h}_{\mathbf{x}} = \begin{bmatrix} \mathbf{h}_{\mathbf{q}} & \mathbf{h}_{\dot{\mathbf{q}}} \end{bmatrix} \tag{4.17}$$

For state observers in independent coordinates, the Jacobian matrices $\mathbf{h_z}$ and $\mathbf{h_{\dot{z}}}$ have to be calculated from $\mathbf{h_q}$ and $\mathbf{h_{\dot{q}}}$, according to Eqs. (3.120) and (3.121), respectively. Then, the Jacobian of the measurement model with respect to the states is build as follows:

$$\mathbf{h_x} = \begin{bmatrix} \mathbf{h_z} & \mathbf{h_{\dot{z}}} \end{bmatrix} \tag{4.18}$$

The process is analogous in the case of the five-bar linkage.

4.6. Implementation

All the methods described in Chapter 3, and the mechanisms and sensor configurations described in this chapter have been implemented in MATLAB®. The structure of the implementation allows to reuse the code with minimal effort, therefore the code has been released as Open Source under the GNU GPL v3[1] license in `https://github.com/MBDS/mbde-matlab/`.

The main steps to use the software are explained hereafter. First, the state observer to be used is defined. For example, `estim = mbeEstimatorDEKF` is defined to use the DEKF method, or `estim = mbeEstimatorUKF` to employ the UKF estimator. Next, one of the defined mechanisms is selected, in this case the five-bar linkage:

```
estim.mech_phys_model = mbeMechModelFiveBars1();
```

Then, the sensors employed have to be declared. For example, the configuration with two encoders in the cranks is shown here:

```
estim.mech_phys_model.installed_sensors =  ...
{mbeSensorPosIndex(7, sen_noise) mbeSensorPosIndex(8, sen_noise)};
```

where the function `mbeSensorPosIndex` takes two arguments: the index of the coordinate to be used as a sensor, and the standard deviation of the noise to be added to that coordinate. In this example, the coordinates 7 and 8 are the angles θ_1 and θ_2 of the five-bar linkage.

Next, the kinds of errors and their scale parameter are provided:

```
estim.bad_model_errors.error_type = [1,2];
estim.bad_model_errors.error_scale = 1;
```

There are several kinds of errors already available. In this example, two of them are applied: 1, gravity error; 2, initial position error. The error scale multiplies a predefined quantity for every kind of error. The predefined value for the errors is 1 m/s^2 for the acceleration of gravity, and $\pi/16$ rad for the initial position.

To finish the configuration, the plant noise and the simulation parameters are set:

```
estim.transitionNoise_Zpp = 0.06;
estim.dt        = 5e-3;
estim.mechanism_type.multirate_sensor_period = estim.dt*10;
estim.end_time = 10.0;
```

[1] See `http://www.gnu.org/licenses/gpl-3.0.en.html`

Here the variance of the acceleration of the plant noise is provided. The covariance matrix of the transition noise is calculated as explained in Section 3.6.1. As the plant noise is not Gaussian additive noise, the plant noise becomes a tuning parameter which must be adjusted to obtain the best performance possible from the filter, but it does not have a clear physical significance. The other parameters impose a time step of the simulation of 5 ms, a sampling period of the sensors of 50 ms, and a simulation time of 10 s.

Finally, the simulations are run by invoking the next command:

```
estim.run_offline();
```

This command executes two multibody models: one playing the role of the real mechanism, and other with some the differences defined with `error_type`, pretending that are modeling errors. From the first model, the ground truth and the sensor measurements are build, adding the noise defined by the user. Finally, the state observer is run, using the model with errors as its plant, and the virtual sensors built from the perfect model as its measurements. The data from the execution of the two multibody model are saved. Therefore, if several tests are executed consecutively without changing the parameters of the simulation, only the state observer is run.

The structure of the library was not designed to be efficient, but easy to use and flexible instead, thus allowing to change any parameter of the simulations easily. Moreover, it provides a means to evaluate the relative efficiency of one method compared to the others, which is useful to provide guidance before implementing a more specific program for a particular problem.

4.7. Tests and results

As said before, the noise of the sensors considered in these tests is white and Gaussian. This hypothesis can be justified by the fact that, in real circumstances, the noise comes from many independent sources. Then, because of the central limit theorem, assuming that the noise follows a Gaussian distribution is reasonable. In a real case, the noise from the sensors must be analyzed to obtain its properties as part of the identification process.

However, few errors are dominant in the plant, usually producing acceleration errors which are not Gaussian nor white. Therefore, the covariance matrix of the plant noise has no physical meaning, but it has to be adjusted to provide the best results possible. This process is guided by examining the innovation sequence. When the covariance matrices of the filters are properly set, the innovation sequence must behave like white noise [47]. Therefore, the power spectral density (PSD) of the innovation sequence should be flat. If the lower frequencies are predominant, the plant noise should be increased to correct this behavior, and vice versa.

The two mechanisms described before, with two levels of modeling error, and with the three different sets of sensors previously depicted have been considered here. The time step employed by the multibody simulation was 5e-3 s in all the tests, but different sampling rates for the sensors were considered in every configuration, from 10 to 200 Hz, which represents the frequency range of the majority of commonly used sensors. However, this range been selected only as a reference, but none of the observers is limited to it. The upper limit of the frequency is defined by the frequency of the multibody simulation. If higher rates were to be used, the time step should be reduced. The lower limit depends both on the quality of the multibody simulation, and on the information provided by the available measurements.

Only the methods in independent coordinates and the error-state formulation were tested. The methods in dependent coordinates were discarded because they showed worst accuracy and stability.

The discrete-time methods were found to be more robust and accurate than the CEKF, but the latter is maintained here as a reference because it has been employed in several works previously [1, 3, 36].

The sensors considered in the tests are gyroscopes (angular rate sensors) and encoders. Their measurements were built from a multibody simulation of the mechanism, and then a sequence of pseudo-random noise with a normal distribution with mean 0 and standard deviation of $\pi/180$ was added. The units of the noise are rad/s for the gyroscopes and rad for the encoders. The same sequence of pseudo-random noise was used for all the tests for a fair comparison. Slightly different results are expected with other sequences of noise. However, the relative behavior of the methods is the same with different sequences of noise.

The tests consist in the mechanisms starting from rest and falling under the action of their own weight for ten seconds. The errors intentionally introduced in the model are the value of the acceleration of gravity, representing an error in a force model which affects during all the simulation, and an error in the initial position of the mechanism. The initial position error is specially useful to show that the methods based only in gyroscopes can provide, in some configurations, accurate estimations of the position, which is not usually reported in the bibliography. Two levels of modeling error were considered: 0.5 m/s^2 error in gravity acceleration with an initial position error of $\pi/32$ rad, and 1 m/s^2 with an initial error of $\pi/16$ rad.

4.7.1. Tests with position sensors

In these tests it was assumed that angular measurements of the angle of the cranks of the mechanisms are available. Encoders are usually the preferred option to measure rotational motion since they can provide accurate measurements at a high sampling rate. For this reason, if enough encoders are available, a state observer is not usually needed. However, not all the position sensors have such a high sampling rate or accuracy. Thus, these tests are employed to verify the behavior of the state observers when generic position sensors are available, but considering that they can have a slower sampling rate than actual encoders, because they could be other kinds of position sensors, such as a GPS, laser or ultrasonic distance sensor.

The results from the four-bar and five-bar linkages are shown in Figure 4.4 and Figure 4.5 respectively, where no significant differences can be observed. The worst accuracy is obtained from the CEKF. Moreover, it is the less stable when the measurements are not available at every time step, being the only method which is not able to handle all the tested sampling rates. All the other methods considered, namely, DEKF, UKF with trapezoidal rule integrator (UKF TR), UKF with forward Euler integrator (UKF FE), and errorEKF provide the same level of accuracy.

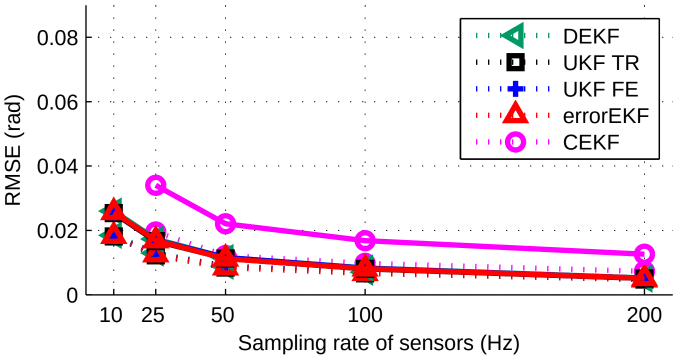


Figure 4.4: RMS error of the crank angle in the four-bar linkage with an encoder in the crank. Dashed lines represent the low modeling error test, while solid lines represent the high modeling error. The CEKF method cannot handle all the sampling rates of the sensors tested with the other methods.

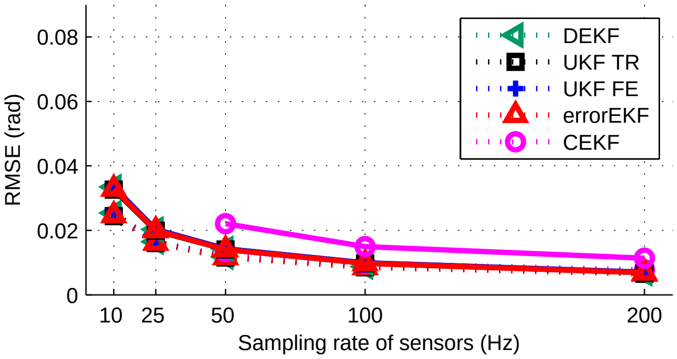


Figure 4.5: RMS error of cranks angles in the five-bar linkage with encoders in the cranks. Dashed lines represent the low modeling error test, while solid lines represent the high modeling error.

Moreover, they provide the same precision for both levels of modeling error if the sampling rate of the measurements is high enough, but when the sensors have a slower sampling rate, the test with low modeling error provides best results. It is also interesting to remark that the RMS error of the sensor is $\pi/180 \approx 0.0175$ rad. All the discrete-time methods achieve a smaller error if the sampling rate of the sensor is, at least 50 Hz.

With this sensor configuration, the sensor provides measurements of the desired magnitude, and hence the measurement model is linear. This is the reason why the methods based on the extended Kalman filter can get the same accuracy level than the unscented Kalman filters.

4.7.2. Tests with velocity sensors

Two different configurations using only velocity sensors are considered in each mechanism, leading to different situations. In the first configuration, the angular rate sensors (gyroscopes) are fixed to the coupler link of the four-bar mechanism, and to both coupler links in the five-bar linkage. Since both mechanisms present a closed-loop topology, the velocity and the position of such elements are related, and hence the absolute position of the mechanism can be inferred from velocity measurements, even if the initial position is not accurately known. Here, both the geometry of the mechanism and the characteristics of the motion affect the accuracy of the state observer. The results for the four-bar linkage are shown in Figure 4.6. Now the UKF methods show the best accuracy, independently of the type of integrator used. The CEKF is the next in accuracy with high sampling rate from the sensors, but if measurements are not available at every time step, its performance degrades faster than the other methods, and it cannot deal with all the sampling rates whereas the other methods can. The worst accuracy in this test was provided by the DEKF and the errorEKF methods, but they can deal with low frequency measurements from the sensors. Moreover, it seems that their performance degrades with higher sampling rates. However, this is not true. Indeed, at the beginning of the simulations the system is not observable with this sensor configuration, because the mechanism starts from rest. When the simulation starts, the first measurements are not coherent because of the error in the initial position, and this produces that the first measurements in this particular test take the state observer farther from the true solution instead of closer. For this reason, the test with the highest measurement frequency has to recover a bigger error from the beginning. The absolute value of the error considering sampling rates of 50, 100, and 200 Hz is shown in Figure 4.7 to illustrate this phenomenon. A plot of the crank angle is shown in Figure 4.8, and its error during the whole test is displayed in Figure 4.9, using the DEKF formulation with a sampling rate of 200 Hz. The confidence interval represented in Figure 4.9 is calculated from the values of the covariance matrix of the state estimation uncertainty $\mathbf{P}$. It can be seen how the confidence interval expands and shrinks depending on the motion of the mechanism, since it depends on both the velocity of the coupler bar and the relation between this velocity and the position of the crank.

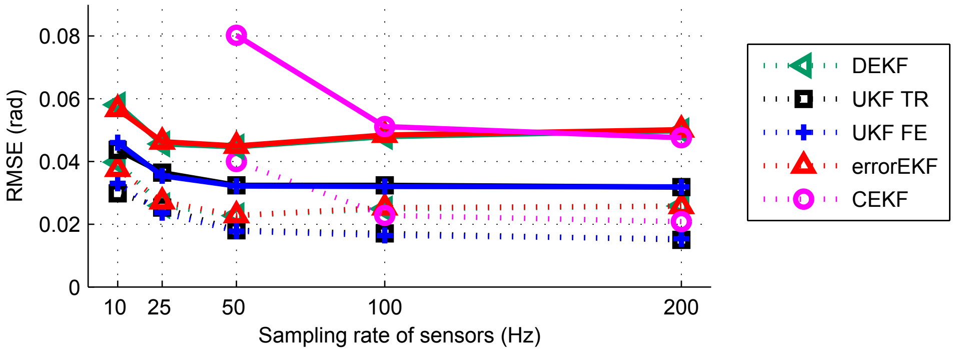


Figure 4.6: RMS error of the crank angle in the four-bar linkage with an angular rate sensor in the coupler link. Dashed lines represent the low modeling error test, while solid lines represent the high modeling error.

The results from the tests with the five-bar mechanism are shown in Figure 4.10.

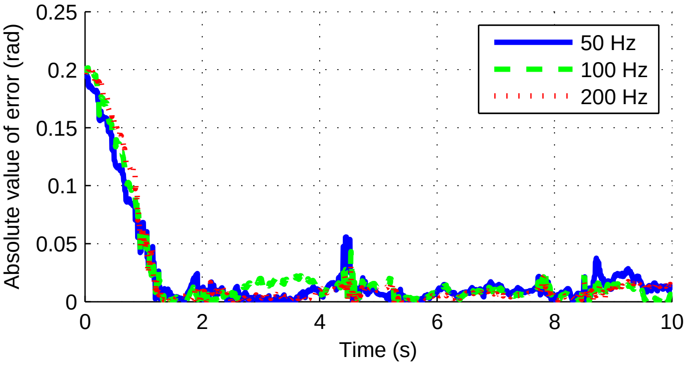


Figure 4.7: Absolute value of the error of the crank angle in the four-bar linkage with an angular rate sensor in the coupler link, with sampling rates of 200, 100, and 50 Hz. These tests were performed with the DEKF with $\pi/16$ initial error, and 1 m/s^2 of error in gravity acceleration.

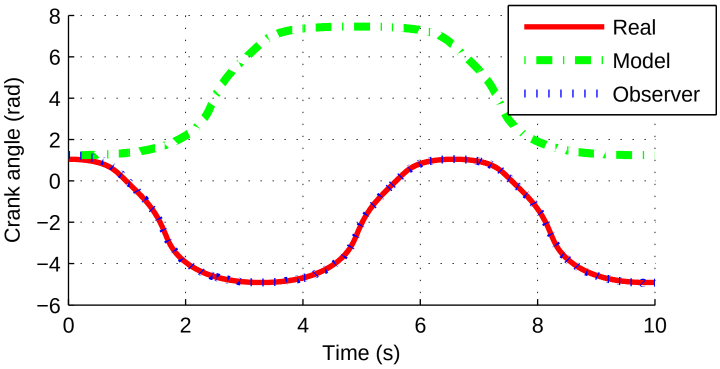


Figure 4.8: Angle of the crank of the four-bar mechanism for the real mechanism, the model without corrections, and the state observer with an angular rate sensor in the coupler link. This test was performed with the DEKF formulation, 200 Hz sampling rate for the sensor, $\pi/16$ initial error, and 1 m/s^2 of error in gravity acceleration.

Again, the UKF methods have the best accuracy. However, in this test the errorEKF is closer to them than the DEKF. The CEKF obtained the worst results, and it was not able to run when measurements were not provided at every time step.

The second configuration tested with velocity sensors consists in two gyroscopes installed on the cranks of the mechanisms. This problem is more challenging than the previous one because the cranks can have any velocity at any position, hence there is no relationship between position and velocity. However, if the accelerations depend on the positions, the UKF methods can still provide information of the position from velocity measurements. The EKF methods cannot deal with this problem, so they are not considered here. In this case, the acceleration varies with the position of the cranks because the only actuating force is the gravity. The results are similar for both mechanisms, as shown in Figure 4.11 for the four-bar linkage and in Figure 4.12 for the five-bar linkage. In this configuration, the type of integrator considerably determines the accuracy of the methods, providing better results the

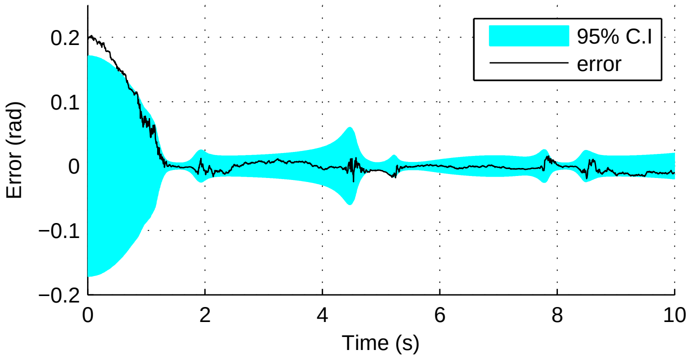


Figure 4.9: Error of the crank angle in the four-bar linkage with an angular rate sensor in the coupler link, and its 95% confidence interval. This test was performed with the DEKF formulation, 200 Hz sampling rate for the sensor, $\pi/16$ initial error, and 1 m/s^2 of error in gravity acceleration.

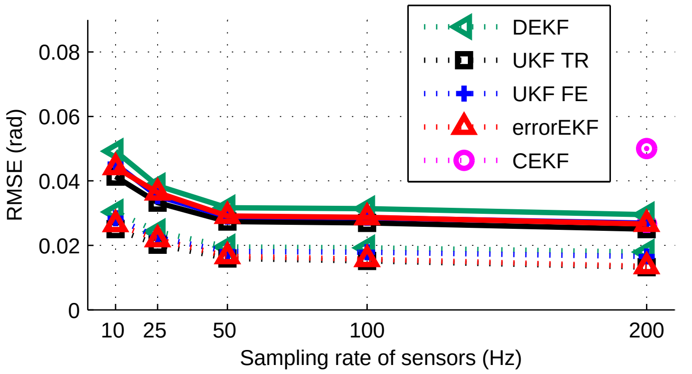


Figure 4.10: RMS error of the cranks angles of the five-bar linkage with angular rate sensors in the intermediate links. Dashed lines represent the low modeling error test, while solid lines represent the high modeling error.

trapezoidal rule. This difference increases when the errors in the plant are smaller, because if the errors in force model are high enough, the integration error becomes negligible.

Figure 4.13 shows the error of the test performed with the UKF FE with 1 m/s^2 of error in the acceleration of gravity and $\pi/16$ rad of initial position error.

4.7.3. Computational cost

For a state observer to be useful, it must be run in real time. However, the tests presented here were run in MATLAB®, and the design of the software was intended to provide flexibility instead of efficiency, not being appropriated to be used as practical state observers. Nevertheless, the computational cost of these methods is evaluated to provide guidance in order to select a state observer for future applications. Every test takes a different time to be run, depending on the number and type of sensors , their sampling frequency, etc. However, as a comparison

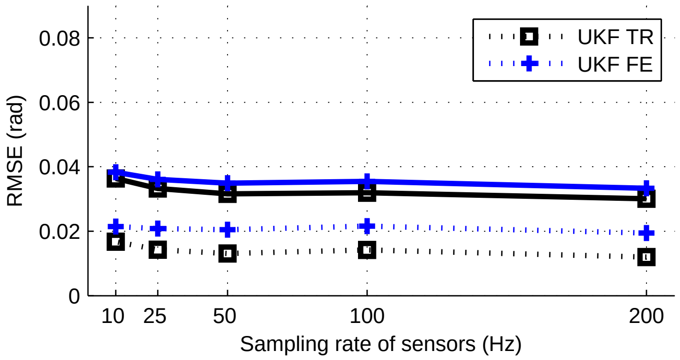


Figure 4.11: RMS error of the crank angle in the four-bar linkage with an angular rate sensor in the crank. Dashed lines represent the low modeling error test, while solid lines represent the high modeling error test.

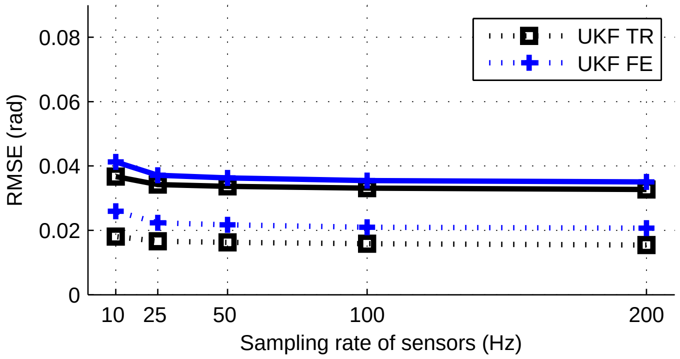


Figure 4.12: RMS error of the crank angles of the five-bar linkage with angular rate sensors in the cranks. Dashed lines represent the low modeling error test, while solid lines represent the high modeling error test.

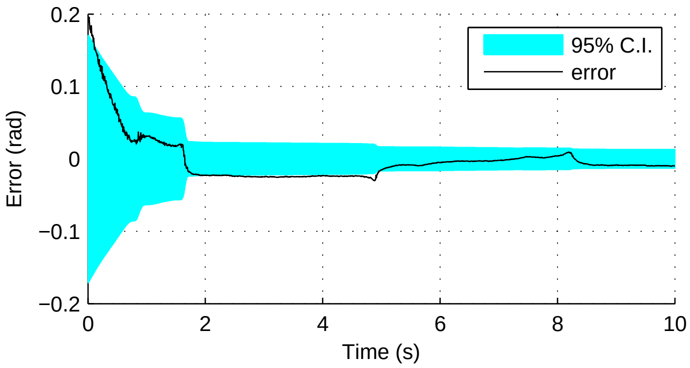


Figure 4.13: Error of the crank angle in the four-bar linkage with an angular rate sensor in the crank, and its 95% confidence interval. This test was performed with 200 Hz sampling rate for the sensor, $\pi/16$ initial error, and 1 m/s^2 of error in gravity acceleration.

among the different formulations is intended here, only the results from the tests with position sensors at 200 Hz are depicted in Figure 4.14. It can be seen that the two discrete-time extended Kalman filters are the fastest methods, being the errorEKF the only one which runs faster than real time. The CEKF method is much slower than its discrete counterparts. As for the unscented Kalman filters, the formulation employing the forward Euler integrator is about twice faster than the same formulation using the trapezoidal rule. The UKF FE was faster than the CEKF for the problems considered in this chapter. However, its computational cost grows faster as the number of degrees of freedom of the mechanism increases, so it is expected that it would be slower than the CEKF for problems of a higher number of degrees of freedom.

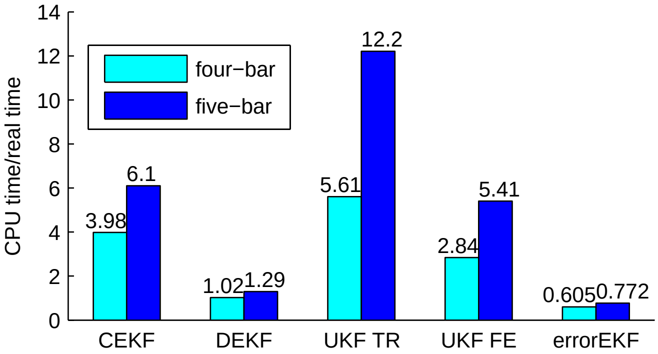


Figure 4.14: Real time factor of the different methods running with position sensors at 200 Hz.

4.7.4. Observability

The selection of the sensors needed to achieve the estimation of the desired magnitudes is a problem to be addressed when the state observer is designed. The study of the observability provides guidance in order to select the necessary sensors during the design process.

The state of a model is said to be observable if it is uniquely determined by the system model, its inputs and its outputs [46]. From the results obtained in section 4.7, the more interesting methods to be studied are the DEKF and the errorEKF for their efficiency, and the UKF due to its accuracy. Therefore, only discrete-time methods in independent coordinates are considered in this section.

The extended Kalman filters used here are based on nonlinear models. However, due to the discretization scheme employed (the forward Euler integrator), the resulting transition matrices are constant, and the nonlinearities of the transition models are only revealed in the calculation of the accelerations.

The sensor models, however, can be linear or not, depending on the kinds of sensors installed and their configuration. For example, the tests performed with the four-bar mechanism considering an encoder or a gyroscope on the first link lead to linear measurement models, while the test with a gyroscope on the coupler has a nonlinear measurement model.

As some of the cases have time varying matrices, the concept of local observability is used here, determined by the rank of the following matrix [69]:

$$\mathbf{M}_{lo} = \begin{bmatrix} \mathbf{h}_{\mathbf{x}k} \\ \mathbf{h}_{\mathbf{x}k+1}\mathbf{f}_{\mathbf{x}k} \\ \mathbf{h}_{\mathbf{x}k+2}\mathbf{f}_{\mathbf{x}k+1}\mathbf{f}_{\mathbf{x}k} \\ \vdots \\ \mathbf{h}_{\mathbf{x}k+l-1}\mathbf{f}_{\mathbf{x}k+l-2}\dots\mathbf{f}_{\mathbf{x}k} \end{bmatrix} \tag{4.19}$$

If $\mathbf{M}_{lo}$ is full rank, the system is observable. However, errors in the algorithm (not achieving a perfect fulfillment of the constraints, numerical errors, etc.) can make $\mathbf{M}_{lo}$ full rank when the system is actually non-observable. Moreover, the observability of the system can be weak, or progressively become non-observable. For these reasons, instead of the rank of the matrix, the condition number of the matrix, $\kappa(\mathbf{M}_{lo})$ is used [46], since it is a quantitative assessment of the observability. The condition number is calculated as the relation between the maximum and the minimum singular value of $\mathbf{M}_{lo}$:

$$\kappa(\mathbf{M}_{lo}) = \frac{\sigma_{max}}{\sigma_{min}} \tag{4.20}$$

Since $\kappa(\mathbf{M}_{lo})$ can reach very high values, its logarithm is the magnitude to be studied. The lower value this magnitude has, the better observability the system has.

When using sampled methods, such as the UKF, the matrices $\mathbf{f}_\mathbf{x}$ and $\mathbf{h}_\mathbf{x}$ used to build $\mathbf{M}_{lo}$ are not available. However, the UKF can be seen as a particular case of the linear regression Kalman filter [70], in which the matrices $\mathbf{f}_\mathbf{x}$ and $\mathbf{h}_\mathbf{x}$ are obtained as a statistical linear regression of several points propagated through the transition and measurement functions. Once these matrices are obtained from the transition and measurement updates of the UKF, they can be used to build the local observability matrix [71].

Using this method, the observability of the EKF with constant matrices can be examined before implementing the state observer. In the tests presented in this chapter with the four-bar mechanism, the case with the encoder and the case with the angular rate sensor on the first link can be analyzed this way. With the encoder, the system is always observable. However, if a gyroscope is used in the first link, the system is never observable.

Unfortunately, when the EKF methods with a nonlinear measurement models or the UKF algorithms are employed, the matrices needed to perform the observability analysis are not available beforehand. Consequently, a simulation of the method under the expected working conditions has to be performed, limiting the interest of this observability analysis. However, the observability analysis is still useful to detect conditions in which the observability is weaker or temporarily lost under some working conditions.

The observability of the errorEKF, DEKF and UKF was examined during a test with each one of the configurations tested. The results obtained from the DEKF and the errorEKF were similar, being observable with the encoders on the cranks, and with the gyroscopes on the coupler bars, and non-observable when the gyroscope was installed on the cranks, both for the four-bar and the five-bar linkages.

The logarithm of the condition number of $\mathbf{M}_{lo}$ of the DEKF for the different tests performed with the four-bar linkage are shown in Figures 4.15 to 4.17. The results obtained with the UKF are shown in Figures 4.18 to 4.20. It can be seen that the tests with encoders or gyroscopes on the coupler bar are almost identical to those obtained with the EKF methods. However, in the test with gyroscopes on the cranks, the UKF is still observable, while the EKF methods are not.

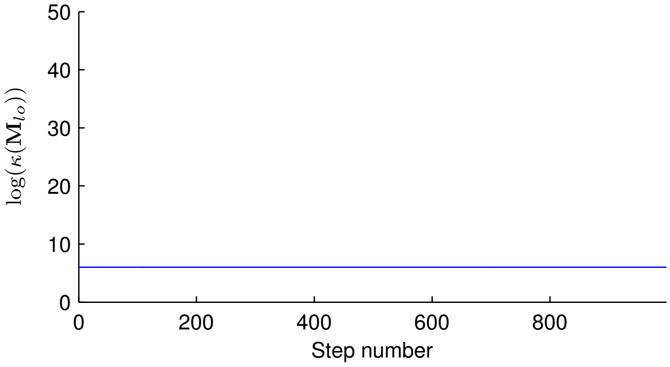


Figure 4.15: Observability of the four-bar mechanism with an encoder on the crank and the DEKF method.

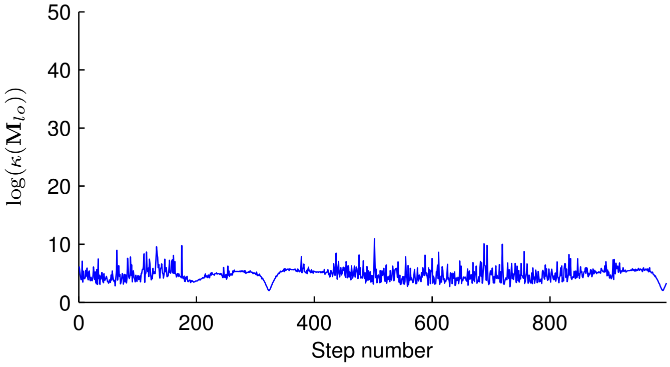


Figure 4.16: Observability of the four-bar mechanism with a gyroscope on the coupler and the DEKF method.

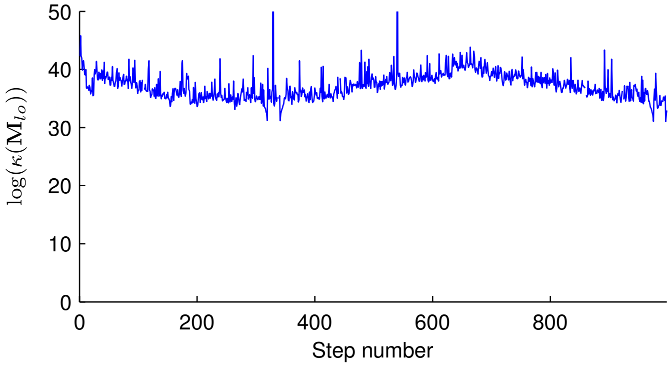


Figure 4.17: Observability of the four-bar mechanism with a gyroscope on the crankand the DEKF method.

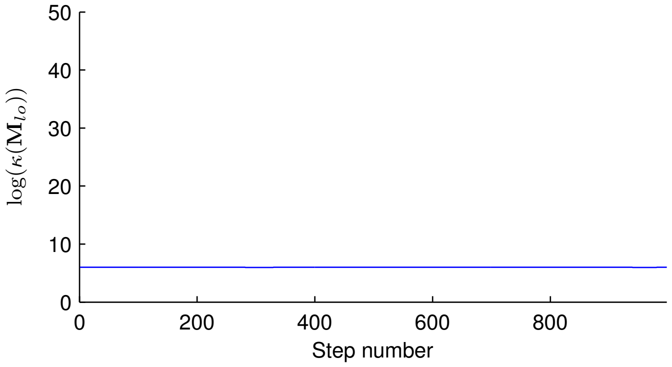


Figure 4.18: Observability of the four-bar mechanism with an encoder on the crank and the UKF method.

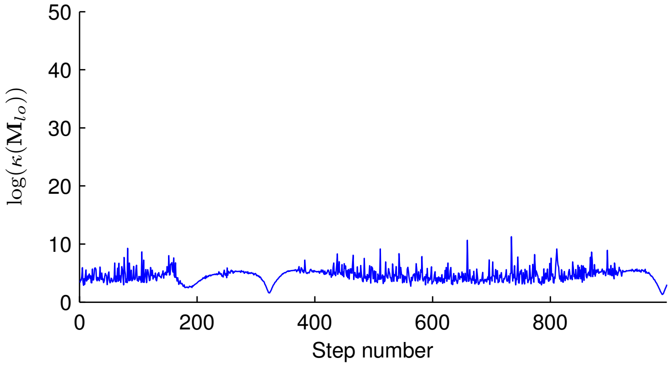


Figure 4.19: Observability of the four-bar mechanism with a gyroscope on the coupler and the UKF method.

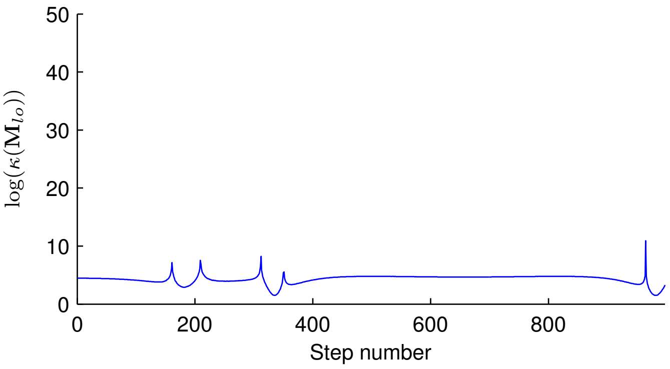


Figure 4.20: Observability of the four-bar mechanism with a gyroscope on the crank and the UKF method.

Chapter 5

Application to an automobile

The only attempt reported in the literature to apply a state observer to a detailed multibody model of an automobile was presented in [3], where an improved version of the CEKF was used, considering that position sensors were available for every degree of freedom of the model at every time step.

In this chapter, a more realistic approach was used: a multibody model of a vehicle prototype was developed, and the sensors available in the prototype were considered to build the state observer, considering also their actual sampling rate. The state observer was also tested with data gathered from actual maneuvers.

From the results obtained in chapter 4, the errorEKF method was considered to implement the state observer. Besides the efficiency of the method, the structure of the errorEKF allows implementing it over an existing multibody model without doing severe changes to the code, making this formulation the best candidate to implement the state observer of a complex model.

5.1. Vehicle prototype

The vehicle employed in this thesis is a full-size steer-by-wire (SBW) vehicle, shown in Figure 5.1. It was developed at the Laboratorio de Ingeniería Mecánica of University of A Coruña, and was used in several works before [72, 73, 4, 74, 60].

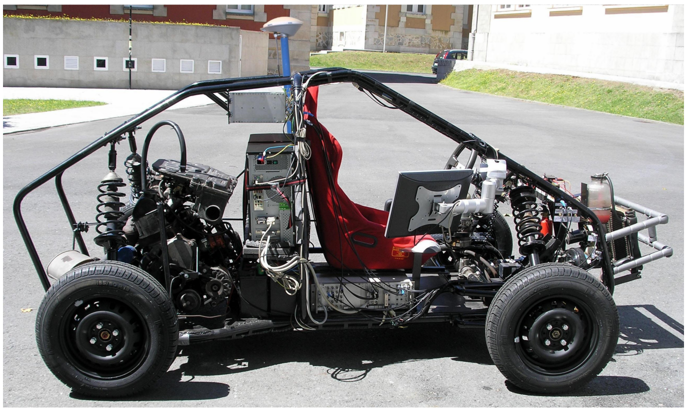


Figure 5.1: Vehicle prototype used in this thesis.

This vehicle is equipped with a four-cylinder gasoline engine. The power is transmitted to the rear wheels through an automatic gearbox, whereas the brake system employs four discs.

On board hardware consists of a desktop computer (Intel Core2 Duo E8500 at 3.16 GHz, 2 Gb of RAM) hosting a PCI data acquisition processor (DAP), model DAP 4200a, made by Microstar Laboratories. The only modifications made to the computer are the power supply and the hard drive. The power supply works with 12 V of direct current, provided by the battery of the prototype. The hard disk drive was changed for a solid state drive (SSD), because it is more resistant to vibrations. The DAP controls the SBW system and samples all the sensors installed on the vehicle at 500 Hz. Then, it sends the data through a PCI bus to the computer. This rate allows for a stable and efficient real-time control of the SBW system. The allocation of the processor of the DAP is accomplished by using a round robin scheduling algorithm, which guarantees low latency in the response of the system. The full list of the sensors installed on board is given in Table 5.1. A high precision differential GPS receiver was added to the vehicle instrumentation. It is not connected to the DAP, but it is connected to the computer via a serial port. The GPS works at 50 Hz.

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

Table 5.1: List of sensors installed on board the vehicle.

Most of the sensors installed on the prototype were selected based on the types of sensors available in production vehicles. Extra sensors, such as the wheel torque sensor and the high precision GPS, were added in order to develop force models and to track the trajectory of the vehicle. Moreover, some of these sensors are employed to feed the multibody model. A detailed description of most of the vehicle instrumentation and its x-by-wire systems can be find in [4]. The only addition since then was the GPS receiver. Its properties and usage are described hereafter.

5.1.1. GPS receiver

A GPS is a common positioning system based on a satellite constellation. Although the generic name of the technology is global navigation satellite system (GNSS), it is usually best known as GPS because it is the name of the first functional satellite-based positioning system, hence this is the name used in this thesis.

A GPS receiver calculates the position of its antenna (actually, a point called *phase center*, which sometimes is not physically inside the antenna) by measuring the distance from this antenna to a constellation of satellites. The distance is determined by measuring the time since a satellite signal is sent until it arrives to the antenna of the receiver. Positioning errors can be produced due to the position errors of the satellites, atmospheric effects, clock drifts, etc. However, if two GPS receivers are available, and one of them is at a known location, this receiver (named *base*) can measure the positioning errors, and the other one (named *rover*) uses this information to correct the calculated position, as shown in Figure 5.2. This technique is called differential GPS. The GPS used in this thesis (the Settop M1 model from Al-top Topografía) employs this technique to achieve up to 1 cm of precision under optimal conditions, and it can reach an output rate of 50 Hz. Moreover, a GPS receiver can provide speed and course measurements based on the Doppler effect. However, course measurements are only valid if the speed is above a certain threshold.

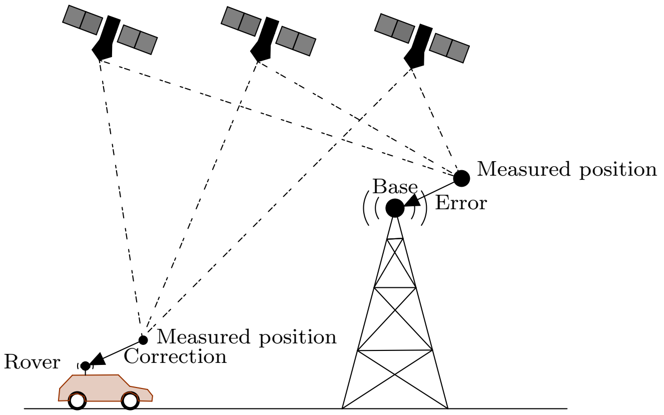
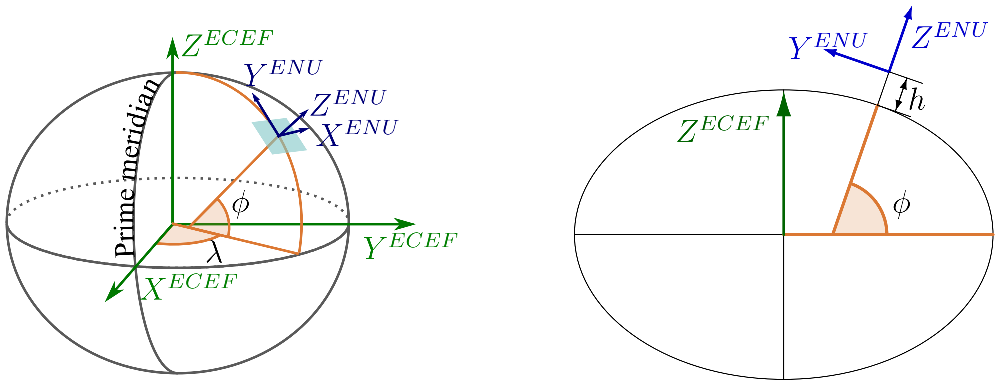


Figure 5.2: Working principle of differential GPS.

The GPS positioning output consists of geodetic coordinates: longitude (λ), latitude (ϕ), and ellipsoidal height (h), as shown in Figure 5.3. By default, these coordinates are referred to the World Geodetic System 1984 (WGS84), the system in which the GPS satellites are located, but if differential corrections are applied, they can include a reference transformation. In this work, corrections provided by the IGN were used to locate the base. These corrections change the reference system from WGS84 to European Terrestrial Reference System 1989 (ETRS89), the official geodetic reference system in Spain. The ellipsoids of both systems are almost identical (about 0.1 mm of difference in one of the semiaxes of the ellipsoid), and the only significant difference is that the WGS84 is fixed to the center of gravity of the Earth, while the ETRS89 moves with the Eurasian plate. The main parameters of both systems are included in Table 5.2, taken from [75] and [76].

The geodetic parameters of latitude ϕ and height h are defined in terms of the ellipsoid normal at the user's position (see Figure 5.3b). Notice that, unless the user is on the poles or the equator, the ellipsoid normal does not point exactly towards the center of the Earth.

In order to be used along with the multibody model, the geodetic coordinates are not very useful, so they must be transformed to the same reference axis of



(a) ECEF and ENU reference systems.

(b) Definition of ellipsoidal altitude, h.

Figure 5.3: Reference systems employed to covert the GPS measurements.

Reference system	**Semi-major axis** a (m)	**Semi-minor axis** b (m)	**Ellipsoid flattening,** $\frac{a-b}{b}$
WGS84	6 378 137	6 356 752.314 25	1/298.257 223 563
ETRS89	6 378 137	6 356 752.314 14	1/298.257 222 101

Table 5.2: Main parameters of WGS84 and ETR89 ellipsoids.

the multibody simulation: the ENU, defined in Figure 5.3. This transformation is carried out in two steps. First, a transformation from geodetic to Earth-centered Earth-fixed reference system (ECEF) is performed, following the next equations [77]:

$$\mathbf{r}^{ECEF} = \begin{bmatrix} \dfrac{a\cos\lambda}{\sqrt{1+(1-e^2)\tan^2\phi}} + h\cos\lambda\cos\phi \\ \\ \dfrac{a\sin\lambda}{\sqrt{1+(1-e^2)\tan^2\phi}} + h\sin\lambda\cos\phi \\ \\ \dfrac{a(1-e^2)\sin\phi}{\sqrt{1-e^2\sin^2\phi}} + h\sin\phi \end{bmatrix} \tag{5.1}$$

being a and b are the ellipsoidal semi-axes defined in Table 5.2, and $e = \sqrt{1 - b^2/a^2}$ the ellipsoid eccentricity. Then, the transformation from ECEF to ENU must be accomplished as follows:

$$\mathbf{r}^{ENU} = \mathbf{\Psi}\left(\mathbf{r}_P^{ECEF} - \mathbf{r}_0^{ECEF}\right) \tag{5.2}$$

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

$$\mathbf{\Psi} = \begin{bmatrix} -\sin\lambda & \cos\lambda & 0 \\ \cos\lambda\sin(-\phi) & \sin\lambda\sin(-\phi) & \cos(-\phi) \\ \cos\lambda\cos(-\phi) & \sin\lambda\cos(-\phi) & -\sin(-\phi) \end{bmatrix} \tag{5.3}$$

For every considered scenario, an ENU system is defined, and it is maintained for the whole simulation.

5.1.2. GPS sentences

The GPS receiver provides data in form of text streams, whose format is defined by the NMEA 0183 standard [78]. From the sentences defined in this standard, GGA was used for positioning information, and RMC to obtain speed and course information. The format of the sentences is provided hereafter. Next, an example of a GGA sentence is provided:

```
$GPGGA,100412.58,4329.61417292,N,00814.86569421,W,4,14,0.7,6.846,M,53.031,M,1.6,1503*5A
```

The fields of the GGA sentence are described in Table 5.3.

Name	**Example**	**Unit**	**Description**
Message ID	$GPGGA		GGA protocol header
UTC Time	100412.58	hhmmss.ss	
Latitude	4329.61417292	ddmm.mmmmmmmm	
N/S Indicator	N		N = north, S = south
Longitude	00814.86569421	dddmm.mmmmmmmm	
E/W Indicator	W		E = east, W = west
Position Fix Indicator	4		Indicates the quality of the position fix[a]
Satellites used	14		
HDOP	0.7		Horizontal Dilution of Precision [b]
MSL Altitute	6.846	meters	Altitude with respect to geoid[c]
Units	M		Meters
Geoid Separation	53.031	meters	Height of the geoid above the ellipsoid[d]
Units	M		Meters
Age of Diff. Corr.	1.6	seconds	Age of the differential correction when differential GPS is used
Diff. Reff. Station ID	1503		ID of the differential reference station
Checksum	*5A		
<CR><LF>			End of message termination

[a] The possible values with the GPS receiver used are:

0 Invalid fix

1 GPS fix, without differential corrections

2 DGPS fix, with differential corrections

4 Real Time Kinematic (RTK) mode. This mode is the most precise, achieving up to 1 cm of accuracy.

5 Float RTK.

[b] The Horizontal Dilution of Precision is a parameter which is related to the geometry of the satellites used to calculate he solution. Small values (<1) indicate an ideal satellite distribution.

[c] The geoid is an equipotential surface that coincides with mean see level (MSL) in the ocean and its extension through the continents.

[d] The h used in coordinate transformations in Section 5.1.1 is MSL Altitude + Geoid separation.

Table 5.3: Description of the fields of a GGA sentence.

The RMC sentence provides information of horizontal position, course, and speed. However, it does not provide information about the position fix indicator nor the altitude. An example of a RMC sentence is provided hereafter:

```
$GPRMC,092204.999,A,4250.5589,S,14718.5084,E,0.00,89.68,211200,,*25
```

The fields of a RMC sentence are described in Table 5.4. Although not present in this example (nor employed in this thesis), sometimes the magnetic deviation is also provided in this sentence.

5.2. Multibody model of the vehicle

The real-time multibody model was defined with mixed coordinates, i.e., natural and some relative coordinates [31]. The use of relative coordinates was especially

Name	Example	Unit	Description
Message ID	$GPRMC		RMC protocol header
UTC Time	092204.999	hhmmss.sss	
Status	A		A = data valid, V = data not valid
Latitude	4250.5589	ddmm.mmmmm	
N/S indicator	S		N = north, S = south
Longitude	14718.5084	dddmm.mmmm	
E/W	E		E = east, W = west
Speed Over Ground	0.00	knots	Horizontal speed with respect to the ground
Course over ground	89.68	degrees	Orientation of the trajectory with respect to true north
date	211200	ddmmyy	
Checksum	*25		
<CR><LF>			End of message termination

Table 5.4: Description of the fields of a RMC sentence.

beneficial for the model of the wheels, since they are fast rotating bodies. The details of the method used to model the wheels are given in Section 5.2.1.

The multibody model of the vehicle consists of 18 rigid bodies (see Figure 5.4): the chassis, the four wheels, four knuckles, three bars for the steering mechanism, four arms for the front suspensions (double wishbone), and two more arms for the rear suspensions (MacPherson strut). It has 169 coordinates, with 14 degrees of freedom: 6 for the rigid body motion of the chassis, 4 for the suspension and 4 more for the rotation of the wheels. The steering is kinetically guided, so it is not a degree of freedom from the multibody point of view. The reader is referred to [4] for further details on the multibody model, where all the inertial and geometric properties are given. The chapter 3 of [4], where the model is described, is provided here as the Appendix A for the convenience of the reader. The improvements made to the multibody model of the vehicle are detailed in the present thesis, including the new model of the wheels, and the improved models of forces such as the brake torque and the rolling resistance.

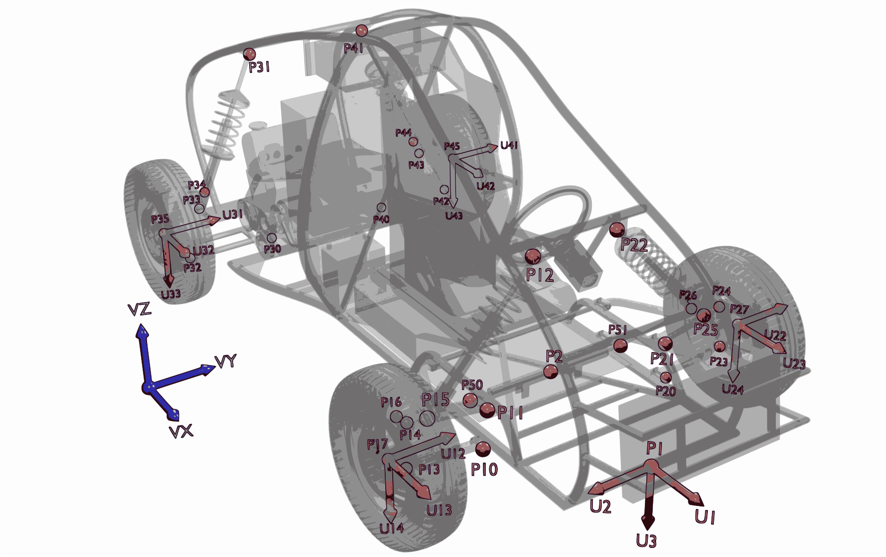


Figure 5.4: Points and main vectors of the multibody model.

The inputs for the multibody model are given by sensors installed on the prototype: a hydraulic pressure sensor in the brake line, an in-wheel torque sensor attached to one of the rear wheels, and an encoder measuring the position of the steering rack and pinion mechanism. The hydraulic pressure sensor is used as an input for a brake model, which is explained in Section 5.3. The output of this brake model is the torque applied to the front wheels. The torque measured by the in-wheel torque sensor is applied to the rear wheels. The output of this sensor is the sum of the drive and brake torques applied to the wheel. As both rear wheels have the same brake system, and the engine power is delivered through an open differential, it is assumed that the torque is the same in both rear wheels.

While developing the model, special care was put on its geometric characterization. For example, the prototype has some construction errors which lead to wheel misalignments. This fact produces a rise of the rolling resistance and has effects on the steering behavior. For this reason, the exact orientation of the wheels was measured with a wheel aligner, and then the actual orientation was introduced into the multibody model.

The multibody simulation was programmed using the MBSLIM [79], a multibody simulation library developed at the Laboratorio de Ingeniería Mecánica of the University of A Coruña. The formulation used was the augmented Lagrangian of index 3 with position and velocity projections [57], described in section 3.3.2, combined with the trapezoidal rule integrator.

5.2.1. Wheel model

The easiest way to define a wheel with natural coordinates is using one point and three vectors attached to the wheel ($\mathbf{q}_0 = (\mathbf{r}_1\, \mathbf{v}_1\, \mathbf{v}_2\, \mathbf{v}_3)^\top$, see Figure 5.5). However this modeling technique is not well suited to be applied to fast rotating elements: it leads to high energy dissipation, the convergence is slow, thus many iterations are needed to solve the dynamic system, and the simulations present instabilities for speeds above 14 m/s. For these reasons, a new approach was developed in this thesis: the wheel is modeled using point $\mathbf{r}_1$, vectors $\mathbf{v}_1$, $\mathbf{w}_2$, $\mathbf{w}_3$, and angle α (see Figure 5.5). Vectors $\mathbf{w}_2$ and $\mathbf{w}_3$ are now attached to the knuckle and α is the spin angle of the wheel. In order to develop this model, some conditions were assumed:

1. The vector of coordinates of the wheel is $\mathbf{q} = (\mathbf{r}_1\, \mathbf{v}_1\, \mathbf{w}_2\, \mathbf{w}_3\, \alpha)^\top$.

2. Both the center of mass of the wheel and the point $\mathbf{r}_1$ are aligned with the spin axis of the wheel. The distance between them is d_g. The wheel and the knuckle share point $\mathbf{r}_1$.

3. The moments of inertia of the wheel I_1, I_2, and I_3 are calculated with respect to the center of mass and following the orientation of vectors $\mathbf{v}_1$, $\mathbf{w}_2$ and $\mathbf{w}_3$. The wheel has rotational symmetry, so $I_2 = I_3$.

4. Vectors $\mathbf{v}_1$, $\mathbf{w}_2$ and $\mathbf{w}_3$ are orthonormal.

The mass matrix $\mathbf{M}_0$ is already known if the $\mathbf{q}_0$ coordinates are used [31]. However, since the coordinates $\mathbf{q}$ are employed instead, a transformation is required to obtain $\mathbf{M}$.

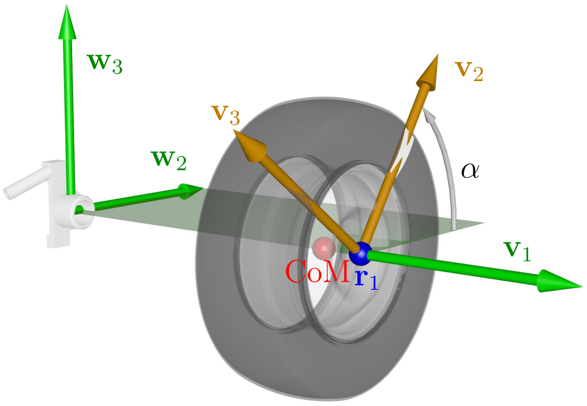


Figure 5.5: Wheel model.

The coordinates $\mathbf{q}_0$ are related to the multibody coordinates $\mathbf{q}$ through a transformation matrix $\mathbf{T}^*$.

$$\mathbf{q}_0 = \mathbf{T}^*\mathbf{q} \tag{5.4}$$

$$\mathbf{T}^* = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3\cos(\alpha) & \mathbf{I}_3\sin(\alpha) & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3\sin(\alpha) & \mathbf{I}_3\cos(\alpha) & \mathbf{0}_{3\times1} \end{bmatrix} \tag{5.5}$$

The time derivative of $\mathbf{q}_0$ can be calculated by deriving Eq. (5.4).

$$\dot{\mathbf{q}}_0 = \mathbf{T}^*\dot{\mathbf{q}} + \dot{\mathbf{T}}^*\mathbf{q} \tag{5.6}$$

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

$$\dot{\mathbf{T}}^* = \dot{\alpha} \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3\sin(\alpha) & \mathbf{I}_3\cos(\alpha) & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3\cos(\alpha) & -\mathbf{I}_3\sin(\alpha) & \mathbf{0}_{3\times1} \end{bmatrix} \tag{5.7}$$

Due to the particular structures of $\dot{\mathbf{T}}^*$ and $\mathbf{q}$, the term $\dot{\mathbf{T}}^*\mathbf{q}$ can be rearranged as follows:

$$\dot{\mathbf{T}}^*\mathbf{q} = \mathbf{B}^*\dot{\mathbf{q}} = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\sin(\alpha)\mathbf{w}_2 + \cos(\alpha)\mathbf{w}_3 \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\cos(\alpha)\mathbf{w}_2 - \sin(\alpha)\mathbf{w}_3 \end{bmatrix} \dot{\mathbf{q}} \tag{5.8}$$

where $\mathbf{B}^*$ is an auxiliary matrix employed to ease the notation. Combining this result with Eq. (5.6) leads to

$$\dot{\mathbf{q}}_0 = (\mathbf{T}^* + \mathbf{B}^*)\,\dot{\mathbf{q}} = \bar{\mathbf{T}}\dot{\mathbf{q}} \tag{5.9}$$

where matrix $\bar{\mathbf{T}} = (\mathbf{T}^* + \mathbf{B}^*)$ projects the velocities of the multibody coordinates over the system fixed to the wheel. This matrix $\bar{\mathbf{T}}$ is used to transform the mass matrix when coordinates $\mathbf{q}$ are employed, following the next equation [31]:

$$\mathbf{M} = \bar{\mathbf{T}}^\top\mathbf{M}_0\bar{\mathbf{T}} \tag{5.10}$$

Applying this transformation and taking into account all the simplifications assumed in the wheel model, the final expression for the mass matrix results:

$$\mathbf{M} = \begin{bmatrix} m\mathbf{I}_3 & md_g\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times1} \\ md_g\mathbf{I}_3 & \left(md_g^2 + I_2 - \frac{1}{2}I_1\right)\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \frac{1}{2}I_1\mathbf{I}_3 & \mathbf{0}_{3\times3} & \frac{1}{2}I_1\mathbf{w}_3 \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \frac{1}{2}I_1\mathbf{I}_3 & -\frac{1}{2}I_1\mathbf{w}_2 \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \frac{1}{2}I_1\mathbf{w}_3^\top & -\frac{1}{2}I_1\mathbf{w}_2^\top & I_1 \end{bmatrix} \tag{5.11}$$

5.2.1.1. Generalized forces vector assembly

Since the mass matrix obtained in section 5.2.1 is not constant, some velocity-dependent inertia forces have to be calculated and added to the generalized forces vector, thus it can be split in two parts, according to the following equation:

$$\mathbf{Q} = \mathbf{Q}_{ext} + \mathbf{Q}_v \tag{5.12}$$

where $\mathbf{Q}_{ext}$ is the vector of the external forces, and $\mathbf{Q}_v$ is the vector of velocity-dependent inertia forces.

In order to fit the multibody equations, the external forces must be projected over the multibody variables, i.e., point $\mathbf{r}_1$, vectors $\mathbf{v}_1$, $\mathbf{w}_2$ and $\mathbf{w}_3$, and angle α. To ease the projection process, an auxiliary system attached to the wheel is considered. It is composed of point $\mathbf{r}_1$ and vectors $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$ (see Figure 5.5), although vectors $\mathbf{v}_2$ and $\mathbf{v}_3$ are not part of the wheel model now. The virtual power $\dot{W}^*$ developed by a force $\mathbf{F}$ applied at one point $\mathbf{r}$ of the wheel is

$$\dot{W}^* = \mathbf{F}^\top\dot{\mathbf{r}}^* = \mathbf{F}^\top\mathbf{C_p}\dot{\mathbf{q}}_0^* = \mathbf{Q}_0^\top\dot{\mathbf{q}}_0^* \tag{5.13}$$

where $\dot{\mathbf{r}}^*$ is a virtual velocity compatible with system constraints, and $\mathbf{C_p}$ relates $\mathbf{r}$ to $\mathbf{q}_0$:

$$\mathbf{r} = \mathbf{r}_1 + c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + c_3\mathbf{v}_3 = \mathbf{C_p}\mathbf{q}_0 \tag{5.14}$$

$$\mathbf{C_p} = \begin{pmatrix} \mathbf{I}_3 & c_1\mathbf{I}_3 & c_2\mathbf{I}_3 & c_3\mathbf{I}_3 \end{pmatrix} \tag{5.15}$$

According to Eq. (5.13), the generalized forces vector would be $\mathbf{Q} = \mathbf{C_p}^\top\mathbf{F}$, but since vectors $\mathbf{v}_2$ and $\mathbf{v}_3$ are not part of the multibody model, additional transformations must be done in order to assemble the generalized forces vector.

$$\dot{W}^* = \mathbf{F}^\top\dot{\mathbf{r}}^* = \mathbf{F}^\top\left(\mathbf{C_p}\dot{\mathbf{q}}_0^*\right) = \mathbf{F}^\top\left(\mathbf{C_p}\bar{\mathbf{T}}\dot{\mathbf{q}}^*\right) \tag{5.16}$$

Identifying terms, the generalized forces vector corresponding to the external forces results as follows:

$$\mathbf{Q}_{ext} = \left(\mathbf{C_p}\bar{\mathbf{T}}\right)^\top\mathbf{F} \tag{5.17}$$

Velocity-dependent inertia forces in Eq. (5.12) are calculated by applying the Lagrange equations to the kinetic energy of the system. Let T be the kinetic energy of the system, which is calculated as follows:

$$T = \frac{1}{2}\dot{\mathbf{q}}^\top\mathbf{M}\dot{\mathbf{q}} \tag{5.18}$$

The Lagrange equation in dependent coordinates is as follows:

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\mathbf{q}}}\right) - \frac{\partial T}{\partial \mathbf{q}} + \mathbf{\Phi_q}^{\top}\boldsymbol{\lambda} = \mathbf{Q} \tag{5.19}$$

From this equation, the velocity-dependent terms yields:

$$\mathbf{Q}_v = \frac{1}{2}\left(\mathbf{M_q}\dot{\mathbf{q}}\right)^{\top}\dot{\mathbf{q}} - \dot{\mathbf{M}}\dot{\mathbf{q}} \tag{5.20}$$

where

$$\dot{\mathbf{M}} = \begin{pmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \frac{1}{2}I_1\dot{\mathbf{w}}_3 \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\frac{1}{2}I_1\dot{\mathbf{w}}_2 \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \frac{1}{2}I_1\dot{\mathbf{w}}_3^{\top} & -\frac{1}{2}I_1\dot{\mathbf{w}}_2^{\top} & 0 \end{pmatrix} \tag{5.21}$$

$$\mathbf{M_q}\dot{\mathbf{q}} = \begin{pmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \frac{1}{2}I_1\dot{\alpha}\mathbf{I}_3 & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\frac{1}{2}I_1\dot{\alpha}\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & -\frac{1}{2}I_1\dot{\mathbf{w}}_3^{\top} & \frac{1}{2}I_1\dot{\mathbf{w}}_2^{\top} & 0 \end{pmatrix} \tag{5.22}$$

Therefore,

$$\mathbf{Q}_v = \begin{pmatrix} \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} \\ -I_1\dot{\mathbf{w}}_3\dot{\alpha} \\ I_1\dot{\mathbf{w}}_2\dot{\alpha} \\ 0 \end{pmatrix} \tag{5.23}$$

5.3. Force models

Again, most of the force models were already present in [4], and are included in the Appendix A. However, some of them have been improved to achieve better results, since they are paramount in order to get accurate multibody simulations. All of the force models are described hereafter for the sake of clarity.

At low speed maneuvers, the most important forces are those coming from the engine and brakes, the tire forces, such as normal, longitudinal, and lateral forces, and the rolling resistance. The aerodynamic forces were not considered in this work, because they become important only at speeds higher than those considered in the tests performed in this thesis.

5.3.1. Drive and brake forces

The drive torque is applied to the rear wheels through an open differential, and the brake systems are identical on both rear wheels, thus it is assumed that the torque at both rear wheels is the same. A torque sensor was installed inside one of the rear wheels, so that no brake, engine nor gearbox models are needed for the rear wheels.

On the front wheels, the applied torque comes solely from the disc brake system, but the only available sensor is the brake pressure sensor, hence a brake model has to be applied. The brake model used is based on the tangential force model described in [59], but neglecting its viscous component, resulting in a brake torque T_b as follows:

$$T_b = \kappa T_{st} + (1 - \kappa)\, T_{sl} \tag{5.24}$$

where T_{st} is a spring-damper torque representing the stiction component of the brake torque, while T_{sl} is its the sliding component, and κ is a weighting factor which is velocity- dependent and provides a smooth transition from the stiction to the sliding situations:

$$\kappa = e^{-\left(\frac{|\dot{\alpha}|}{\dot{\alpha}_{ref}}\right)^2} \tag{5.25}$$

being $\dot{\alpha}_{ref}$ a parameter to adjust the transition from stiction to sliding. If the wheel rotates continuously, the stress of the spring would grow boundlessly. To avoid that, the stiction point should be updated when the value of the stiction torque exceeds the maximum brake torque available, which is the sliding torque. This torque can be computed as

$$T_{sl} = 2\mu F_n R_{eq} \tag{5.26}$$

where μ is the friction coefficient, F_n is the normal force between the disk and the pad, and R_{eq} is a the distance between wheel spin axis and the center of pressure of the pad, assuming constant pressure in between the pad and the disc. The normal force between the disc and the pad cannot be measured while driving the prototype in an obvious way. Instead, a pressure sensor was installed at the brake line. In order to know how the hydraulic pressure is transmitted to the brake pads, the caliper was removed and a load cell was put in between the pads (see Figure 5.6). With this experimental setup the brake pedal was stepped, while logging data from the load cell and the hydraulic pressure sensor. The force data was approximated with a cubic polynomial (see Figure 5.7), so the force made by the piston over the pad can be calculated knowing the hydraulic pressure P_h, given by the pressure sensor installed on the prototype, and the brake piston area A_{piston}.

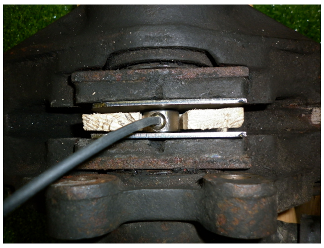


Figure 5.6: Load cell between the brake pads.

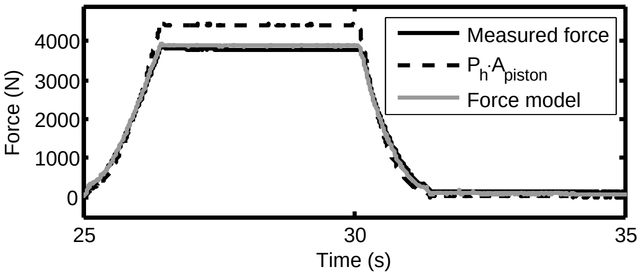


Figure 5.7: Force model for the brake pads.

After the normal force is determined, the only unknown is the friction coefficient between the disc and the pads. To estimate this coefficient, the wheel torque sensor was installed in one of the front wheels and some braking maneuvers were performed. Knowing the hydraulic brake pressure and the torque applied to the wheel, the friction coefficient can be easily determined using Eq. (5.26). This friction coefficient is $\mu = 0.35$ for the front brakes.

5.3.2. Tire forces

All the brake, drive and cornering forces are transmitted to the ground through the tires, so an accurate tire model should be used. Numerous mathematical models have been reported in the literature. They differ in accuracy and complexity depending on the areas of application. As described in [80], they can be classified in the following four different groups:

Simple tire models They are aimed at static and quasi-static vehicle dynamics analysis and at the design of vehicle control systems. Combined slip is not taken into account.

Empirical models They are based on non-linear mathematical approximations of tire forces and moments or interpolation of test data, like the famous versions (1987, 1989, 1993, 1996, 2002) of the Magic Formula [81, 82, 83, 84].

Physical models These models describe the kinematics and dynamics of the tire contact patch in detail.

Finite-element tire models The tire is modeled by a detailed finite element mesh for the complete tire structure including the compressed air. Almost any physical phenomenon can be taken into account. However, for vehicle dynamics analysis, the computational effort is yet too high for application on a regular basis.

The area of application of the tire model in this research is related to non-linear vehicle handling in real-time. As a consequence, the model should be as simple as possible in order to maintain low computational cost while fulfilling the requirements imposed by the test maneuvers, i.e. the tire model should be able to represent properly situations in which the wheel is blocked while driving, or maintaining the car stopped in an inclined road. Moreover, some dynamics behaviors of the tire, such as longitudinal and lateral deflections, are necessary in order to easily assess the equilibrium position of the vehicle at the beginning of the simulation while standing still. If the tire model does not manage these situations, it would be impossible to start the simulation if the initial position is not perfectly known for every degree of freedom.

In this work, part of the TMeasy tire model has been applied [85]. This is an empirical-physical model, in the sense that first, curve fitting using few parameters is necessary to adjust the tire characteristic curves and then, dynamical behaviors of the tire are considered. This model is similar to another recent empirical-physical model, the PAC2002 [80]. Both are used for low frequency applications. Some equations of the TMeasy model are reminded hereafter in order to explain how they have been combined with the multibody formulation.

5.3.2.1. Normal forces

The prototype is intended to be driven on paved surfaces. Therefore, the test track is modeled as a rigid triangle mesh, built with data taken from a topographical survey of the track. Moreover, the mesh is georeferenced in order to be able to use GPS measurements. More detail about the construction of the test track object is provided in section 5.4.

In this work the wheels are considered as spheres. This approximation is accurate for road driving. The contact detection routine is explained in [59]. After a wheel contact is detected, the normal force is applied as a spring-damper force. However, the force of the spring-damper element is limited such that it can produce compression forces over the road, but traction forces are not allowed.

5.3.2.2. Generalized tire force

Longitudinal and lateral tire forces, F_x and F_y, depend on the longitudinal and lateral slips, s_x and s_y respectively, which are calculated as follows:

$$s_x = \frac{-(v_x - r_D\Omega)}{r_D|\Omega|\hat{s}_x + v_n} \tag{5.27}$$

$$s_y = \frac{-v_y}{r_D|\Omega|\hat{s}_y + v_n} \tag{5.28}$$

where v_x and v_y are the longitudinal and lateral contact point velocities, r_D the dynamic radius of the wheel, Ω its angular velocity, $\hat{s}_x$ and $\hat{s}_y$ are two coefficients which allow to give more weight to the longitudinal or to the lateral slips when they are combined to provided the generalized slip, presented hereafter. A small fictitious velocity v_n is added to avoid the singularity when the wheel is not rotating. In this work, $v_n = 10^{-15}$ has been taken. This value avoids the singularity at $\Omega = 0$ without affecting the tire model behavior.

Longitudinal and lateral slips are combined to provide the generalized slip, s_g, as follows:

$$s_g = \sqrt{s_x^2 + s_y^2} \tag{5.29}$$

Once the generalized slip is known, the total tangential force F provided by the tire can be calculated if the tire characteristic is known. In this research, the exact characteristic of the tire was not available, and hence a linear approximation was used taking typical values from the literature [86]. The tire characteristic employed in this work is shown in Figure 5.8, where μ is the road-tire friction coefficient, F_n is the normal force to the road, and $s_{g,c}$ is the critical slip.

Once the total tangential force F is calculated, longitudinal and lateral tire forces, F_x and F_y, are obtained as follows:

$$F_x = F\frac{s_x}{s_g} \tag{5.30}$$

$$F_y = F\frac{s_y}{s_g} \tag{5.31}$$

These are the steady-state forces developed by the tire. However, when the force varies, the deflection of the tire have to be taken into consideration, as explained in the next section.

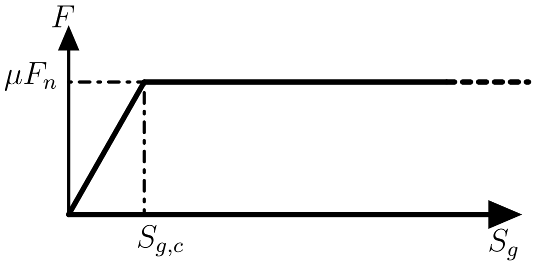


Figure 5.8: Linerized tire characteristic.

5.3.2.3. TMeasy first order tire dynamics

As mentioned before, the effects of tire deflection affect the tire behavior when the forces vary. This phenomenon is specially important when the vehicle moves at very low speed, when small displacements can produce high slips. If tire deflection is not considered, these high slips introduce unrealistic high forces. What actually happens is that the forces acting in the contact patch deflect longitudinally and laterally the tire as shown in Figures 5.9 and 5.10.

According to the *TMeasy* model, on the one hand, the tire forces F_x and F_y can be reasonably represented by the first order approximations shown in Eqs. (5.32) and (5.33).

$$\underbrace{\mathrm{F_x}(\nu_\mathrm{x}+\dot{\mathrm{x}}_\mathrm{e})}_{\mathrm{F_x^D}} \simeq F_x(\nu_x) + \frac{\partial F_x}{\partial \nu_x}\left[(\nu_x+\dot{x}_e)-\nu_x\right] = \underbrace{\mathrm{F_x}}_{\mathrm{F_x^S}} + \frac{\partial F_x}{\partial \nu_x}\dot{x}_e \tag{5.32}$$

$$\underbrace{\mathrm{F_y}(\nu_\mathrm{y}+\dot{\mathrm{y}}_\mathrm{e})}_{\mathrm{F_y^D}} \simeq F_y(\nu_y) + \frac{\partial F_y}{\partial \nu_y}\left[(\nu_y+\dot{y}_e)-\nu_y\right] = \underbrace{\mathrm{F_y}}_{\mathrm{F_y^S}} + \frac{\partial F_y}{\partial \nu_y}\dot{y}_e \tag{5.33}$$

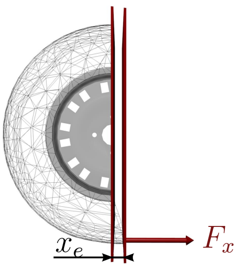


Figure 5.9: Longitudinal tire deflection due to the contact forces.

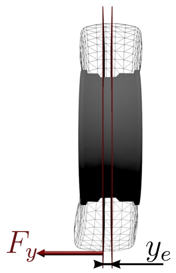


Figure 5.10: Lateral tire deflection due to the contact forces.

where $\dot{\mathrm{x}}_\mathrm{e}$ and $\dot{\mathrm{y}}_\mathrm{e}$ are the longitudinal and lateral tire deflection velocities, F_x^D and F_y^D are the dynamic tire forces, F_x^S and F_y^S are the steady-state tire forces and ν_x and ν_y are the longitudinal and lateral velocities of the contact point. On the other hand, the tire dynamic forces can also be calculated by considering that the tire consists of two spring-damper elements, one in the longitudinal direction and the other in the lateral direction. Both equations are presented in Eqs. (5.34) and (5.35).

$$\mathrm{F_x^D} \simeq \mathrm{c_x x_e} + \mathrm{d_x \dot{x}_e} \tag{5.34}$$

$$\mathrm{F_y^D} \simeq \mathrm{c_y y_e} + \mathrm{d_y \dot{y}_e} \tag{5.35}$$

where $\mathrm{c_x}$, $\mathrm{c_y}$, $\mathrm{d_x}$, $\mathrm{d_y}$ are the stiffness and damping parameters for the longitudinal and lateral tire deflections. After that, Eq. (5.32) can be combined with Eq. (5.34) to yield a first order differential equation, shown in Eq. (5.36) for the longitudinal deflection of the tire. The first order differential equation for the lateral deflection, shown in Eq. (5.37), is obtained using Eqs. (5.33) and (5.35).

$$\left(\nu_{\mathrm{x}}^{*}\mathrm{d_x} + \frac{\partial \mathrm{F_x^S}}{\partial \mathrm{s_x}}\right)\mathrm{\dot{x}_e} = -\frac{\mathrm{F}}{\mathrm{s_g}}(\nu_{\mathrm{x}} - \mathrm{r_D}\Omega) - \nu_{\mathrm{x}}^{*}\mathrm{c_x x_e} \tag{5.36}$$

$$\left(\nu_{\mathrm{y}}^{*}\mathrm{d_y} + \frac{\partial \mathrm{F_y^S}}{\partial \mathrm{s_y}}\right)\mathrm{\dot{y}_e} = -\frac{\mathrm{F}}{\mathrm{s_g}}\nu_{\mathrm{y}} - \nu_{\mathrm{y}}^{*}\mathrm{c_y y_e} \tag{5.37}$$

where Ω is the angular velocity of the wheel, $\nu_{\mathrm{x}}^{*} = \mathrm{r_D}|\Omega|\mathrm{\hat{s}_x} + \nu_{\mathrm{n}}$, $\nu_{\mathrm{y}}^{*} = \mathrm{r_D}|\Omega|\mathrm{\hat{s}_y} + \nu_{\mathrm{n}}$ and $\mathrm{s_g}$ is the generalized slip. Both differential equations have to be integrated at each time step to calculate the tire forces by means of Eqs. (5.34) and (5.35). To this end, in this research, both differential equations are solved for every iteration (corresponding to a time step) of integration of the multibody simulation. As the variables of Eqs. (5.36) and (5.37) are constant during each iteration (except x_e, $\dot{x}_e$, y_e, $\dot{y}_e$,), both equations become first order differential equations with constant coefficients. Their analytic solutions are presented in Eqs. (5.38) and (5.39).

$$\begin{array}{ll} \mathrm{x_e} = \dfrac{\mathrm{a_2}}{\mathrm{a_1}} + \mathrm{a_3}\,\mathrm{e}^{-\mathrm{a_1}\Delta \mathrm{t_i}} & a_1 = \dfrac{\nu_x^{*} c_x}{\left(\nu_x^{*} d_x + \dfrac{\partial F_x}{\partial s_x}\right)} \\ a_2 = -\dfrac{\frac{F}{s_g}(\nu_x - r_D\Omega)}{\left(\nu_x^{*} d_x + \dfrac{\partial F_x}{\partial s_x}\right)} & a_3 = x_{e0} - \dfrac{a_2}{a_1} \end{array} \tag{5.38}$$

$$\begin{array}{ll} \mathrm{y_e} = \dfrac{\mathrm{b_2}}{\mathrm{b_1}} + \mathrm{b_3}\,\mathrm{e}^{-\mathrm{b_1}\Delta \mathrm{t_i}} & b_1 = \dfrac{\nu_y^{*} c_y}{\left(\nu_y^{*} d_y + \dfrac{\partial F_y}{\partial s_y}\right)} \\ b_2 = -\dfrac{\frac{F}{s_g\,\nu_y}}{\left(\nu_y^{*} d_y + \dfrac{\partial F_y}{\partial s_y}\right)} & b_3 = y_{e0} - \dfrac{b_2}{b_1} \end{array} \tag{5.39}$$

where $\mathrm{x_{e0}}$ and y_{e0} are the initial longitudinal and lateral tire deflection for each integration time step. Consequently, for each iteration of an integration time step, x_e and y_e are calculated using Eqs. (5.38) and (5.39), and $\dot{x}_e$ and $\dot{y}_e$ are obtained by means of Eqs. (5.36) and (5.37). When the integration time step changes, x_{e0} and y_{e0} must be updated with the latest values of x_e and y_e. Solving the differential equations analytically at each iteration allows calculating efficiently and accurately the tire forces.

5.3.2.4. TMeasy blocked wheel with tire sliding

A small modification has to be made to the previous equations to allow the tire model to take into account situations in which the wheel is blocked but the tire is sliding. Although this situation is not common in normal driving conditions, it occurs much more frequently in simulation. For example, at the beginning of a simulation, an initial location and orientation are given to the vehicle and then, it is dropped on the test track. Subsequently, the wheel rotation velocities are zero and, longitudinal and lateral slips appear due to the suspension deflections and the test track inclination. When a tire is sliding, for example laterally, $\mathrm{y_e}$ increases until it reaches a value which should be maintained during the whole sliding. In such a situation, as $\Omega = 0$, Eq. (5.39) becomes Eq. (5.40). It can be seen from Eq. (5.40) that $e^{-b_1 \Delta t_i}$ should be equal to 1 to get $y_e = y_{e0}$. This means that the tire lateral deflection during the iterations of the Newton-Raphson method y_e is equal to the tire lateral deflection at the beginning of the corresponding integration time step y_{e0}, and so on for the next integration time steps until the tire stops sliding. $\mathrm{e^{-b_1 \Delta t_i}} \simeq 1$ implies that $-b_1 \Delta t_i \simeq 0$. As $\left.\dfrac{\partial F_y}{\partial s_y}\right|_{s_g > s_c} = 0$, it follows that $b_1 = \dfrac{c_y}{d_y}$. Unfortunately, after having substituted $\mathrm{c_y}$, $\mathrm{d_y}$ and $\mathrm{\Delta t_i}$, $-b_1 \Delta t_i \neq 0$. Therefore, Eqs. (5.38) and (5.39) have to be modified in order to maintain $\mathrm{y_e}$ constant during the sliding. In the *TMeasy* model, it has been chosen to change the derivative as shown in Eq. (5.41). In this way, when $\mathrm{s_g} < \mathrm{s_c}$ ($\mathrm{s_c}$ is the critical slip), the correct derivative is well approximated (if linear tire curves are used, the derivative is exact) and, when $\mathrm{s_g} > s_c$, the derivative is largely greater than zero, thus making $-\mathrm{b_1 \Delta t_i} \simeq 0$. The same modification has to be made to the derivative of the longitudinal tire forces.

$$\mathrm{y_e} = \frac{\mathrm{b_2}}{\mathrm{b_1}} + \left(\mathrm{y_{e0}} - \frac{\mathrm{b_2}}{\mathrm{b_1}}\right) \mathrm{e^{-b_1 \Delta t_i}}$$

$$b_1 = \frac{\nu_n c_y}{\left(\nu_n d_y + \left.\dfrac{\partial F_y}{\partial s_y}\right|_{s_g > s_c}\right)} \qquad b_2 = -\frac{\frac{F}{s_g}\,\nu_y}{\left(\nu_n d_y + \left.\dfrac{\partial F_y}{\partial s_y}\right|_{s_g > s_c}\right)} \tag{5.40}$$

$$\frac{\partial \mathrm{F_y}}{\partial \mathrm{s_y}} \rightarrow \frac{\mathrm{F_y}}{\mathrm{s_y}} = \mathrm{F_y}\frac{1}{\mathrm{s_y}} = \frac{\mathrm{F}}{\mathrm{s_g}}\mathrm{s_y}\frac{1}{\mathrm{s_y}} = \frac{\mathrm{F}}{\mathrm{s_g}} \tag{5.41}$$

5.3.3. Rolling resistance

Rolling resistance is one of the most important road loads. In fact, it is the most important force while driving at low speed on level and paved roads. Its effect is surpassed by the aerodynamic forces only at speeds higher than about 25 m/s [86]. Although rolling resistance is a force produced in the tread of the tire, in this work it was added as a brake torque, following the next equation,

$$T_{rr} = C_{rr} N R_w \tag{5.42}$$

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

$$C_{rr} = \frac{a_r - a_t}{g\cos(\gamma)} \tag{5.43}$$

Being a_r the acceleration of the prototype, a_t the theoretical acceleration, g the gravity acceleration, and γ the angle of the ramp with the horizontal plane (see Figure 5.11). Wheel misalignments lead to an increment of the rolling resistance, and the brakes also produces some friction. These phenomena are already considered in the model of the vehicle. In order not to consider them twice, the theoretical acceleration was determined by performing with the complete multibody model the same maneuver that was done with the prototype, but without taking into account the rolling resistance forces. The rolling resistance coefficient obtained from this test was $C_{rr} = 0.017$.

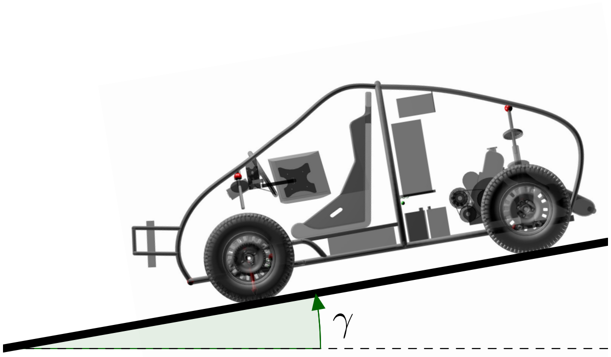


Figure 5.11: Schema of the rolling resistance experiment.

5.4. Simulation environment

In this thesis, the simulation environment developed in [4] was already available (see Appendix A). This environment was made from a topographical survey made with a total station to represent the geometry of the track properly. However, the test track previously employed was not suitable for the present work due to the sky visibility, highly limited by the surrounding trees and buildings, which prevented the GPS receiver from working accurately.

Consequently, a different test track was employed, and hence its computational model had to be made, following a methodology similar to that used in [4]. This time, a car park in A Malata (Ferrol, Spain) was employed. The topographical survey was made with the GPS, which provides up to 1 cm horizontal and 2 cm vertical accuracy.

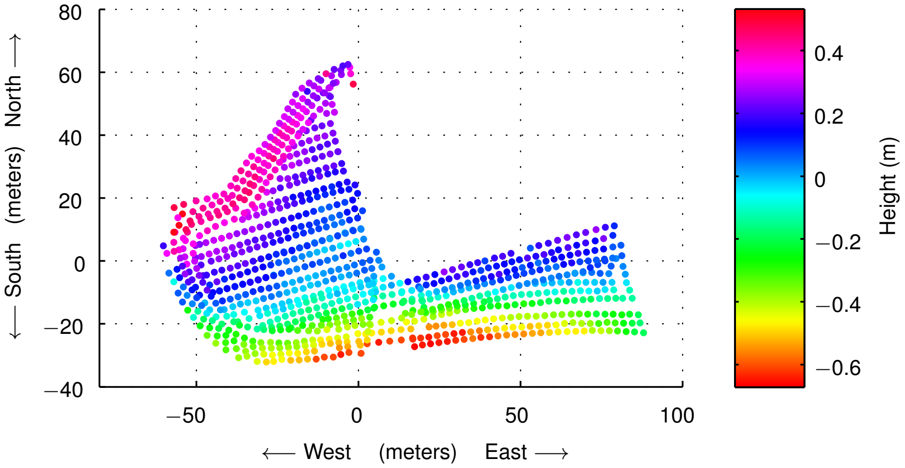


Figure 5.12: Points taken during the topographical survey, expressed in the ENU system.

The total number of points measured was 957, as shown in Figure 5.12. In order to build the triangular mesh needed to generate the ground of the simulation, a new set of points at a distance of 1 meter was obtained by means of the natural neighbor interpolation. Finally, a constrained Delaunay triangularization was applied to obtain the final mesh shown in Figure 5.13.

To make the graphical object of the ground, shown in Figure 5.14, a tarmac texture was added to the mesh, and the lines of the car park were drawn using an ortophoto of the test track, Figure 5.15

5.5. On board implementation of the multibody model

This multibody model has been installed on board the prototype, and it can be run while driving the vehicle. Executing a complex multibody model in real time is challenging. Moreover, in this case, the program should be able to deal with the different sampling rates of the sensors and the integration frequency of the multibody model. For these reasons, a multithread program was developed. This structure allows to deal with the different sampling rates easily, and, furthermore, it takes advantage of the multicore processors available nowadays in personal computers.

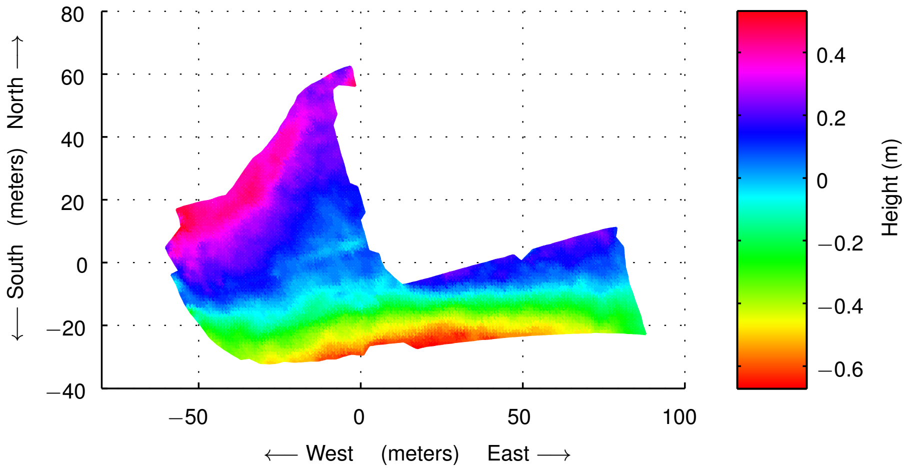


Figure 5.13: Triangle mesh build from the topographical survey, expressed in the ENU system.

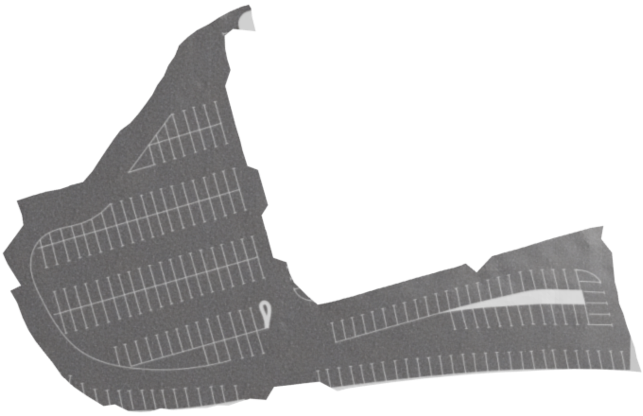


Figure 5.14: Graphical object used in the simulations.

For the sake of efficiency, the core of the multibody simulation was programmed in Fortran 2003, while other parts, such as the graphical output, the DAP interface, or the GPS interface were programmed in C++.

The flow of the program can be seen in Figure 5.16. First, the initialization phase is performed. During this phase, the program reads a file where some parameters can be selected: the tolerance during the integration of the dynamical equations, the scenario, etc. After this configuration process, the initial position is set. In order to do that, the vehicle should be stopped while the program starts, and a measurement is taken with the GPS. After that, the program requires the vehicle to be driven a short distance in a straight line and then stopped again, so the GPS takes another measurement. With these two measurements, the initial position and attitude of the vehicle can be determined.

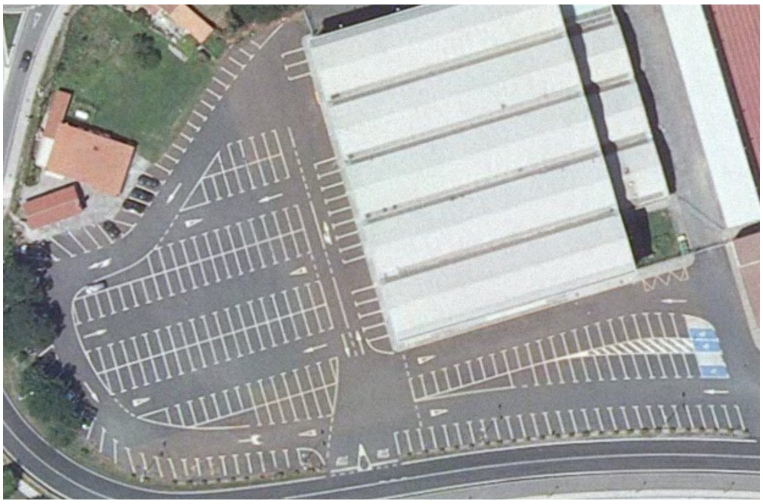


Figure 5.15: Orthophoto of the test track (PNOA Ortophoto, courtesy of IGN).

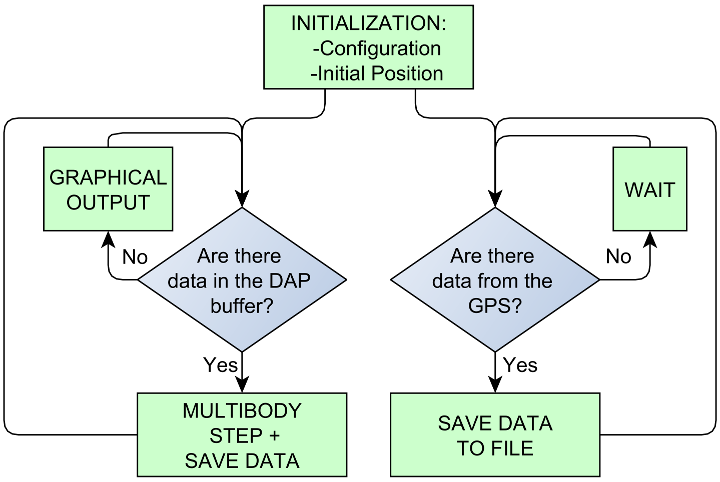


Figure 5.16: Flowchart of the program.

After the initialization is performed, the DAP and the GPS interfaces start gathering data in two separated threads. While the DAP sampling works at 500 Hz, the GPS measurements are produced and parsed at 50 Hz. The sampling process is carried out by the DAP board. Since the DAP runs a real-time operating system, the sampling process is used as the time reference, so that the multibody simulation (which runs at 250 Hz) performs an integration step every two samples of the DAP. If the multibody simulation is ready to take an integration step, but the DAP has not gathered two samples yet, then the 3D graphical output is generated. Although the graphical output is not strictly necessary in this application, it is useful for verifying the behavior of the simulation while driving. With this structure, the execution of the multibody model has priority over the graphical output.

The whole state observer was not implemented yet on board the vehicle. However, from the multibody model, already implemented, it should be little work to get to the full implementation of the state observer.

5.6. Sensor models

From all the sensors installed on the vehicle (see Table 5.1), some of them are used to feed the force models, such as the brake model. Other sensors are not used in the multibody model, but they can be used as measurements in a Kalman filter to correct the state of the multibody simulation. The models of such sensors are developed hereafter. The Jacobian matrices of these models with respect to the generalized coordinates $\mathbf{q}$ and velocities $\dot{\mathbf{q}}$ are also developed henceforth. From these Jacobians, the Jacobian matrices with respect the states of the filter are calculated following Eqs. (3.120) and (3.121).

5.6.1. GPS position model

The GPS receiver provides position and velocity measurements, but in this section only the position information is approached. The chassis of the prototype is modeled by using a point and three non-coplanar unit vectors. Although there are more points and vector in the actual chassis model, they are not considered here, since they are not necessary to develop the measurement model of the GPS. Let $\mathbf{r}_1$ be the position vector of point $\mathbf{p}_1$, and $\mathbf{v}_1$, $\mathbf{v}_2$, and $\mathbf{v}_3$ three orthonormal vectors. Then, the vector part of interest of the $\mathbf{q}$ vector results:

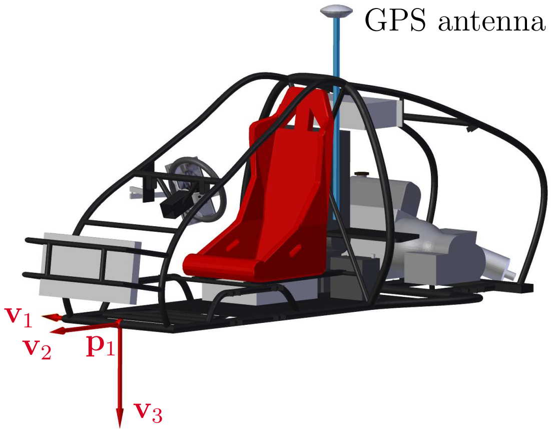


Figure 5.17: GPS measurement model.

$$\mathbf{q} = \begin{bmatrix} \dots & \mathbf{r}_1^\top & \mathbf{v}_1^\top & \mathbf{v}_2^\top & \mathbf{v}_3^\top & \dots \end{bmatrix}^\top \tag{5.44}$$

The GPS is fixed to the chassis of the vehicle, and hence the position of its antenna can be written as a linear combination of the point and the vectors of the chassis with constant coefficients, resulting the following measurement model:

$$\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{r}_{GPS} = \mathbf{r}_1 + K_1\mathbf{v}_1 + K_2\mathbf{v}_2 + K_3\mathbf{v}_3 \tag{5.45}$$

where K_1, K_2, and K_3 are the coefficients of the linear combination. Then, the partial derivatives with respect to the positions are straightforward:

$$\mathbf{h_q} = \frac{\partial \mathbf{r}_{GPS}}{\partial \mathbf{q}} = \begin{bmatrix} \mathbf{0} & \dots & \mathbf{I}_3 & K_1\mathbf{I}_3 & K_2\mathbf{I}_3 & K_3\mathbf{I}_3 & \dots & \mathbf{0} \end{bmatrix} \tag{5.46}$$

However, some terms of this Jacobian matrix are neglected because they produce undesirable effects: the terms related with the position of the GPS antenna produce corrections in both position and orientation of the chassis. However, the desired effect is to correct only its position, but not the orientation. Therefore, the Jacobian finally employed is the next one:

$$\mathbf{h_q} = \frac{\partial \mathbf{r}_{GPS}}{\partial \mathbf{q}} = \begin{bmatrix} \mathbf{0} & \dots & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & \mathbf{0} \end{bmatrix} \tag{5.47}$$

The partial derivative of the measurement model of the GPS positioning with respect to velocity is null:

$$\mathbf{h_{\dot{q}}} = \frac{\partial \mathbf{r}_{GPS}}{\partial \dot{\mathbf{q}}} = \begin{bmatrix} \mathbf{0} & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & \mathbf{0} \end{bmatrix} \tag{5.48}$$

5.6.2. GPS velocity model

GPS receivers provide velocity measurements in addition to the position data. Velocity measurements are independent from position measurements because they are not obtained by derivation, but measuring the frequency of the signals received from the satellites. These frequencies vary when the relative velocities between the receiver and the satellites change due to the Doppler effect. Velocity measurements are usually better than position measurements in single antenna GPSs. However, velocity measurements do not improve when differential GPS techniques are employed.

Although the velocity is measured in three-dimensional space, the output protocol of the GPS only provides a speed measurement in the plane, and the angle of the trajectory with respect to the north, usually called *course over ground* . This measurements can be easily converted to Cartesian coordinates, thus providing x and y components of the velocity. The measurement model is as follows:

$$\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \dot{\mathbf{r}}_{GPS} = \begin{bmatrix} \dot{\mathbf{r}}_{1x} \\ \dot{\mathbf{r}}_{1y} \end{bmatrix} + K_1 \begin{bmatrix} \dot{v}_{1x} \\ \dot{v}_{1y} \end{bmatrix} + K_2 \begin{bmatrix} \dot{v}_{2x} \\ \dot{v}_{2y} \end{bmatrix} + K_3 \begin{bmatrix} \dot{v}_{3x} \\ \dot{v}_{3y} \end{bmatrix} \tag{5.49}$$

The Jacobian matrices with respect to positions $\mathbf{q}$ and velocities $\dot{\mathbf{q}}$ result:

$$\mathbf{h_q} = \frac{\partial \dot{\mathbf{r}}_{GPS}}{\partial \mathbf{q}} = \begin{bmatrix} \mathbf{0} & \dots & \mathbf{0}_{2\times3} & \mathbf{0}_{2\times3} & \mathbf{0}_{2\times3} & \mathbf{0}_{2\times3} & \dots & \mathbf{0} \end{bmatrix} \tag{5.50}$$

$$\mathbf{h_{\dot{q}}} = \frac{\partial \dot{\mathbf{r}}_{GPS}}{\partial \dot{\mathbf{q}}} = \begin{bmatrix} 0 & \dots & 1 & 0 & 0 & K_1 & 0 & 0 & K_2 & 0 & 0 & K_3 & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & 1 & 0 & 0 & K_1 & 0 & 0 & K_2 & 0 & 0 & K_3 & 0 & \dots & 0 \end{bmatrix} \tag{5.51}$$

During normal driving, the course over ground is similar to the yaw angle of the vehicle. Therefore, the course over ground is employed as a yaw measurement. This is specially exact if the vehicle goes straight. When the vehicle is sliding or undergoes sharp turns, the approximation is less accurate, but it is still useful to achieve the observability of the yaw angle.

5.6.3. Angular rate measurements

The vehicle prototype used in this thesis has an inertial measurement unit (IMU), which provides angular rate and linear acceleration measurements. This section deals with the angular rate measurement model. In order to obtain the measurement model of the gyroscopes, the angular velocity in their local axes must be computed.

Let us consider a coordinate system $Oxyz$, and a rigid body represented by the system $O'x'y'z'$, as shown in Figure 5.18. The point P belongs to the rigid body represented by the system $O'x'y'z'$, and its position can be written as follows [53]:

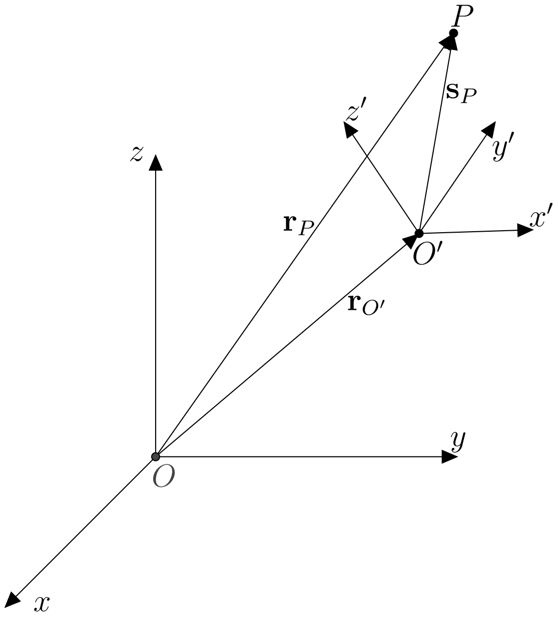


Figure 5.18: Vector diagram of the point of a solid in space.

$$\mathbf{r}_P = \mathbf{r}_{O'} + \mathbf{s}_P = \mathbf{r}_{O'} + \boldsymbol{\Psi}\mathbf{s}'_P \tag{5.52}$$

where $\boldsymbol{\Psi}$ is the rotation matrix of the rigid body, and $\mathbf{s}'_P$ is the vector $\mathbf{s}_P$ expressed in the $O'x'y'z'$ system.

The velocity of point P can be expressed as follows:

$$\dot{\mathbf{r}}_P = \dot{\mathbf{r}}_{O'} + \dot{\boldsymbol{\Psi}}\mathbf{s}'_P = \dot{\mathbf{r}}_{O'} + \dot{\boldsymbol{\Psi}}\boldsymbol{\Psi}^\top\mathbf{s}_P = \dot{\mathbf{r}}_{O'} + \boldsymbol{\Omega}\mathbf{s}_P \tag{5.53}$$

where $\boldsymbol{\Omega}$ is the angular velocity tensor of the solid $O'x'y'z'$:

$$\boldsymbol{\Omega} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{5.54}$$

Eq. (5.53) can be written as function of the local angular velocity tensor $\mathbf{\Omega}'$, as follows:

$$\dot{\mathbf{r}}_P = \dot{\mathbf{r}}_{O'} + \mathbf{\Psi}\mathbf{\Omega}'\mathbf{s}'_P \tag{5.55}$$

From Eqs. (5.53) and (5.55), the angular velocity tensor expressed in the local coordinates of the solid $O'x'y'z'$ can be obtained:

$$\mathbf{\Psi}\mathbf{\Omega}'\mathbf{s}'_P = \mathbf{\Omega}\mathbf{s}_P \Rightarrow \mathbf{\Omega}' = \mathbf{\Psi}^\top\dot{\mathbf{\Psi}} \tag{5.56}$$

Assuming the chassis model employed in the previous sections, the position and velocity vectors, $\mathbf{q}$ and $\dot{\mathbf{q}}$, are expressed as follows (see Figure 5.19):

$$\mathbf{q} = \begin{bmatrix} \dots & \mathbf{r}_1^\top & \mathbf{v}_1^\top & \mathbf{v}_2^\top & \mathbf{v}_3^\top & \dots \end{bmatrix}^\top \tag{5.57}$$

$$\dot{\mathbf{q}} = \begin{bmatrix} \dots & \dot{\mathbf{r}}_1^\top & \dot{\mathbf{v}}_1^\top & \dot{\mathbf{v}}_2^\top & \dot{\mathbf{v}}_3^\top & \dots \end{bmatrix}^\top \tag{5.58}$$

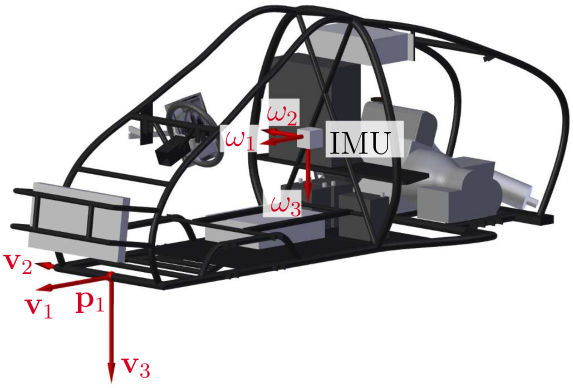


Figure 5.19: Agular rates of the IMU.

From these coordinate and velocity vectors, the rotation matrix of the chassis $\mathbf{\Psi}_{CH}$, and its derivative $\dot{\mathbf{\Psi}}_{CH}$ result:

$$\mathbf{\Psi}_{CH} = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \end{bmatrix} \tag{5.59}$$

$$\dot{\mathbf{\Psi}}_{CH} = \begin{bmatrix} \dot{\mathbf{v}}_1 & \dot{\mathbf{v}}_2 & \dot{\mathbf{v}}_3 \end{bmatrix} \tag{5.60}$$

Applying Eq. (5.56) to the model of the chassis, and writing its angular rate vector $\boldsymbol{\omega}_{CH}$ in local coordinates yields:

$$\boldsymbol{\omega}_{CH} = \begin{bmatrix} \mathbf{v}_3^\top\dot{\mathbf{v}}_2 & \mathbf{v}_1^\top\dot{\mathbf{v}}_3 & \mathbf{v}_2^\top\dot{\mathbf{v}}_1 \end{bmatrix}^\top \tag{5.61}$$

If the IMU is not aligned with the reference frame of the chassis, an additional rotation matrix $\mathbf{\Psi}_{IMU}$ is needed. The columns of this rotation matrix are the axes of the IMU expressed in the local coordinates of the chassis, and hence it is a constant matrix. Then, the angular rates measured by the gyroscopes $\boldsymbol{\omega}_{IMU}$ (see Figure 5.19) can be written as follows:

$$\mathbf{h}(\mathbf{q},\dot{\mathbf{q}}) = \boldsymbol{\omega}_{IMU} = \begin{bmatrix} \omega_1 & \omega_2 & \omega_3 \end{bmatrix}^\top = \mathbf{\Psi}_{IMU}^\top \begin{bmatrix} \mathbf{v}_3^\top\dot{\mathbf{v}}_2 & \mathbf{v}_1^\top\dot{\mathbf{v}}_3 & \mathbf{v}_2^\top\dot{\mathbf{v}}_1 \end{bmatrix}^\top \tag{5.62}$$

The Jacobian matrices of the measurement models with respect to the coordinates $\mathbf{q}$ and velocities $\dot{\mathbf{q}}$ yield:

$$\mathbf{h}_{\mathbf{q}} = \frac{\partial \boldsymbol{\omega}_{IMU}}{\partial \mathbf{q}} = \boldsymbol{\Psi}_{IMU}^{\top} \begin{bmatrix} \mathbf{0} & \dots & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \dot{\mathbf{v}}_2^{\top} & \dots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0}_{1\times3} & \dot{\mathbf{v}}_3^{\top} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \dots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \dot{\mathbf{v}}_1^{\top} & \mathbf{0}_{1\times3} & \dots & \mathbf{0} \end{bmatrix} \tag{5.63}$$

$$\mathbf{h}_{\dot{\mathbf{q}}} = \frac{\partial \boldsymbol{\omega}_{IMU}}{\partial \dot{\mathbf{q}}} = \boldsymbol{\Psi}_{IMU}^{\top} \begin{bmatrix} \mathbf{0} & \dots & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{v}_3^{\top} & \mathbf{0}_{1\times3} & \dots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{v}_1^{\top} & \dots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0}_{1\times3} & \mathbf{v}_2^{\top} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \dots & \mathbf{0} \end{bmatrix} \tag{5.64}$$

5.6.4. Acceleration measurements

The observers considered in this thesis do not include accelerations as part of their states. Therefore, acceleration information cannot be used to correct the states, since the acceleration is considered constant at a given time step. However, accelerometers cannot measure gravity acceleration (see e.g.: [87]). This fact is considered when building the measurement model of the accelerometers, and hence they provide some orientation information, useful to stabilize the magnitudes related to the roll and pitch motions. An schematic of the accelerometers is shown in Figure 5.20.

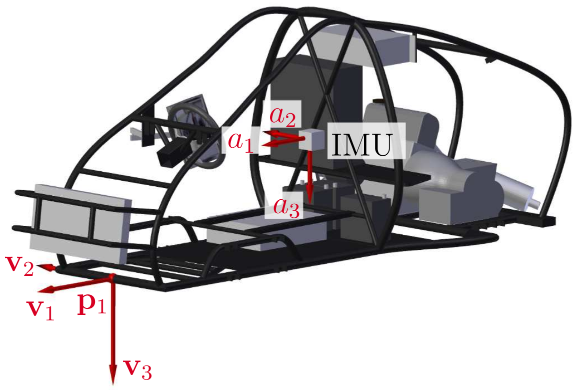


Figure 5.20: Linear accelerations of the IMU.

In order to build the measurement model of the accelerometer, the gravity acceleration is removed from the acceleration of the point where the IMU is located, and expressed in global coordinates:

$$\mathbf{a}_{IMU}^{global} = \ddot{\mathbf{r}}_1 + K_1\ddot{\mathbf{v}}_1 + K_2\ddot{\mathbf{v}}_2 + K_3\ddot{\mathbf{v}}_3 - \mathbf{g} \tag{5.65}$$

where K_1, K_2 and K_3 represent the coefficients of liner combination of the IMU in the chassis frame, and $\mathbf{g}$ is the acceleration of gravity. After the acceleration sensed by the accelerometers is calculated, it is expressed in the IMU local axes by means of the rotation matrices of the chassis, $\boldsymbol{\Psi}_{CH}$, and the rotation matrix from the chassis to the IMU, $\boldsymbol{\Psi}_{IMU}$, defined in the previous section:

$$\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{a}_{IMU} = \begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix}^{\top} = \boldsymbol{\Psi}_{IMU}^{\top} \boldsymbol{\Psi}_{CH}^{\top} \mathbf{a}_{IMU}^{global} \tag{5.66}$$

The Jacobian matrices of this measurement model with respect to $\mathbf{q}$ and $\dot{\mathbf{q}}$ yield:

$$\mathbf{h_q} = \frac{\partial \mathbf{a}_{IMU}}{\partial \mathbf{q}} = \mathbf{\Psi}_{IMU}^{\top} \begin{bmatrix} \mathbf{0} & \dots & \mathbf{0}_{1\times3} & \mathbf{a}_{IMU}^{global} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \dots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{a}_{IMU}^{global} & \mathbf{0}_{1\times3} & \dots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{a}_{IMU}^{global} & \dots & \mathbf{0} \end{bmatrix} \tag{5.67}$$

$$\mathbf{h_{\dot{q}}} = \frac{\partial \mathbf{a}_{IMU}}{\partial \dot{\mathbf{q}}} = \mathbf{\Psi}_{IMU}^{\top} \begin{bmatrix} \mathbf{0} & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & \mathbf{0} \end{bmatrix} \tag{5.68}$$

5.6.5. Wheel angle measurements

Wheel angles of the vehicle are measured with Hall effect sensors. The sensors are fixed to the knuckles, and aiming at the disc brakes, where 40 wholes were drilled, providing a resolution of 9 degrees. The direction of rotation cannot be determined with this system, but in this thesis forward motion is always assumed.

The angles of the wheels are part of the coordinate vector, thus the measurement model and its Jacobian matrices with respect to position and velocities are straightforward:

$$\mathbf{q} = \begin{bmatrix} \dots & \theta_1 & \theta_2 & \theta_3 & \theta_4 & \dots \end{bmatrix} \tag{5.69}$$

$$\dot{\mathbf{q}} = \begin{bmatrix} \dots & \dot{\theta}_1 & \dot{\theta}_2 & \dot{\theta}_3 & \dot{\theta}_4 & \dots \end{bmatrix} \tag{5.70}$$

$$\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 & \theta_4 \end{bmatrix}^{\top} \tag{5.71}$$

$$\mathbf{h_q} = \begin{bmatrix} \mathbf{0} & \dots & \mathbf{I}_4 & \dots & \mathbf{0} \end{bmatrix} \tag{5.72}$$

$$\mathbf{h_{\dot{q}}} = \begin{bmatrix} \mathbf{0} & \dots & \mathbf{0}_{4\times4} & \dots & \mathbf{0} \end{bmatrix} \tag{5.73}$$

In this case, as the angles θ_1, θ_2, θ_3 and θ_4 are degrees of freedom of the model, the Jacobian matrices with respect to the independent coordinates and velocities, $\mathbf{h_z}$ and $\mathbf{h_{\dot{z}}}$, are constant and have expressions analogous to the Jacobian matrices with respect to the dependent coordinates and velocities, $\mathbf{h_q}$ and $\mathbf{h_{\dot{q}}}$.

5.7. State observer

The study in chapter 4 shows that the errorEKF method was the fastest among all the methods tested. Moreover, due to its structure, it can be applied over an existing multibody model with only few modifications to the code. Although the UKF methods share this advantage, and they are even easier to implement because the calculation of the Jacobian matrices of the measurement models is avoided, their computational cost make this family of methods impractical to implement a state observer based on a multibody model with several degrees of freedom. Consequently, the errorEKF method was selected to be implemented with the model of the vehicle.

5.7.1. Design of the observer

The multibody model of the vehicle has 14 degrees of freedom: the six degrees of freedom of a rigid body in space for the chassis, the motion of the suspensions, and the rotation angle of the wheels. The steering is not a degree of freedom because it has been kinematically guided.

When the measurements provided by the sensors available at the vehicle are examined, it becomes apparent that the motion of the chassis and the wheel angles are observable, but the displacements of the suspensions are not. Therefore, the non-observable degrees of freedom are not included in the state vector of filter, and they behave according to the dynamics predicted by the multibody model, without corrections from the Kalman filter. However, corrections to other variables can affect the behavior of these variables trough their effect on the dynamics of the model.

The state of the the observer of the vehicle results as follows:

$$\mathbf{x} = \begin{bmatrix} \Delta r_{1x} & \Delta r_{1y} & \Delta r_{1z} & \Delta\psi & \Delta v_{1z} & \Delta v_{2z} & \Delta\theta_1 & \Delta\theta_2 & \Delta\theta_3 & \Delta\theta_4 & \dots \\ \Delta\dot{r}_{1x} & \Delta\dot{r}_{1y} & \Delta\dot{r}_{1z} & \Delta\dot{\psi} & \Delta\dot{v}_{1z} & \Delta\dot{v}_{2z} & \Delta\dot{\theta}_1 & \Delta\dot{\theta}_2 & \Delta\dot{\theta}_3 & \Delta\dot{\theta}_4 \end{bmatrix}^\top \tag{5.74}$$

Where $\mathbf{r}_1 = \begin{bmatrix} r_{1x} & r_{1y} & r_{1z} \end{bmatrix}^\top$ is the position vector of point $\mathbf{p}_1$, ψ is the yaw angle of the vehicle, v_{1z} and v_{2z} are the vertical components of vectors $\mathbf{v}_1$ and $\mathbf{v}_2$, which represent the degrees of freedom of the pitch and roll angles respectively, and θ_1, θ_2, θ_3 and θ_4 are the wheel angles, as shown in Figure 5.21.

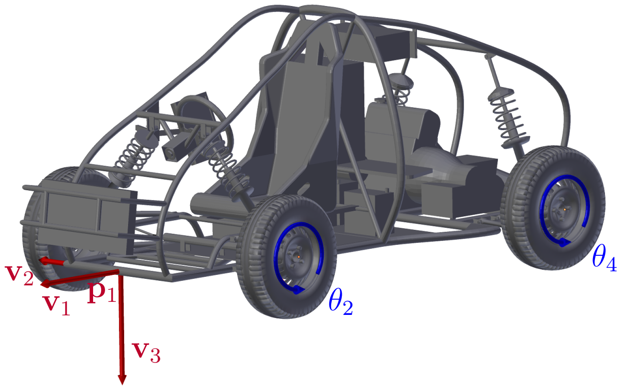


Figure 5.21: Main variables of the multibody model of the vehicle.

As the method employed is the errorEKF, the states are the errors of the multibody model at position and velocity level, such that the final estimation of a magnitude is the prediction provided by the multibody simulation added to the estimation of its error provided by the Kalman filter. For example, the estimated value of the degrees of freedom of the vehicle model is as follows:

$$\underbrace{\hat{\mathbf{z}}}_{Estimation} = \underbrace{\mathbf{z}}_{MB\ Prediction} + \underbrace{\Delta\hat{\mathbf{z}}}_{Est.\ error} \tag{5.75}$$

5.7.2. Sequential application of the measurements

The measurements are used in the Kalman filter (and EKF) to correct the states and to update the estimation of the covariance matrix of state estimation uncertainty. In general, this can be written in matrix form, as in Eqs. 3.8 to 3.11. However, in most applications (included the one at hand), the noise of the different measurements is uncorrelated, meaning that the covariance matrix of measurement noise $\mathbf{\Sigma}^S$ is diagonal. Under these circumstances, the measurements can be applied sequentially, by means of scalar expressions [46]. Several benefits arise from the application of this strategy:

- Reduced computational cost: it can be demonstrated that the number of operations grows with the cube of the number of measurements using the equations in vector form, while the relation is linear when the scalar form is employed [46].
- Improved numerical accuracy: the scalar implementation makes the quantity $\mathbf{h_x P h_x}^\top + \mathbf{\Sigma}^S$ a scalar, thus avoiding matrix inversion when the Kalman gain $\mathbf{K}$ is calculated.
- Natural management of multirate: if not all the measurements are available at every time step, they are only applied when available, but the size of the matrices does not need to be changed.

The scalar update is performed following the next equations:

$$\tilde{\mathbf{y}}_k^{[i]} = \mathbf{o}_k^{[i]} - \left[\mathbf{h}(\hat{\mathbf{x}}_k^{[i-1]})\right]^{[i]} \tag{5.76}$$

$$\mathbf{\Sigma}_k^{[i]} = \mathbf{h_x}_k^{[i]} \mathbf{P}_k^{[i]} \left[\mathbf{h_x}_k^{[i]}\right]^\top + \mathbf{\Sigma}^{S[i]}_k \tag{5.77}$$

$$\mathbf{K}_k^{[i]} = \frac{\mathbf{P}_k^{[i]} \left[\mathbf{h_x}_k^{[i]}\right]^\top}{\mathbf{\Sigma}_k^{[i]}} \tag{5.78}$$

$$\hat{\mathbf{x}}_k^{[i]} = \hat{\mathbf{x}}_k^{[i-1]} + \mathbf{K}_k^{[i]} \tilde{\mathbf{y}}_k^{[i]} \tag{5.79}$$

$$\mathbf{P}_k^{[i]} = \mathbf{P}_k^{[i-1]} - \mathbf{K}_k^{[i]} \mathbf{h_x}_k^{[i]} \mathbf{P}_k^{[i-1]} \tag{5.80}$$

where the superindex $[i]$ indicates the number of the measurement being applied, $\mathbf{\Sigma}^{S[i]}_k$ is the i^{th} diagonal element of the covariance matrix of measurement noise, $\mathbf{\Sigma}^S$, and $\mathbf{h_x}_k^{[i]}$ is the i^{th} row of the matrix $\mathbf{h_x}_k$. The process is started at every time step with $\mathbf{P}_k^{[0]} = \mathbf{P}_k^-$ and $\hat{\mathbf{x}}_k^{[0]} = \hat{\mathbf{x}}_k^-$. After all the available measurements at one time step are applied, $\mathbf{P}_k^{[i]}$ becomes $\mathbf{P}_k^+$, and $\hat{\mathbf{x}}_k^{[i]}$ becomes $\hat{\mathbf{x}}_k^+$.

The scalar application of the measurements is particularly convenient to deal with GPS measurements, which involve different issues. First, the GPS receiver has a lower sampling rate than the other sensors installed on the vehicle. Moreover, the GPS signal is not always available, mainly due to the blockage of the signal produced by trees, buildings or other obstacles. Finally, even when the GPS provides a solution, the course over ground measurement is not reliable if the speed is below a certain threshold, and hence it is discarded if the measured speed is below 0.5

m/s. With the vector form update, several cases should have been programmed, depending on the availability of the measurements, but with the scalar update, all the available measurements are sequentially applied.

5.8. Tests and results

A low speed maneuver was performed with the prototype used in this thesis. As the state observer has not been implemented yet, the state observer was run offline with the data gathered during the test. Two tests were accomplished with these data. In the first one, the measurements were applied without modifications. However, the GPS receiver employed in this work is too expensive to be employed in a commercial vehicle. Therefore, in the second test, the data gathered by the GPS were modified to emulate the properties of a more affordable device.

5.8.1. Test 1: GPS data without modifications

This test is used to demonstrate that the multibody simulation can be corrected if enough information is provided. Unfortunately, there are not redundant sensors, an hence there is not a reference solution to compare with. For this reason, the GPS measurements are used both as sensors for the filter, and also as the reference to check the accuracy of the observer. The only remarkable advantage of the filter in this example is that it increases the rate of the available measurements, since the sampling rate of the GPS is 50 Hz, while the state observer provides data at 250 Hz, which is a more adequate rate to perform any control action.

The results achieved in this test are shown in Figures 5.22 to 5.25. It can be seen that, at position level, the results from the GPS and the state observer are overlapped. At velocity level, the concordance is worst. However, the measurements from the GPS at velocity level are less reliable than those at position level, since the differential corrections are only applied to the positioning data. The multibody model without corrections is also shown to demonstrate that it presents an obvious drift if the corrections are not applied. Although the exact reason behind this drift is unknown, any small variation in the rolling resistance coefficient, or a little friction due to an aged brake caliper is enough to produce this noticeable discrepancy.

Regarding the computational cost of the method, the actual maneuver lasts 55.744 s, and it took 32.06 seconds to run the state observer in an Intel Core i5 CPU 650 at 3.20 GHz with 4 Gb of RAM. Although the method is faster than real time for the whole simulation, a more detailed study of the time employed in every time step is to be done. Moreover, an iteration limit of the multibody integrator might have to be set to ensure that every time step lasts less than the desired threshold.

5.8.2. Test 2: GPS data emulating a low-cost GPS receiver

The only GPS receiver installed on the vehicle prototype is the high precision device employed in the previous section. Therefore, to verify the behavior of the observation algorithm with other devices, the measurements provided to the observer

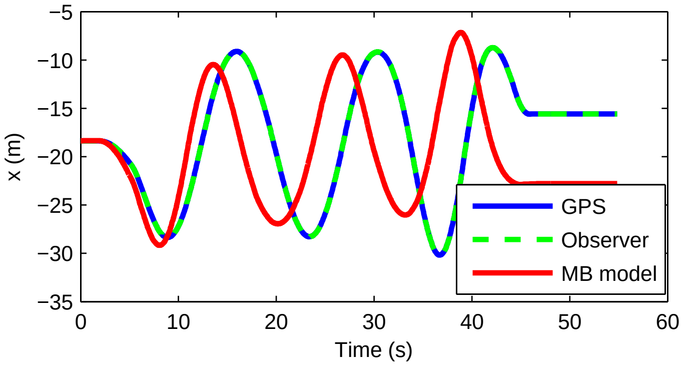


Figure 5.22: Position along x axis (east).

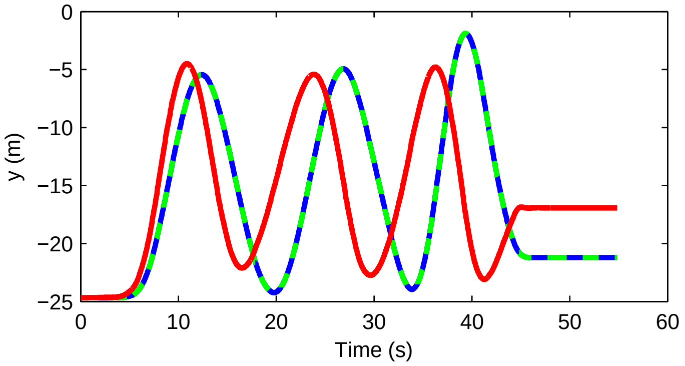


Figure 5.23: Position along y axis (north).

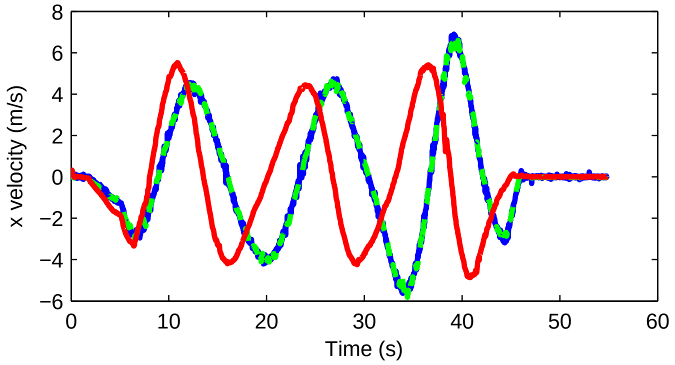


Figure 5.24: Velocity along x axis (east).

were modified to emulate data provided by a low-cost GPS receiver. According to [88], it is reasonable considering that the accuracy of a GPS receiver is under 3.5 m 95 % of the time in horizontal positioning, and 5 in vertical. Assuming that the positioning error follows a Gaussian distribution, these errors are equivalent to noise with mean 0 and standard deviation of 1.786 m in horizontal, and 2.551 m in vertical.

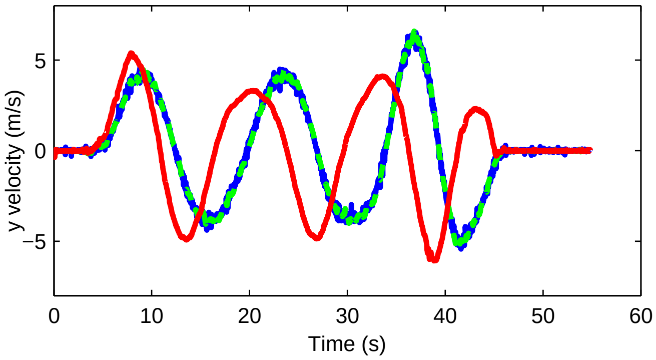


Figure 5.25: Velocity along y axis (north).

Moreover, the sampling rate of a low-cost GPS receiver rarely exceeds 10 Hz. In this example, it is assumed that the receiver provides data at 5 Hz. At velocity level, the accuracy is the same, thus the only modification is the sampling rate. The results under these circumstances are shown in Figures 5.26 to 5.29. Here the GPS with differential corrections is used as the reference to show that the proposed state observer is much more accurate than the GPS measurements provided to it, while maintaining an output data rate of 250 Hz. The root mean squared error in horizontal position in this test is 0.562 m with respect to the RTK GPS, while the measurements of the emulated GPS have a root mean squared error of 1.786 m. Regarding velocities, the results provided are not so accurate, since the observer shows some delay which can be appreciated in Figures 5.28 and 5.29.

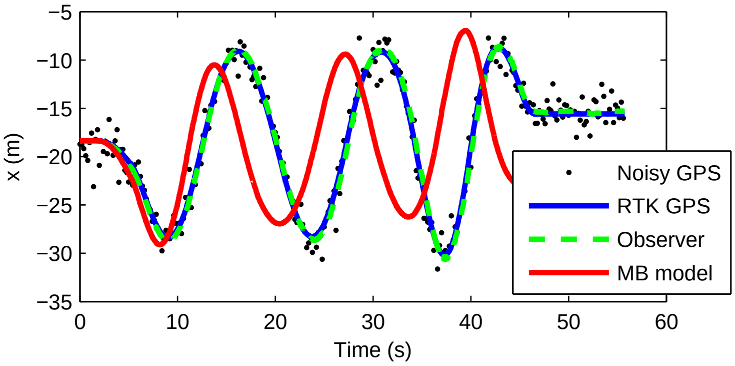


Figure 5.26: Position along x axis (east), emulating measurements from a low-cost GPS receiver.

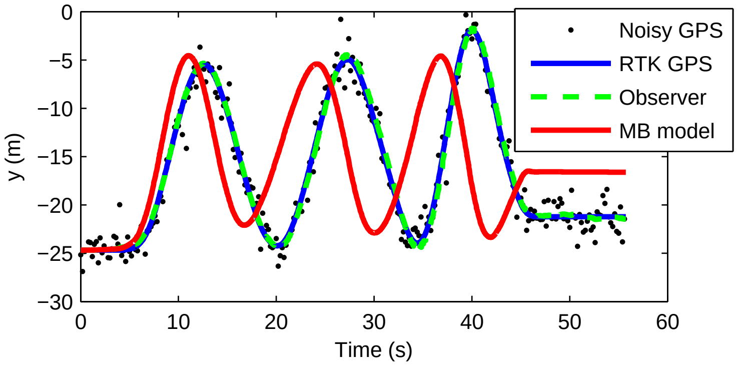


Figure 5.27: Position along y axis (north), emulating measurements from a low-cost GPS receiver.

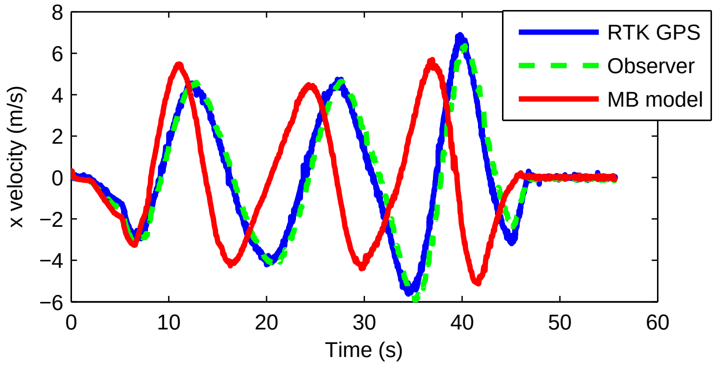


Figure 5.28: Velocity along x axis (east), emulating measurements from a low-cost GPS receiver.

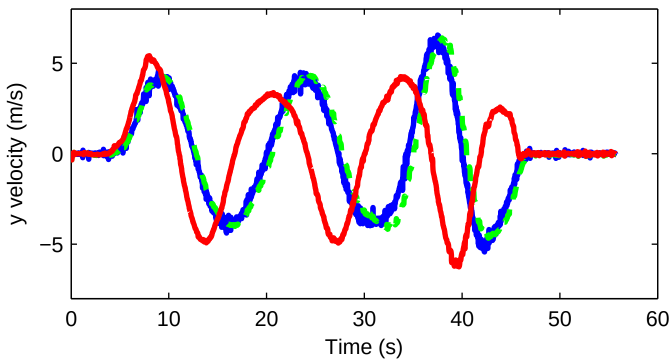


Figure 5.29: Velocity along y axis (north), emulating measurements from a low-cost GPS receiver.

Chapter 6

Conclusions and future work

6.1. Conclusions

This thesis can be divided into two main parts. In the first one, a benchmark was build to test many multibody-based state observers with two simple mechanisms, a four-bar and a five-bar linkages. This part is covered in chapter 4. The knowledge acquired during the first part was then applied to a state observer of a vehicle, treated in chapter 5.

In the first part, a MATLAB$^{®}$ code was developed, starting with the continuous-time extended Kalman filter (CEKF) method, which had been developed in previous works. Then, new methods were implemented and tested, starting with the discrete version of the CEKF, called discrete-time extended Kalman filter (DEKF), and an unscented filter employing positions and velocities as states, the unscented Kalman filter (UKF).

All these methods estimate positions and velocities of the degrees of freedom of the mechanisms, thus position and velocity problems have to be solved every time step. As this is a costly process, the concept of perfect measurements was tested. This method consists in imposing the constraints as if they were sensor measurements. However, if the constraints are nonlinear, they have to be imposed in an iterative fashion. Two methods were tested with this idea, the smoothly constrained Kalman filter (SCKF), and the discrete-time iterated extended Kalman filter with perfect measurements (DIEKFpm). Nevertheless, it was found that none of the methods are as stable as the methods in independent coordinates, and the results provided by them were worst, hence they were discarded.

Finally, an indirect method was tested. This method, called error-state extended Kalman filter (errorEKF), estimates the deviation suffered by the multibody simulation, and then it is corrected.

The accuracy achieved by the errorEKF method is at the level of the DEKF, but with a lower computational cost. In addition, other advantage of the method is that it can be implemented over exiting multibody simulations without making changes to the multibody formulation nor the integrator. The UKF also shares this advantage, and achieves a better accuracy, but its computational cost is much higher. Therefore, it cannot be used in real time in problems of a considerable size, but it is worth considering it in problems with a reduced number of degrees of freedom.

In the second part, a state observer based on the multibody model of a vehicle was implemented. The first part of this task consisted in the improvement of the multibody model of the vehicle already available, with two main objectives: improving the accuracy of the model, and reducing its computational cost. To achieve these aims, the model of the wheel was changed from natural coordinates to relative coordinates, while natural coordinates are still maintained in all the other elements. This change avoids the integration of the Cartesian components of fast rotating vectors, thus improving the conservation of the mechanical energy, and making the integration of the model faster and more robust. The brake model was also improved, characterizing its parameters by means of experiments. The rolling resistant coefficient was also experimentally determined.

After that, the multibody model was installed on board the vehicle, to demonstrate that it can be run in real time while driving the prototype. The inputs of the multibody model were taken from the on-board sensors available at the vehicle.

Finally, this model was used to implement a state observer using the errorEKF method. Although this observer has not been implemented on board the vehicle yet due to time constraints, it was run faster than real time in an offline simulation with data taken from a real test. The previous experience running the multibody model on board the vehicle suggests that the state observer should work smoothly.

6.2. Future work

The research done in this thesis leaves some uncovered topics which can become future research lines. The first one is the implementation of the state observer on board the vehicle to check that it can still be run without stability issues. Moreover, a study of the time employed in each time step should be performed to verify that all of them are under real time. It is possible that an iteration limit is needed to guarantee this point.

The verification of unmeasured magnitudes such as the sideslip angle would be also desirable, both in simulation and with the actual vehicle, although doing it with the real vehicle would require the installation of new sensors. A comparison with the estimation provided by simpler observers reported in the literature would be also welcome.

As a future extension of the errorEKF method, the estimation of the input forces could be added. This can be done by adding the acceleration error to the states and then performing the inverse dynamics of these errors to estimate the input forces which have to be added to compensate the current acceleration deviation. This method should be more accurate for slow varying forces than the random walk usually employed in the augmented Kalman filter.

Finally, all the noises considered in this thesis are white and Gaussian. However, not all the noises fall within this classification. In particular, plant noise is mainly produced by wrong force models, which tend to produce a biased estimation. Characterizing these noises with shaping filters could help to improve the accuracy of the estimations.

Bibliography

[1] Javier Cuadrado, Daniel Dopico, Antonio Barreiro, and Emma Delgado. Real-time state observers based on multibody models and the extended Kalman filter. *Journal of Mechanical Science and Technology*, 23(4):894–900, 2009.

[2] Javier Cuadrado, Daniel Dopico, José A. Pérez, and Roland Pastorino. Influence of the sensored magnitude in the performance of observers based on multibody models and the extended Kalman filter. In *ECCOMAS Thematic Conference on Multibody Dynamics*, pages 126–127, Warsaw, Poland, 2009.

[3] Javier Cuadrado, Daniel Dopico, José A. Pérez, and Roland Pastorino. Automotive observers based on multibody models and the extended Kalman filter. *Multibody System Dynamics*, 27(1):3–19, 2012.

[4] Roland Pastorino. *Experimental validation of a multibody model for a vehicle prototype and its application to automotive state observers*. PhD thesis, University of A Coruña, 2012.

[5] Harold W. Sorenson. Least-squares estimation: from Gauss to Kalman. *IEEE Spectrum*, 7(7):63–68, 1970.

[6] Aaron Radke and Zhiqiang Gao. A survey of state and disturbance observers for practitioners. *2006 American Control Conference, Vols 1-12*, 1-12:5183–5188, 2006.

[7] Rudolf E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.

[8] Simon J. Julier, Jeffrey K. Uhlmann, and Hugh F. Durrant-Whyte. A new approach for filtering nonlinear systems. In *Proceedings of the American Control Conference*, volume 3, pages 1628–1632, 1995.

[9] Mohinder Grewal and Angus Andrews. Applications of Kalman filtering in aerospace 1960 to the present [Historical perspectives]. *IEEE Control Systems Magazine*, 30(3):69–78, 2010.

[10] WHO. Global status report on road safety 2015, 2015.

[11] Laura R. Ray. Nonlinear state and tire force estimation for advanced vehicle control. *IEEE Transactions on Control Systems Technology*, 3(1):117–124, 1995.

[12] Laura R. Ray. Nonlinear tire force estimation and road friction identification: simulation and experiments. *Automatica*, 33(10):1819–1833, 1997.

[13] Paul J.TH. Venhovens and Karl Naab. Vehicle dynamics estimation using Kalman filters. *Vehicle System Dynamics*, 32(2-3):171–184, 1999.

[14] Matthew C. Best, Timothy J. Gordon, and Philip J. Dixon. An extended adaptive Kalman filter for real-time state estimation of vehicle handling dynamics. *Vehicle System Dynamics*, 34(1):57–75, 2000.

[15] David M. Bevly, J. Christian Gerdes, Christopher Wilson, and Gengsheng Zhang. The use of GPS based velocity measurements for improved vehicle state estimation. In *Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No.00CH36334)*, volume 4, pages 2538–2542. IEEE, 2000.

[16] Jihan Ryu, Eric J. Rossetter, and J. Christian Gerdes. Vehicle sideslip and roll parameter estimation using GPS. In *Proceedings of the International Symposium on Advanced Vehicle Control (AVEC), Hiroshima, Japan*, Hiroshima, 2002.

[17] Joanny Stephant, Ali Charara, and Dominique Meizel. Virtual sensor: Application to vehicle sideslip angle and transversal forces. *IEEE Transactions on Industrial Electronics*, 51(2):278–289, 2004.

[18] Christopher D. Gadda, Paul Yih, and J. Christian Gerdes. Incorporating a model of vehicle dynamics in a diagnostic system for steer-by-wire vehicles. In *Proceedings of the 7th Int. Symposium on Advanced Vehicle Control (AVEC)*, pages 779–784, 2004.

[19] Thomas A. Wenzel, Keith J. Burnham, Mike V. Blundell, and Russell A. Williams. Dual extended Kalman filter for vehicle state and parameter estimation. *Vehicle System Dynamics*, 44(2):153–171, 2006.

[20] Thomas A. Wenzel, Keith J. Burnham, Mike V. Blundell, and Russell A. Williams. Kalman filter as a virtual sensor: applied to automotive stability systems. *Transactions of the Institute of Measurement and Control*, 29(2):95–115, 2007.

[21] Guillaume Baffet, Ali Charara, Daniel Lechner, and Damien Thomas. Experimental evaluation of observers for tire–road forces, sideslip angle and wheel cornering stiffness. *Vehicle System Dynamics*, 46(6):501–520, 2008.

[22] Guillaume Baffet, Ali Charara, and Daniel Lechner. Estimation of vehicle sideslip, tire force and wheel cornering stiffness. *Control Engineering Practice*, 17(11):1255–1264, 2009.

[23] King Tin Leung, James F. Whidborne, David Purdy, and Phil Barber. Road vehicle state estimation using low-cost GPS/INS. *Mechanical Systems and Signal Processing*, 25(6):1988–2004, 2011.

[24] Moustapha Doumiati, Ali Charara, Alessandro Victorino, and Daniel Lechner. *Vehicle Dynamics Estimation using Kalman Filtering: Experimental Validation.* Wiley-ISTE, 2012.

[25] Xu Li, Xiang Song, and Chingyao Chan. Reliable vehicle sideslip angle fusion estimation using low-cost sensors. *Measurement*, 51(1):241–258, 2014.

[26] Marco Gadola, Daniel Chindamo, Marco C. Romano, and Fabrizio Padula. Development and validation of a Kalman filter-based model for vehicle slip angle estimation. *Vehicle System Dynamics*, 52(1):68–84, 2014.

[27] Homer Rahnejat. Multi-body dynamics: historical evolution and application. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 214(1):149–173, 2000.

[28] Nikki V. Orlandea, Donald A. Calahan, and Milton A. Chace. A sparsity-oriented approach to the dynamic analysis and design of mechanical systems—Part 1. *Journal of Engineering for Industry*, 99(3):773–779, 1977.

[29] Parviz E. Nikravesh. *Computer-aided analysis of mechanical systems.* Prentice-Hall, Engelwood Cliffs, NJ, USA, 1988.

[30] Paulo Flores, Jorge Ambrósio, José C.P. Claro, and Hamid M. Lankarani. *Kinematics and dynamics of multibody systems with imperfect joints*, volume 34 of *Lecture Notes in Applied and Computational Mechanics.* Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[31] Javier García de Jalón and Eduardo Bayo. *Kinematic and dynamic simulation of multibody systems: the real time challenge.* Springer-Verlag, 1994.

[32] Taichi Shiiba and Yoshihiro Suda. Evaluation of driver's behavior with multibody-based driving simulator. *Multibody System Dynamics*, 17(2-3):195–208, 2007.

[33] Michael W. Sayers. Vehicle models for RTS applications. *Vehicle System Dynamics*, 32(4-5):421–438, 1999.

[34] Roland Pastorino, Francesco Cosco, Frank Naets, Wim Desmet, and Javier Cuadrado. Hard real-time multibody simulations using ARM-based embedded systems. *Multibody System Dynamics*, 37(1):127–143, 2016.

[35] Roland Pastorino, Dario Richiedei, Javier Cuadrado, and Alberto Trevisani. State estimation using multibody models and unscented Kalman filters. In *The 2nd Joint International Conference on Multibody System Dynamics*, pages 245–246, Stuttgart, Germany, 2012.

[36] Roland Pastorino, Dario Richiedei, Javier Cuadrado, and Alberto Trevisani. State estimation using multibody models and non-linear Kalman filters. *International Journal of Non-Linear Mechanics*, 53:83–90, 2013.

[37] Frank Naets, Roland Pastorino, Javier Cuadrado, and Wim Desmet. Online state and input force estimation for multibody models employing extended Kalman filtering. *Multibody System Dynamics*, 32(3):317–336, 2014.

[38] Frank Naets, Javier Cuadrado, and Wim Desmet. Stable force identification in structural dynamics using Kalman filtering and dummy-measurements. *Mechanical Systems and Signal Processing*, 50-51:235–248, 2015.

[39] Ilaria Palomba, Dario Richiedei, and Alberto Trevisani. Simultaneous estimation of kinematic state and unknown input forces in rigid-link multibody systems. In *ECCOMAS Thematic Conference. Multibody Dynamics 2015*, Barcelona, 2015.

[40] Ilaria Palomba, Dario Richiedei, and Alberto Trevisani. Kinematic state estimation for rigid-link multibody systems by means of nonlinear constraint equations. *Multibody System Dynamics*, doi:10.1007/s11044-016-9515-x, 2016.

[41] José L. Torres-Moreno, José L. Blanco-Claraco, Emilio Sanjurjo, Miguel Á. Naya, and Antonio Giménez-Fernández. Towards benchmarking of state estimators for multibody dynamics. In *The 3rd Joint International Conference on Multibody System Dynamics. The 7th Asian Conference on Multibody Dynamics*, pages 261–262, Busan, Korea, 2014.

[42] José L. Torres-Moreno, José L. Blanco-Claraco, Emilio Sanjurjo, Antonio Giménez-Fernández, and Miguel Á. Naya. A testbed for benchmarking state observers in multibody dynamics. In *ECCOMAS Thematic Conference on Multibody Dynamics 2015*, Barcelona, 2015.

[43] Emilio Sanjurjo, José L. Blanco-Claraco, José L. Torres-Moreno, and Miguel Á. Naya. Testing the efficiency and accuracy of multibody-based state observers. In *ECCOMAS Thematic Conference on Multibody Dynamics*, pages 1595–1606, Barcelona, 2015.

[44] José L. Torres-Moreno, José L. Blanco-Claraco, Antonio Giménez-Fernández, Emilio Sanjurjo, and Miguel Á. Naya. Online kinematic and dynamic-state estimation for constrained multibody systems based on IMUs. *Sensors*, 16(3):333, 2016.

[45] Peter S. Maybeck. *Stochastic models, estimation, and control.* Elsevier, 1979.

[46] Mohinder Grewal and Angus Andrews. *Kalman filtering: theory and practice using MATLAB®*. Wiley, New Jersey, 2008.

[47] Dan Simon. *Optimal state estimation: Kalman, H infinity, and nonlinear approaches.* Wiley, Hoboken, 2006.

[48] Rudolf. E. Kalman and Richard S. Bucy. New results in linear filtering and prediction theory. *Journal of Basic Engineering*, 83(1):95, 1961.

[49] Arend L. Schwab and J. P. Meijaard. How to draw Euler angles and utilize Euler parameters. In *Volume 2: 30th Annual Mechanisms and Robotics Conference, Parts A and B*, pages 259–265, Philadelphia, Pennsylvania, 2006. ASME.

[50] Ivo Boniolo, Mara Tanelli, and Sergio M. Savaresi. Roll angle estimation in two-wheeled vehicles. In *2008 IEEE International Conference on Control Applications*, pages 31–36, San Antonio, Texas, 2008. IEEE.

[51] Marc Schlipsing, Jan Salmen, Benedikt Lattke, Kai Gerd Schroter, and Hermann Winner. Roll angle estimation for motorcycles: Comparing video and inertial sensor approaches. In *2012 IEEE Intelligent Vehicles Symposium*, pages 500–505. IEEE, 2012.

[52] Roberto Lot, Vittore Cossalter, and Matteo Massaro. Real-time roll angle estimation for two-wheeled vehicles. In *Proceedings of the ASME 2012 11th Biennial Conference on Engineering Systems Design and Analysis ESDA2012*, pages 687–693, Nantes, France, 2012. ASME.

[53] Edward J. Haug. *Computer aided kinematics and dynamics of mechanical systems*. Allyn & Bacon, Inc., Needham Heights, MA, USA, 1989.

[54] Brian Paul and Dusan Krajcinovic. Computer analysis of machines with planar motion: Part 1—Kinematics. *Journal of Applied Mechanics*, 37(3):697–702, 1970.

[55] Pradip N. Sheth and John J. Uicker. IMP (Integrated Mechanisms Program), a computer-aided design analysis system for mechanisms and linkage. *Journal of Engineering for Industry*, 94(2):454–464, 1972.

[56] Donald A. Smith, Milton A. Chace, and A. C. Rubens. The automatic generation of a mathematical model for machinery systems. *Journal of Engineering for Industry*, 95(2):629–635, 1973.

[57] Daniel Dopico, Francisco González, Javier Cuadrado, and József Kövecses. Determination of holonomic and nonholonomic constraint reactions in an index-3 augmented Lagrangian formulation with velocity and acceleration projections. *Journal of Computational and Nonlinear Dynamics*, 9(4):041006, 2014.

[58] Miguel Á. Naya, Daniel Dopico, José Á. Pérez, and Javier Cuadrado. Real-time multi-body formulation for virtual-reality-based design and evaluation of automobile controllers. *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics*, 221(2):261–276, 2007.

[59] Daniel Dopico, Alberto Luaces, Manuel González, and Javier Cuadrado. Dealing with multiple contacts in a human-in-the-loop application. *Multibody System Dynamics*, 25(2):167–183, 2010.

[60] Roland Pastorino, Emilio Sanjurjo, Alberto Luaces, Miguel Á. Naya, Wim Desmet, and Javier Cuadrado. Validation of a real-time multibody model for an X-by-wire vehicle prototype through field testing. *Journal of Computational and Nonlinear Dynamics*, 10(3):031006, 2015.

[61] Javier Cuadrado, Daniel Dopico, and Miguel Á. Naya. Penalty, semi-recursive and hybrid methods for MBS real-time dynamics in the context of structural integrators. *Multibody System Dynamics*, 12(2):117–132, 2004.

[62] Eric A. Wan and Rudolph Van Der Merwe. The unscented Kalman filter for nonlinear estimation. In *Proceedings of the IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium*, pages 153–158, 2000.

[63] Rudolph Van Der Merwe, Eric A Wan, and Simon I Julier. Nonlinear estimation and sensor-fusion : applications to integrated navigation. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Providence, Rhode Island, 2004.

[64] Geeter, Jan De and Van Brussel, Hendrik and Schutter, Joris De and Decréton, Marc. A smoothly constrained Kalman filter. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(10):1171–1177, 1997.

[65] Dan Simon and Tien Li Chia. Kalman filtering with state equality constraints. *IEEE Transactions on Aerospace and Electronic Systems*, 38(1):128–136, 2002.

[66] Stergios I. Roumeliotis, Gaurav S. Sukhatme, and George A. Bekey. Circumventing dynamic modeling: evaluation of the error-state Kalman filter applied to mobile robot localization. In *Proceedings 1999 IEEE International Conference on Robotics and Automation*, volume 2, pages 1656–1663. IEEE, 1999.

[67] Daniel Dopico, Yitao Zhu, Adrian Sandu, and Corina Sandu. Direct and adjoint sensitivity analysis of ordinary differential equation multibody formulations. *Journal of Computational and Nonlinear Dynamics*, 10(1):011012, 2014.

[68] Charles F. Van Loan. Computing integrals involving the matrix exponential. *IEEE Transactions on Automatic Control*, 23(3):395–404, 1978.

[69] Zhe Chen. Local observability and its application to multiple measurement estimation. *IEEE Transactions on Industrial Electronics*, 38(6):491–496, 1991.

[70] Tine Lefebvre, Herman Bruyninckx, and Joris De Schuller. Comment on “A new method for the nonlinear transformation of means and covariances in filters and estimators” [with authors’ reply]. *IEEE Transactions on Automatic Control*, 47(8):1406–1409, 2002.

[71] Guoquan P. Huang, Anastasios I. Mourikis, and Stergios I. Roumeliotis. A quadratic-complexity observability-constrained unscented Kalman filter for SLAM. *IEEE Transactions on Robotics*, 29(5):1226–1243, oct 2013.

[72] Roland Pastorino, Miguel Á. Naya, José A. Pérez, and Javier Cuadrado. X-by-wire vehicle prototype: a steer-by-wire system with geared PM coreless motors. In *7th EUROMECH Solid Mechanics Conference*, pages 737–738, Lisbon, 2009.

[73] Roland Pastorino, Daniel Dopico, Emilio Sanjurjo, and Miguel Á. Naya. Validation of a multibody model for an x-by-wire vehicle prototype through field

testing. In *ECCOMAS Thematic Conference. Multibody Dynamics 2011*, pages 144–145, Brussels, 2011.

[74] Emilio Sanjurjo, Roland Pastorino, Pasquale Gallo, and Miguel Á. Naya. Implementation Issues of an on Board Real-Time Multibody Model. In *The 3rd Joint International Conference on Multibody System Dynamics. The 7th Asian Conference on Multibody Dynamics.*, pages 249–250, Busan, Korea, 2014.

[75] Defense Mapping Agency. DMA technical manual 8358.1. Datums, ellipsoids, grids, and grid reference systems. Technical report, 1990.

[76] EUREF. http://www.euref.eu/. Accessed: 2016-05-16.

[77] Elliott D Kaplan and Christopher J Hegarty. *Understanding GPS: principles and applications.* Artech House, Norwood, second edition, 2006.

[78] NMEA. http://www.nmea.org/. Accessed: 2016-05-19.

[79] MBSLIM. http://lim.ii.udc.es/MBSLIM/. Accessed: 2016-05-20.

[80] E. Kuiper and Jan J. M. Van Oosten. The PAC2002 advanced handling tire model. *Vehicle System Dynamics*, 45(sup1):153–167, 2007.

[81] Egbert Bakker, Lars Nyborg, and Hans B. Pacejka. Tyre modelling for use in vehicle dynamics studies. Technical report, 1987.

[82] Egbert Bakker, Hans B. Pacejka, and Lars Lidner. A new tire model with an application in vehicle dynamics studies. Technical report, 1989.

[83] Hans B. Pacejka and Igo J. M. Besselink. Magic formula tyre model with transient properties. *Vehicle System Dynamics*, 27(S1):234–249, 1997.

[84] Hans B. Pacejka. *Tire and Vehicle Dynamics.* 2012.

[85] Wolfgang Hirschberg, Georg Rill, and Heinz Weinfurter. Tire model TMeasy. *Vehicle System Dynamics*, 45(S1):101–119, 2007.

[86] Thomas D. Gillespie. *Fundamentals of vehicle dynamics.* Society of Automotive Engineers, 1992.

[87] Paul D. Groves. *Principles of GNSS, inertial, and multisensor integrated navigation systems.* Artech House, London, 2008.

[88] Global positioning system standard sositioning service serformance standard. Technical report, Official U.S. Government information about the Global Positioning System (GPS) and related topics, 2008.

Appendices

Appendix A

Former multibody model

The chapter 3 of [4], where the former multibody model of the vehicle is described, is reproduced hereafter for the convenience of the reader.

3.1 Vehicle modeling

3.1.1 Multibody formulation and integrator

As explained in section 1.2, MB vehicle models can be built either employing commercial MB softwares or self-developed MB codes. This work deals with the second option. The XBW vehicle prototype presented in section 2.2 has been modeled using fully Cartesian dependent coordinates, also called natural coordinates (García de Jalón and Bayo, 1994; García de Jalón, 2007). For three dimensional MB systems, these coordinates describe the positions of each element by means of the Cartesian coordinates of *basic points* distributed throughout all the elements and the Cartesian components of several unit vectors. Each element of the system should have a sufficient number of points and vectors linked to it so that their motion completely defines that of the element. The chosen MB formulation is an *index 3 augmented Lagrangian (I3AL) formulation with mass–damping–stiffness–orthogonal projections in velocities and accelerations*. It is explained in detail below.

The constraints that relate the dependent coordinates can first be grouped as shown in eq. (3.1). The Lagrange's equations for a constrained mechanical system are presented in eq. (3.2). Equations (3.1) and (3.2) constitutes a system of DAEs. Its solution yields the values of n_d dependent coordinates as well as the m Lagrange multipliers. Instead of solving the system using this approach, it is possible to introduce some penalty terms following the alternative penalty formulation approach (Bayo et al., 1988). The resulting equations, shown in eq. (3.3) yield the *augmented Lagrangian (AL) formulation* (García de Jalón and Bayo, 1994), where the penalty terms are zero if the constraints are satisfied. In this method, in order to avoid using explicitly eq. (3.1), the Lagrange multipliers are calculated iteratively, as shown in eq. (3.4). This last equation represents the progressive introduction of forces that help to fulfill better the constraints of eq. (3.1). Finally, the iterative process of eq. (3.4) can be introduced in eq. (3.3), leading to eq. (3.5) that is used to iterate until $\|\mathbf{q}_{\mathrm{i}+1} - \mathbf{q}_{\mathrm{i}}\| \leq \epsilon$, where ϵ is user–defined.

$$\boldsymbol{\Phi}(\mathbf{q}, \mathrm{t}) = \mathbf{0} \tag{3.1}$$

$$\mathbf{M}\ddot{\mathbf{q}} + \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\lambda} = \mathbf{Q} \tag{3.2}$$

$$\mathbf{M}\ddot{\mathbf{q}} + \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\alpha}(\ddot{\boldsymbol{\Phi}} + 2\boldsymbol{\omega}\boldsymbol{\zeta}\dot{\boldsymbol{\Phi}} + \boldsymbol{\omega}^2\boldsymbol{\Phi}) + \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\lambda}^* = \mathbf{Q} \tag{3.3}$$

$$\boldsymbol{\lambda}_{\mathrm{i}+1}^* = \boldsymbol{\lambda}_{\mathrm{i}}^* + \boldsymbol{\alpha}(\ddot{\boldsymbol{\Phi}} + 2\boldsymbol{\omega}\boldsymbol{\zeta}\dot{\boldsymbol{\Phi}} + \boldsymbol{\omega}^2\boldsymbol{\Phi}) \quad \text{with } \boldsymbol{\lambda}_0^* = \mathbf{0} \tag{3.4}$$

$$(\mathbf{M} + \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\alpha}\boldsymbol{\Phi}_{\mathbf{q}})\ddot{\mathbf{q}}_{\mathrm{i}+1} = \mathbf{M}\ddot{\mathbf{q}}_{\mathrm{i}} - \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\alpha}(\dot{\boldsymbol{\Phi}}_{\mathbf{q}}\dot{\mathbf{q}} + \dot{\boldsymbol{\Phi}}_{\mathrm{t}} + 2\boldsymbol{\omega}\boldsymbol{\zeta}\dot{\boldsymbol{\Phi}} + \boldsymbol{\omega}^2\boldsymbol{\Phi}) \tag{3.5}$$

where i is the index for the iterative process (i=0,1,2...), $\boldsymbol{\Phi}$ are the constraints, $\mathbf{q}$ is the vector of dependent coordinates, $\mathbf{M}$ is the mass matrix, $\boldsymbol{\Phi}_{\mathbf{q}}$ is the Jacobian matrix of the constraint equations, $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}^*$ are the Lagrange multipliers, $\mathbf{Q}$ contains the external forces, the velocity–dependent inertia forces and those obtained from a potential, $\boldsymbol{\Phi}_t$ is the partial derivative of the constraints with respect to time and $\boldsymbol{\alpha}$, $\boldsymbol{\zeta}$ and $\boldsymbol{\omega}$ contain the penalty factors (usually $> 10^7$, its dimension depends on the type of constraints), the dimensionless damping ratios (usually $\simeq 1$) and the natural frequencies (usually $\simeq 10$ rad/s) for each constraint.

The AL formulation yields a solution set of $\mathbf{q}^*$, $\dot{\mathbf{q}}^*$ and $\ddot{\mathbf{q}}^*$ that enforces the penalty system $(\ddot{\boldsymbol{\Phi}} + 2\boldsymbol{\omega}\boldsymbol{\zeta}\dot{\boldsymbol{\Phi}} + \boldsymbol{\omega}^2\boldsymbol{\Phi})$ to be exactly equal to zero (within machine precision) but not each individual constraint. In order to achieve full constraint satisfaction, constraint cleaning through mass–orthogonal projections has been proposed by Bayo and Ledesma

(1996). The main idea of this approach is to force the set of DAE to meet the underlying ordinary differential equation (ODE) by taking the solution to the constraint manifold where $\mathbf{\Phi}$, $\dot{\mathbf{\Phi}}$ and $\ddot{\mathbf{\Phi}}$ are all equal to zero. A modification of the projection method for an index 3 formulation, in which the positions $\mathbf{q}$ are the primary variables in the integration process, has been proposed by Cuadrado et al. (2000). It consists in mass–damping–stiffness–orthogonal projections in velocities and accelerations that improve the computational efficiency of the method. The integration process yields a set of velocities $\dot{\mathbf{q}}^*$ that does not completely satisfy $\dot{\mathbf{\Phi}} = \mathbf{0}$. This solution is therefore projected to the velocity constraint manifold to obtain a set of velocities $\dot{\mathbf{q}}$ that satisfy $\dot{\mathbf{\Phi}} = 0$. In a similar way, for the accelerations, the solution $\ddot{\mathbf{q}}^*$ is projected to the acceleration constraint manifold to obtain a set of accelerations $\ddot{\mathbf{q}}$ that satisfy $\ddot{\mathbf{\Phi}} = 0$. The mass–damping–stiffness–orthogonal projections in velocities and accelerations are obtained through the constrained minimization problems shown in eq. (3.6) and eq. (3.7) respectively. Each minimization problem can be solved for instance using the AL method.

$$\min_{\dot{\mathbf{q}}} \mathrm{V} = \frac{1}{2}(\dot{\mathbf{q}} - \dot{\mathbf{q}}^*)\mathrm{W}(\dot{\mathbf{q}} - \dot{\mathbf{q}}^*) \quad \text{subject to } \dot{\mathbf{\Phi}}(\mathbf{q}, \dot{\mathbf{q}}, \mathrm{t}) = \mathbf{0} \tag{3.6}$$

$$\min_{\ddot{\mathbf{q}}} \mathrm{V} = \frac{1}{2}(\ddot{\mathbf{q}} - \ddot{\mathbf{q}}^*)\mathrm{W}(\ddot{\mathbf{q}} - \ddot{\mathbf{q}}^*) \quad \text{subject to } \ddot{\mathbf{\Phi}}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathrm{t}) = \mathbf{0} \tag{3.7}$$

$$\left[\mathrm{W} + \frac{\Delta \mathrm{t}^2}{4}\mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\alpha}\mathbf{\Phi}_{\mathbf{q}}\right]\dot{\mathbf{q}}_{\mathrm{i}} = \mathrm{W}\dot{\mathbf{q}}_{\mathrm{i}}^* - \frac{\Delta \mathrm{t}^2}{4}\mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\alpha}\mathbf{\Phi}_{\mathrm{t}} \tag{3.8}$$

$$\left[\mathrm{W} + \frac{\Delta \mathrm{t}^2}{4}\mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\alpha}\mathbf{\Phi}_{\mathbf{q}}\right]\ddot{\mathbf{q}}_{\mathrm{i}} = \mathrm{W}\ddot{\mathbf{q}}_{\mathrm{i}}^* - \frac{\Delta \mathrm{t}^2}{4}\mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\alpha}(\dot{\mathbf{\Phi}}_{\mathbf{q}}\dot{\mathbf{q}} + \dot{\mathbf{\Phi}}_{\mathrm{t}}) \tag{3.9}$$

where $\mathrm{W} = \left(\mathbf{M} + \frac{\Delta \mathrm{t}}{2}\mathbf{C} + \frac{\Delta \mathrm{t}^2}{4}\mathbf{K}\right)$. As the projections in velocities and accelerations enforce the constraints $\dot{\mathbf{\Phi}}$ and $\ddot{\mathbf{\Phi}}$ to be equal to zero (within machine precision), the equations of motion (eq. (3.3)) and the iterative process of the Lagrange multipliers (eq. (3.4)) can be simplified as can be seen in eq. (3.10) and eq. (3.11).

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\alpha}^*\mathbf{\Phi} + \mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\lambda}^* = \mathbf{Q} \tag{3.10}$$

$$\boldsymbol{\lambda}_{\mathrm{i}+1}^* = \boldsymbol{\lambda}_{\mathrm{i}}^* + \boldsymbol{\alpha}^*\mathbf{\Phi}_{\mathrm{i}+1} \quad \text{with } \boldsymbol{\lambda}_0^* = \boldsymbol{\lambda}_{\mathrm{k}}^* \tag{3.11}$$

where i is the index for the iterative process ($\mathrm{i} = 0, 1, 2\ldots$), $\boldsymbol{\alpha}^*$ are the penalty factors (they do not have the same value than $\boldsymbol{\alpha}$ previously mentioned) and $\boldsymbol{\lambda}_k^*$ are the Lagrange multipliers of the previous time step. As integration scheme, the implicit single-step trapezoidal rule with fixed time step has been employed. The corresponding difference equations for velocities and accelerations are presented in eqs. (3.12) and (3.13).

$$\dot{\mathbf{q}}_{\mathrm{k}+1} = \frac{2}{\Delta \mathrm{t}}\mathbf{q}_{\mathrm{k}+1} + \hat{\dot{\mathbf{q}}}_{\mathrm{k}} \quad \text{with } \hat{\dot{\mathbf{q}}}_{\mathrm{k}} = -\left(\frac{2}{\Delta \mathrm{t}}\mathbf{q}_{\mathrm{k}} + \dot{\mathbf{q}}_{\mathrm{k}}\right) \tag{3.12}$$

$$\ddot{\mathbf{q}}_{\mathrm{k}+1} = \frac{4}{\Delta \mathrm{t}^2}\mathbf{q}_{\mathrm{k}+1} + \hat{\ddot{\mathbf{q}}}_{\mathrm{k}} \quad \text{with } \hat{\ddot{\mathbf{q}}}_{\mathrm{k}} = -\left(\frac{4}{\Delta \mathrm{t}^2}\mathbf{q}_{\mathrm{k}} + \frac{4}{\Delta \mathrm{t}}\dot{\mathbf{q}}_{\mathrm{k}} + \ddot{\mathbf{q}}_{\mathrm{k}}\right) \tag{3.13}$$

The equations of the integrator can be introduced into eq. (3.10) to establish the dynamical equilibrium at time step (k+1). They are shown in eq. (3.14) after having been scaled by a factor of $\Delta \mathrm{t}/4$ for numerical reasons.

$$\begin{aligned} \mathbf{g}(\mathbf{q}_{\mathrm{k}+1}) &\equiv \mathbf{M}\mathbf{q}_{\mathrm{k}+1} + \frac{\Delta \mathrm{t}^2}{4}\mathbf{\Phi}_{\mathbf{q}_{\mathrm{k}+1}}^{T}(\boldsymbol{\alpha}\mathbf{\Phi}_{\mathrm{k}+1} + \boldsymbol{\lambda}_{\mathrm{k}+1}) - \frac{\Delta \mathrm{t}^2}{4}\mathbf{Q}_{\mathrm{k}+1} + \frac{\Delta \mathrm{t}^2}{4}\mathbf{M}\hat{\ddot{\mathbf{q}}}_{\mathrm{k}} = 0 \\ &\equiv \frac{\Delta \mathrm{t}^2}{4}(\mathbf{M}\ddot{\mathbf{q}} + \mathbf{\Phi}_{\mathbf{q}}^{T}\boldsymbol{\alpha}\mathbf{\Phi} + \mathbf{\Phi}^{T}\boldsymbol{\lambda}^* - \mathbf{Q}) \end{aligned} \tag{3.14}$$

The solution of this nonlinear system can be obtain using for instance the Newton–Raphson method. This method is based on a linearization of $\mathbf{g}$ that consists in replacing the function by the first two terms of its expansion in Taylor series around a certain approximation $\mathbf{q}_{\mathrm{i}}$ to the desired solution, as described in eq. (3.15). To accurately approximate vector $\mathbf{q}_{\mathrm{k+1}}$, an initial approximation of it (i.e. $\mathbf{q}_0$) is calculated using an explicit integrator and then the iterative formula shown eq. (3.15) is used until the approximation error becomes insignificant, as demonstrated eq. (3.16). The tangent matrix, shown in eq. (3.17), has been approximated by removing the negligible terms $\mathbf{\Phi_{qq}}$.

$$\mathbf{g}(\mathbf{q}_{\mathrm{i+1}}) \simeq \mathbf{g}(\mathbf{q}_{\mathrm{i}}) + \frac{\partial \mathbf{g}(\mathbf{q}_{\mathrm{i}})}{\partial \mathbf{q}_{\mathrm{i}}}(\mathbf{q}_{\mathrm{i+1}} - \mathbf{q}_{\mathrm{i}}) = \mathbf{0} \tag{3.15}$$

$$\mathbf{q}_{\mathrm{k+1}} = \mathbf{q}_{\mathrm{i+1}} \quad \text{when } (\mathbf{q}_{\mathrm{i+1}} - \mathbf{q}_{\mathrm{i}}) \simeq 0 \tag{3.16}$$

$$\frac{\partial \mathbf{g}(\mathrm{t}, \mathbf{q})}{\partial \mathbf{q}} \simeq \mathrm{W} + \frac{\Delta \mathrm{t}^2}{4} \mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}} \boldsymbol{\alpha} \mathbf{\Phi}_{\mathbf{q}} \tag{3.17}$$

After convergence has been achieved in the Newton–Raphson method, the resulting velocities $\dot{\mathbf{q}}^*$ and the accelerations $\ddot{\mathbf{q}}^*$ are projected using eqs. (3.8) and (3.9) to get their cleaned counterparts $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$.

3.1.2 Details of the multibody model

The vehicle prototype has been modeled with natural coordinates plus some relative coordinates (angles and distances added for convenience), which are usually referred as mixed coordinates. An integration time step of 5 ms has been used. All the 18 bodies of the prototype have been modeled as rigid bodies. Each element or body of the system should have a sufficient number of points and vectors rigidly attached to it, so that their motion completely defines that of the element. All the points employed in the modeling of the vehicle prototype plus some vectors are presented in fig. 3.1. The rest of vectors and relative coordinates are presented in the following subsections for every body. After having defined the set of natural coordinates as well as the set of relative coordinates, constraint equations are necessary to define the relations between points and vectors. The first constraint equations that have been employed are those of rigid bodies: unit vector – eq. (3.18), constant distance – eq. (3.19) and constant angle – eq. (3.20).

$$\mathbf{u}_n \cdot \mathbf{u}_n - 1 = 0 \tag{3.18}$$

$$\mathbf{r}_{i,j} \cdot \mathbf{r}_{i,j} - \mathrm{L}_{i,j}^2 = 0 \tag{3.19}$$

$$\mathbf{r}_{i,j} \cdot \mathbf{u}_n - \mathrm{L}_{i,j}^2 \; cos\phi = 0 \tag{3.20}$$

where i and j refer to the basic point numbers and n to the unit vector numbers. Some other constraints have been defined as linear combination of previously defined vectors. For instance the unit vector $\mathbf{u}_n$ can be defined as a linear combination of the unit vector $\mathbf{u}_m$ and the segment $(i - j)$ as shown in eq. (3.21).

$$\mathbf{u}_n - \alpha \mathbf{r}_{i,j} - \beta \cdot \mathbf{u}_m = 0 \tag{3.21}$$

where α_1 and α_2 are constant scalar coefficients. Joint constraints have also been used as the one to maintain two vectors aligned (for example $\mathbf{u}_n$ and $\mathbf{r}_{i,j}$) eq. (3.22).

$$\mathbf{u}_n \wedge \mathbf{r}_{i,j} = 0 \tag{3.22}$$

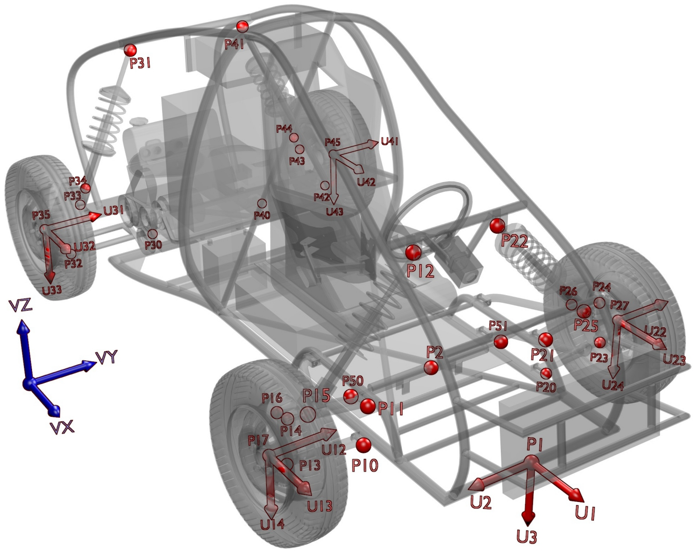


Figure 3.1: All the points and some vectors of the modeling

Varying distance constraint equations have been introduced in relation with the use of mixed coordinates eq. (3.23).

$$\mathbf{r}_{i,j} \cdot \mathbf{r}_{i,j} - \mathrm{s}^2 = 0 \tag{3.23}$$

Finally angle definitions have also been necessary. The equation corresponding to the angle between segments $(i-k)$ and $(k-j)$ when the angle, whose direction is defined by the unit vector $\mathbf{u}$, is not close to 0 nor 180 is shown in eq. (3.24). Otherwise eq. (3.25) is taken.

$$\mathbf{r}_{k,i} \cdot \mathbf{r}_{k,j} - (\mathbf{r}_{k,i} \cdot \mathbf{u})(\mathbf{r}_{k,j} \cdot \mathbf{u}) - \mathrm{L}_{i',i}\mathrm{L}_{j',j} cos\phi = 0 \tag{3.24}$$

$$\mathbf{r}_{k,i} \wedge \mathbf{r}_{k,j} - (\mathbf{r}_{k,i} \cdot \mathbf{u})\mathbf{u} \wedge \mathbf{r}_{k,j} - (\mathbf{r}_{k,j} \cdot \mathbf{u})\mathbf{r}_{k,i} \wedge \mathbf{u} - \mathbf{u}\,\mathrm{L}_{i',i}\mathrm{L}_{j',j} sin\phi = 0 \tag{3.25}$$

Rigid bodies

Chassis – The chassis has been modeled as one rigid body. This element is defined by point p_1 and unit vectors $\mathbf{u}_1$, $\mathbf{u}_2$ and $\mathbf{u}_3$. $\mathbf{u}_1$, $\mathbf{u}_2$ and $\mathbf{u}_3$ form the coordinate system of the element with origin in p_1. Extra basic points (p_2, p_{10}, p_{11}, p_{12}, p_{20}, p_{21}, p_{22}, p_{30}, p_{31}, p_{40}, p_{41}) have been defined in order to automatically consider the joints with the bodies connected to the chassis. As a consequence, the total number of variables for the chassis is 45. All the aforementioned points and vectors are demonstrated in fig. 3.2. The restrictions for the chassis are presented below. The rigid bodies constraints are shown in eqs. (3.26) to (3.31).

$$\mathbf{u}_1 \cdot \mathbf{u}_1 - 1 = 0 \tag{3.26}$$

$$\mathbf{u}_2 \cdot \mathbf{u}_2 - 1 = 0 \tag{3.27}$$

$$\mathbf{u}_3 \cdot \mathbf{u}_3 - 1 = 0 \tag{3.28}$$

$$\mathbf{u}_1 \cdot \mathbf{u}_2 = 0 \tag{3.29}$$

$$\mathbf{u}_1 \cdot \mathbf{u}_3 = 0 \tag{3.30}$$

$$\mathbf{u}_2 \cdot \mathbf{u}_3 = 0 \tag{3.31}$$

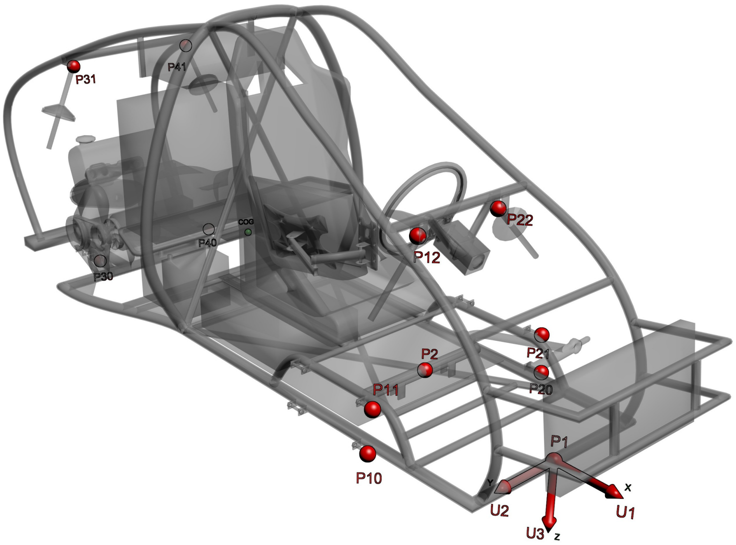


Figure 3.2: Points, vectors, COG, reference set of the chassis

The rest of constraints define the extra points as linear combination of $\mathbf{u}_1$, $\mathbf{u}_2$ and $\mathbf{u}_3$. They are presented in eqs. (3.32) to (3.42).

$$\mathbf{r}_{1,2} - \alpha_2 \cdot \mathbf{u}_1 - \beta_2 \cdot \mathbf{u}_2 - \gamma_2 \cdot \mathbf{u}_3 = 0 \tag{3.32}$$

$$\mathbf{r}_{1,10} - \alpha_{10} \cdot \mathbf{u}_1 - \beta_{10} \cdot \mathbf{u}_2 - \gamma_{10} \cdot \mathbf{u}_3 = 0 \tag{3.33}$$

$$\mathbf{r}_{1,11} - \alpha_{11} \cdot \mathbf{u}_1 - \beta_{11} \cdot \mathbf{u}_2 - \gamma_{11} \cdot \mathbf{u}_3 = 0 \tag{3.34}$$

$$\mathbf{r}_{1,12} - \alpha_{12} \cdot \mathbf{u}_1 - \beta_{12} \cdot \mathbf{u}_2 - \gamma_{12} \cdot \mathbf{u}_3 = 0 \tag{3.35}$$

$$\mathbf{r}_{1,20} - \alpha_{20} \cdot \mathbf{u}_1 - \beta_{20} \cdot \mathbf{u}_2 - \gamma_{20} \cdot \mathbf{u}_3 = 0 \tag{3.36}$$

$$\mathbf{r}_{1,21} - \alpha_{21} \cdot \mathbf{u}_1 - \beta_{21} \cdot \mathbf{u}_2 - \gamma_{21} \cdot \mathbf{u}_3 = 0 \tag{3.37}$$

$$\mathbf{r}_{1,22} - \alpha_{22} \cdot \mathbf{u}_1 - \beta_{22} \cdot \mathbf{u}_2 - \gamma_{22} \cdot \mathbf{u}_3 = 0 \tag{3.38}$$

$$\mathbf{r}_{1,30} - \alpha_{30} \cdot \mathbf{u}_1 - \beta_{30} \cdot \mathbf{u}_2 - \gamma_{30} \cdot \mathbf{u}_3 = 0 \tag{3.39}$$

$$\mathbf{r}_{1,31} - \alpha_{31} \cdot \mathbf{u}_1 - \beta_{31} \cdot \mathbf{u}_2 - \gamma_{31} \cdot \mathbf{u}_3 = 0 \tag{3.40}$$

$$\mathbf{r}_{1,40} - \alpha_{40} \cdot \mathbf{u}_1 - \beta_{40} \cdot \mathbf{u}_2 - \gamma_{40} \cdot \mathbf{u}_3 = 0 \tag{3.41}$$

$$\mathbf{r}_{1,41} - \alpha_{41} \cdot \mathbf{u}_1 - \beta_{41} \cdot \mathbf{u}_2 - \gamma_{41} \cdot \mathbf{u}_3 = 0 \tag{3.42}$$

The total number of constraints for the chassis is 39 (6 for the rigid body constraints and 33 for the linear combinations). Therefore the number of degrees of freedom (DOF) of the chassis is 6. Regarding the chassis mass properties, all the elements fixed to it should be taken into account to calculate its weight and its inertia. On the one hand, it is only necessary to sum the mass of each element for calculating the total mass of the chassis. However, if the heaviest elements were easily taken into account, the mass of smaller elements (wires, fixing elements, etc.) is more difficult to account for. The approach has been to compare the mass of the real chassis with the calculated mass and to sum the difference to the chassis frame mass. Table 3.1 summarizes these masses. On the other hand, the correct mass distribution must be used to calculate the total inertia. In this work, all the chassis elements were assembled in a

Body	Number	Unit mass (kg)	Mass (kg)
chassis frame	1	–	169.3
steering wheel assembly	1	–	1.6
monitor and support	1	–	6.2
seat	1	–	8.4
driver	1	–	75
rack of the amplifier	1	–	7
computer	1	–	10.1
engine and gearbox	1	–	180.4
exhaust system	1	–	10.1
batteries	3	11	33
DAS expansion board rack	1	–	8.1
brake system	1	–	10.3
radiator	1	–	12.1
upper part of the front dampers	2	2.7	5.4
upper part of the rear dampers	2	2	4
Total			541

Table 3.1: Chassis mass properties

COG position (m)		moments of inertia (kg m^2)			
X	-1.489	I_{xx}	+64.571	I_{xy}	-0.900
Y	+0.026	I_{yy}	+306.849	I_{xz}	-2.403
Z	-0.285	I_{zz}	+297.270	I_{yz}	-1.722

Table 3.2: COG coordinates and inertia tensor for the chassis

CAD software to calculate this inertia. Average positions for the upper parts of the front and rear dampers have been used. The position of the center of gravity (COG) and the inertia tensor of this element, expressed in the reference set shown in fig. 3.2, are summarized in table 3.2.

Front right lower wishbone arm – This element is defined by points p_{10}, p_{13} and unit vectors $\mathbf{u}_1$ and $\mathbf{u}_{10}$, as shown in fig. 3.3. $\mathbf{u}_{10}$, $\mathbf{u}_1$ and segment (10–13) form the coordinate system of the element with origin in p_{10}. 6 new variables (p_{13} and $\mathbf{u}_{10}$) are added. It is worth mentioning that as point p_{10} and vector $\mathbf{u}_1$ have been previously defined during the chassis modeling, a revolute joint between the chassis and the front right lower wishbone arm is automatically considered. The constraints for this element are rigid body constraints, eqs. (3.43) to (3.47). As 6 new variables and 5 constraints are added, a new DOF is defined. Considering the previously defined DOFs, the DOF number is at this stage of the modeling is 7. The body mass is 4.2 kg. The position of the COG and the inertia tensor of this element,

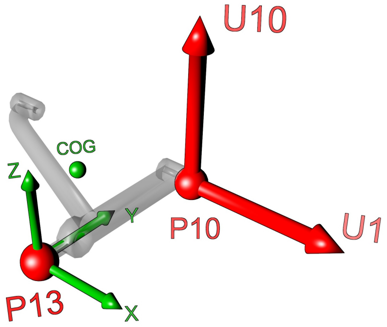


Figure 3.3: Points, vectors, COG, reference set of the front right lower wishbone arm

COG position (m)		moments of inertia (kg m^2)			
X	-0.109	I_{xx}	+0.020	I_{xy}	-0.017
Y	+0.182	I_{yy}	+0.061	I_{xz}	+0.000
Z	+0.0	I_{zz}	+0.081	I_{yz}	+0.000

Table 3.3: COG coordinates and inertia tensor for the front right lower wishbone arm

expressed in the reference set shown in fig. 3.3, are summarized in table 3.3.

$$\mathbf{u}_{10} \cdot \mathbf{u}_{10} - 1 = 0 \tag{3.43}$$

$$\mathbf{u}_{1} \cdot \mathbf{u}_{10} - \cos \phi_{1,10} = 0 \tag{3.44}$$

$$\mathbf{u}_{1} \cdot \mathbf{r}_{10,13} - \mathrm{L}_{10,13} = 0 \tag{3.45}$$

$$\mathbf{u}_{10} \cdot \mathbf{r}_{10,13} = 0 \tag{3.46}$$

$$\mathbf{r}_{10,13} \cdot \mathbf{r}_{10,13} - \mathrm{L}_{10,13}^2 = 0 \tag{3.47}$$

Front right upper wishbone arm – This element is defined by points p_{11}, p_{14} and unit vectors $\mathbf{u}_1$ and $\mathbf{u}_{11}$, as shown in fig. 3.4. $\mathbf{u}_{11}$, $\mathbf{u}_1$ and segment (11–14) form the coordinate system of the element with origin in p_{11}. An extra point, p_{15}, is defined to consider the spherical joint between this element and the front right damper. 9 new variables (p_{14}, p_{15} and $\mathbf{u}_{11}$) are added. It is worth mentioning that as point p_{11} and vector $\mathbf{u}_1$ have been previously defined during the chassis modeling, a revolute joint between the chassis and the front right upper wishbone arm is automatically considered. The constraints for this element are rigid body constraints, eqs. (3.48) to (3.52) and a linear combination, eq. (3.53). As 9 new variables and 8 constraints (5 scalar and one vectorial) are added, a new DOF is defined. Considering the previously defined DOFs, the DOF number at this stage of the modeling is 8. The body mass is 9.5 kg. It is composed of the mass of the wishbone arm (4.2 kg) as well as the mass of the damper (5.3 kg), which is considered to be a point mass applied at p_{15}. This approximation allows to take into the damper mass as the damper is not a body of the MB model. The position of the COG and the inertia tensor of this element, expressed in the

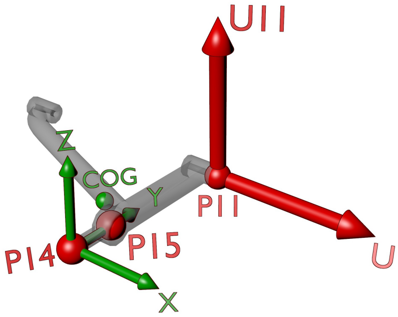


Figure 3.4: Points, vectors, COG, reference set of the front right upper wishbone arm

COG position (m)		moments of inertia (kg m^2)			
X	-0.048	I_{xx}	0.05	I_{xy}	-0.017
Y	0.119	I_{yy}	0.09	I_{xz}	0.000
Z	0.000	I_{zz}	0.139	I_{yz}	0.000

Table 3.4: COG coordinates and inertia tensor for the front right upper wishbone arm

reference set shown in fig. 3.4, are summarized in table 3.4.

$$\mathbf{u}_{11} \cdot \mathbf{u}_{11} - 1 = 0 \tag{3.48}$$

$$\mathbf{u}_{1} \cdot \mathbf{u}_{11} - \cos \phi_{1,11} = 0 \tag{3.49}$$

$$\mathbf{u}_{1} \cdot \mathbf{r}_{11,14} - \cos \phi_{1,11-14} = 0 \tag{3.50}$$

$$\mathbf{u}_{11} \cdot \mathbf{r}_{11,14} - \cos \phi_{11,11-14} = 0 \tag{3.51}$$

$$\mathbf{r}_{11,14} \cdot \mathbf{r}_{11,14} - \mathrm{L}_{11,14}^2 = 0 \tag{3.52}$$

$$\mathbf{r}_{11,15} - \alpha_{15} \cdot \mathbf{r}_{11,14} - \beta_{15} \cdot \mathbf{u}_{1} - \gamma_{15} \cdot \mathbf{u}_{11} = 0 \tag{3.53}$$

Front right wheel knuckle – This element is defined by points p_{13}, p_{14}, p_{16}, p_{17} and unit vector $\mathbf{u}_{12}$, as shown in fig. 3.5. $\mathbf{u}_{12}$, segments (14–13) and (14–16) form the coordinate system of the element with origin in p_{14}. An extra point (p_{17}) is defined to consider the revolute joint between this element and the front right wheel. 9 new variables (p_{16}, p_{17} and $\mathbf{u}_{12}$) are added. It is worth mentioning that as points p_{13} and p_{14} have been previously defined during the modeling of the wishbone arms, two spherical joints between the wishbone arms and the front right wheel knuckle are automatically considered. The constraints for this element are rigid body constraints, eqs. (3.54) to (3.59) and a linear combination, eq. (3.60). As 9 new variables and 9 constraints (6 scalar and one vectorial) are added, the number of DOF remains unchanged. In fact, the rotation of the knuckle due to the steering system has been added but the distance between p_{13} and p_{14} has been set constant.

This element is composed of the wheel knuckle and the brake caliper that is rigidly attached to it. The body mass is 7.4 kg. The position of the COG and the inertia tensor of

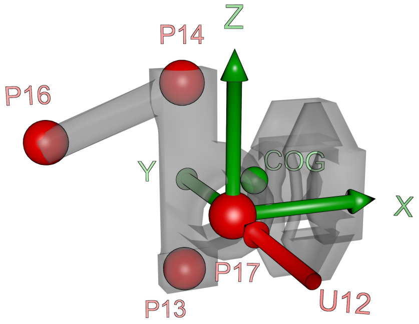


Figure 3.5: Points, vectors, COG, reference set of the front right wheel knuckle

COG position (m)		moments of inertia (kg m^2)			
X	+0.043	I_{xx}	+0.023	I_{xy}	-0.005
Y	+0.055	I_{yy}	+0.044	I_{xz}	-0.007
Z	+0.011	I_{zz}	+0.036	I_{yz}	+0.002

Table 3.5: COG coordinates and inertia tensor for the front right wheel knuckle

this element, expressed in the reference set shown in fig. 3.5, are summarized in table 3.5.

$$\mathbf{u}_{12} \cdot \mathbf{u}_{12} - 1 = 0 \tag{3.54}$$

$$\mathbf{r}_{14,13} \cdot \mathbf{r}_{14,13} - \mathrm{L}^2_{14,13} = 0 \tag{3.55}$$

$$\mathbf{r}_{14,16} \cdot \mathbf{r}_{14,16} - \mathrm{L}^2_{14,16} = 0 \tag{3.56}$$

$$\mathbf{r}_{14,13} \cdot \mathbf{r}_{14,16} - \cos\phi_{14-13,14-16} = 0 \tag{3.57}$$

$$\mathbf{r}_{14,13} \cdot \mathbf{u}_{12} - \cos\phi_{14-13,12} = 0 \tag{3.58}$$

$$\mathbf{r}_{14,16} \cdot \mathbf{u}_{12} - \cos\phi_{14-16,12} = 0 \tag{3.59}$$

$$\mathbf{r}_{14,17} - \alpha_{17} \cdot \mathbf{r}_{14,13} - \beta_{17} \cdot \mathbf{r}_{14,16} - \gamma_{17} \cdot \mathbf{u}_{12} = 0 \tag{3.60}$$

Front left lower wishbone arm – This element is defined by points p_{20}, p_{23} and unit vectors $\mathbf{u}_1$ and $\mathbf{u}_{20}$, as shown in fig. 3.6. $\mathbf{u}_1$, $\mathbf{u}_{20}$ and segment (20–23) form the coordinate system of the element with origin in p_{20}. 6 new variables (p_{23} and $\mathbf{u}_{20}$) are added. It is worth mentioning that as point p_{20} and vector $\mathbf{u}_1$ have been previously defined during the chassis modeling, the revolute joint between the chassis and the front left lower wishbone arm is automatically considered. The constraints for this element are rigid body constraints, eqs. (3.61) to (3.65). As 6 new variables and 5 constraints are added, a new DOF is defined. Considering the previously defined DOFs, the DOF number at this stage of the modeling is 9. The body mass is 4.2 kg. The position of the COG and the inertia tensor of this element,

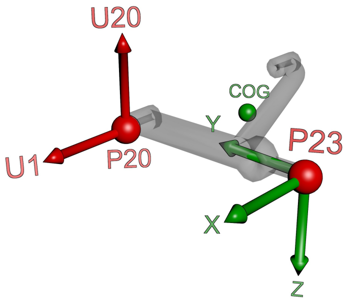


Figure 3.6: Points, vectors, COG, reference set of the front left lower wishbone arm

COG position (m)		moments of inertia (kg m^2)			
X	-0.109	I_{xx}	+0.020	I_{xy}	-0.017
Y	+0.182	I_{yy}	+0.061	I_{xz}	+0.000
Z	+0.0	I_{zz}	+0.081	I_{yz}	+0.000

Table 3.6: COG coordinates and inertia tensor for the front left lower wishbone arm

expressed in the reference set shown in fig. 3.6, are summarized in table 3.6.

$$\mathbf{u}_{20} \cdot \mathbf{u}_{20} - 1 = 0 \tag{3.61}$$

$$\mathbf{r}_{20,23} \cdot \mathbf{r}_{20,23} - \mathrm{L}^2_{20,23} = 0 \tag{3.62}$$

$$\mathbf{u}_1 \cdot \mathbf{u}_{20} - \cos \phi_{1,20} = 0 \tag{3.63}$$

$$\mathbf{r}_{20,23} \cdot \mathbf{u}_1 - \cos \phi_{20-23,1} = 0 \tag{3.64}$$

$$\mathbf{r}_{20,23} \cdot \mathbf{u}_{20} - \cos \phi_{20-23,20} = 0 \tag{3.65}$$

Front left upper wishbone arm – This element is defined by points p_{21}, p_{24} and unit vectors $\mathbf{u}_1$ and $\mathbf{u}_{21}$, as shown in fig. 3.7. $\mathbf{u}_{21}$, $\mathbf{u}_1$ and segment (21–24) form the coordinate system of the element with origin in p_{21}. An extra point, p_{25}, is defined to consider the spherical joint between this element and the front left damper. 9 new variables (p_{24}, p_{25} and $\mathbf{u}_{21}$) are added. It is worth mentioning that as point p_{21} and vector $\mathbf{u}_1$ have been previously defined during the chassis modeling, a revolute joint between the chassis and the front left upper wishbone arm is automatically considered. The constraints for this element are rigid body constraints, eqs. (3.66) to (3.70) and a linear combination, eq. (3.71). As 9 new variables and 8 constraints (5 scalar and one vectorial) are added, a new DOF is defined. Considering the previously defined DOFs, the DOF number at this stage of the modeling is 10. The body mass is 4.2 kg. It is composed of the mass of the wishbone arm (4.2 kg) as well as the mass of the damper (5.3 kg), which is considered to be a point mass applied at p_{15}. This approximation allows to take into the damper mass as the damper is not a body of the MB model. The position of the COG and the inertia tensor of this element, expressed in the

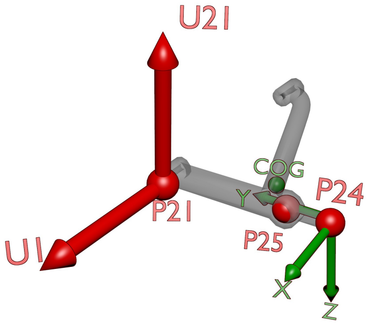


Figure 3.7: Points, vectors, COG, reference set of the front left upper wishbone arm

COG position (m)		moments of inertia (kg m^2)			
X	-0.048	I_{xx}	0.05	I_{xy}	-0.017
Y	0.119	I_{yy}	0.09	I_{xz}	0.000
Z	0.000	I_{zz}	0.139	I_{yz}	0.000

Table 3.7: COG coordinates and inertia tensor for the front left upper wishbone arm

reference set shown in fig. 3.7, are summarized in table 3.7.

$$\mathbf{u}_{21} \cdot \mathbf{u}_{21} - 1 = 0 \tag{3.66}$$

$$\mathbf{u}_{1} \cdot \mathbf{u}_{21} - \cos\phi_{1,21} = 0 \tag{3.67}$$

$$\mathbf{u}_{1} \cdot \mathbf{r}_{21,24} - \cos\phi_{1,21-24} = 0 \tag{3.68}$$

$$\mathbf{u}_{21} \cdot \mathbf{r}_{21,24} - \cos\phi_{21,21-24} = 0 \tag{3.69}$$

$$\mathbf{r}_{21,24} \cdot \mathbf{r}_{21,24} - \mathrm{L}^2_{21,24} = 0 \tag{3.70}$$

$$\mathbf{r}_{21,25} - \alpha_{25} \cdot \mathbf{r}_{21,25} - \beta_{25} \cdot \mathbf{u}_{1} - \gamma_{25} \cdot \mathbf{u}_{21} = 0 \tag{3.71}$$

Front left wheel knuckle – This element is defined by points p_{23}, p_{24}, p_{26}, p_{27} and unit vector $\mathbf{u}_{22}$, as shown in fig. 3.8. $\mathbf{u}_{22}$, segments (24–23) and (24–26) form the coordinate system of the element with origin in p_{24}. An extra point (p_{27}) is defined to consider the revolute joint between this element and the front left wheel. 9 new variables (p_{26}, p_{27} and $\mathbf{u}_{22}$) are added. It is worth mentioning that as points p_{23} and p_{24} have been previously defined during the modeling of the wishbone arms, two spherical joints between the wishbone arms and the front left wheel knuckle are automatically considered. The constraints for this element are rigid body constraints, eqs. (3.72) to (3.77) and a linear combination, eq. (3.60). As 9 new variables and 9 constraints (6 scalar and one vectorial) are added, the number of DOF remains unchanged. In fact, the rotation of the knuckle due to the steering system has been added but the distance between p_{23} and p_{24} has been set constant.

This element is composed of the wheel knuckle and the brake caliper that is rigidly attached to it. The body mass is 7.4 kg. The position of the COG and the inertia tensor of

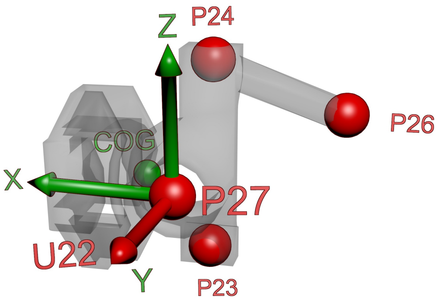


Figure 3.8: Points, vectors, COG, reference set of the front left wheel knuckle

COG position (m)		moments of inertia (kg m^2)			
X	+0.043	I_{xx}	+0.023	I_{xy}	+0.005
Y	-0.055	I_{yy}	+0.044	I_{xz}	-0.007
Z	+0.011	I_{zz}	+0.036	I_{yz}	-0.002

Table 3.8: COG coordinates and inertia tensor for the front left wheel knuckle

this element, expressed in the reference set shown in fig. 3.8, are summarized in table 3.8.

$$\mathbf{u}_{22} \cdot \mathbf{u}_{22} - 1 = 0 \tag{3.72}$$

$$\mathbf{r}_{24,23} \cdot \mathbf{r}_{24,23} - \mathrm{L}^2_{24,23} = 0 \tag{3.73}$$

$$\mathbf{r}_{24,26} \cdot \mathbf{r}_{24,26} - \mathrm{L}^2_{24,26} = 0 \tag{3.74}$$

$$\mathbf{r}_{24,23} \cdot \mathbf{r}_{24,26} - \cos\phi_{24-23,24-26} = 0 \tag{3.75}$$

$$\mathbf{r}_{24,23} \cdot \mathbf{u}_{22} - \cos\phi_{24-23,22} = 0 \tag{3.76}$$

$$\mathbf{r}_{24,26} \cdot \mathbf{u}_{22} - \cos\phi_{24-26,22} = 0 \tag{3.77}$$

$$\mathbf{r}_{24,27} - \alpha_{27} \cdot \mathbf{r}_{24,23} - \beta_{27} \cdot \mathbf{r}_{24,26} - \gamma_{27} \cdot \mathbf{u}_{22} = 0 \tag{3.78}$$

Steering system – This element is defined by points p_{50}, p_{51} and unit vectors $\mathbf{u}_1$ and $\mathbf{u}_3$, as shown in fig. 3.9. $\mathbf{u}_1$, $\mathbf{u}_3$ and segment (50–51) form the coordinate system of the element with origin in p_{50}. 6 new variables (p_{50} and p_{51}) are added. The constraints for this element are rigid body constraints, eqs. (3.79) to (3.81) and a cylindrical joint constraint, eq. (3.82) of which only two are independent. As 6 new variables and 5 constraints are added, a new DOF is defined. Considering the previously defined DOFs, the DOF number at this stage of the modeling is 11. The body mass is 0.57 kg. The position of the COG and the inertia tensor of this element, expressed in the reference set shown in fig. 3.9, are summarized in table 3.9.

$$\mathbf{r}_{50,51} \cdot \mathbf{r}_{50,51} - \mathrm{L}^2_{50,51} = 0 \tag{3.79}$$

$$\mathbf{r}_{50,51} \cdot \mathbf{u}_1 - \cos\phi_{50-51,1} = 0 \tag{3.80}$$

$$\mathbf{r}_{50,51} \cdot \mathbf{u}_3 - \cos\phi_{50-51,3} = 0 \tag{3.81}$$

$$\mathbf{r}_{50,2} \wedge \mathbf{r}_{51,2} = 0 \tag{3.82}$$

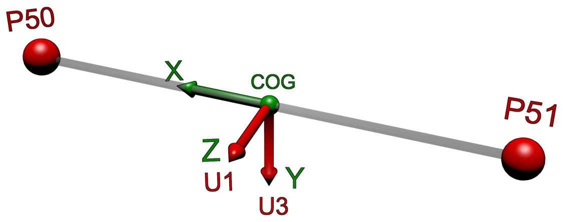


Figure 3.9: Points, vectors, COG, reference set of the steering system

COG position (m)		moments of inertia (kg m^2)		
X	+0.000	I_{xx}	0.00001	I_{xy}
Y	+0.000	I_{yy}	0.0198	I_{xz}
Z	+0.000	I_{zz}	0.0198	I_{yz}

Table 3.9: COG coordinates and inertia tensor for the steering system

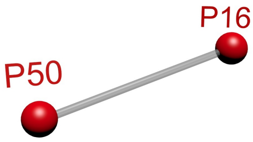


Figure 3.10: Points, vectors, COG, reference set of the right tie rod

Right tie rod – This element is defined by points p_{50} and p_{16} as shown in fig. 3.10. No new variables are added. The only constraint for this element is a rigid body constraint, eq. (3.83). As no new variables and 1 constraint are added, a DOF disappears. In fact, the rotation of the right wheel knuckle is now constrained by the translation of the steering system. Considering the previously defined DOFs, the DOF number at this stage of the modeling is 10. The body mass is 0.29 kg. The inertia of this element is one of a bar. For example, at point 50, this inertia is 0.030 kg m^2 as the tie rod length is 0.32 m.

$$\mathbf{r}_{50,16} \cdot \mathbf{r}_{50,16} - \mathrm{L}_{50,16}^2 = 0 \tag{3.83}$$

Left tie rod – This element is defined by points p_{51} and p_{26} as shown in fig. 3.11. No new variables are added. The only constraint for this element is a rigid body constraint, eq. (3.84). As no new variables and 1 constraint are added, a DOF disappears. In fact, the rotation of the left wheel knuckle is now constrained by the translation of the steering system. Considering the previously defined DOFs, the DOF number at this stage of the modeling is 9. The body mass is 0.29 kg. For example, at point 51, this inertia is 0.030 kg m^2 as the tie rod

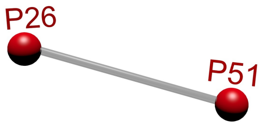


Figure 3.11: Points, vectors, COG, reference set of the left tie rod

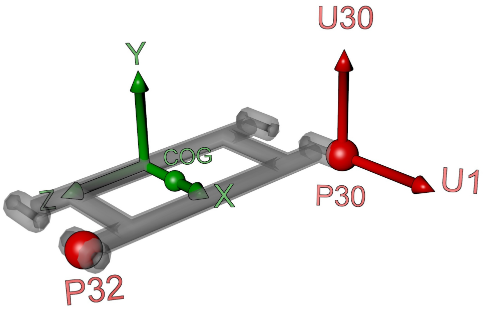


Figure 3.12: Points, vectors, COG, reference set of the rear right wishbone arm

length is 0.32 m.

$$\mathbf{r}_{51,26} \cdot \mathbf{r}_{51,26} - \mathrm{L}^2_{51,26} = 0 \tag{3.84}$$

Rear right wishbone arm – This element is defined by points p_{30}, p_{32} and unit vectors $\mathbf{u}_1$ and $\mathbf{u}_{30}$, as shown in fig. 3.12. $\mathbf{u}_1$, $\mathbf{u}_{30}$ and segment (30–32) form the coordinate system of the element with origin in p_{30}. 6 new variables (p_{32} and $\mathbf{u}_{30}$) are added. It is worth mentioning that as point p_{30} and vector $\mathbf{u}_1$ have been previously defined during the chassis modeling, the revolute joint between the chassis and the rear right wishbone arm is automatically considered. The constraints for this element are rigid body constraints, eqs. (3.85) to (3.89). As 6 new variables and 5 constraints are added, a new DOF is defined. Considering the previously defined DOFs, the DOF number at this stage of the modeling is 10. The body mass is 2.7 kg. The position of the COG and the inertia tensor of this element, expressed in the reference set shown in fig. 3.12, are summarized in table 3.10.

$$\mathbf{u}_{30} \cdot \mathbf{u}_{30} - 1 = 0 \tag{3.85}$$

$$\mathbf{r}_{30,32} \cdot \mathbf{r}_{30,32} - \mathrm{L}^2_{30,32} = 0 \tag{3.86}$$

$$\mathbf{u}_1 \cdot \mathbf{u}_{30} - \cos\phi_{1,30} = 0 \tag{3.87}$$

$$\mathbf{r}_{30,32} \cdot \mathbf{u}_1 - \cos\phi_{30-32,1} = 0 \tag{3.88}$$

$$\mathbf{r}_{30,32} \cdot \mathbf{u}_{30} - \cos\phi_{30-32,30} = 0 \tag{3.89}$$

Rear left wishbone arm – This element is defined by points p_{40}, p_{42} and unit vectors $\mathbf{u}_1$ and $\mathbf{u}_{40}$, as shown in fig. 3.13. $\mathbf{u}_1$, $\mathbf{u}_{40}$ and segment (40–42) form the coordinate system of the element with origin in p_{40}. 6 new variables (p_{42} and $\mathbf{u}_{40}$) are added. It is worth mentioning

COG position (m)		moments of inertia (kg m^2)			
X	+0.060	I_{xx}	+0.055	I_{xy}	+0.000
Y	+0.000	I_{yy}	+0.064	I_{xz}	+0.000
Z	+0.000	I_{zz}	+0.009	I_{yz}	+0.000

Table 3.10: COG coordinates and inertia tensor for the rear right wishbone arm

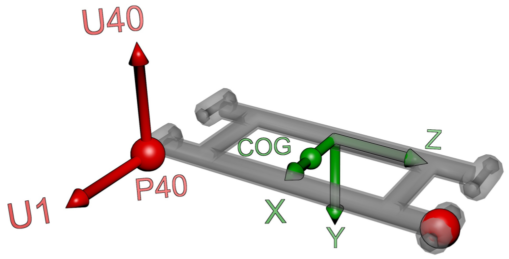


Figure 3.13: Points, vectors, COG, reference set of the rear left wishbone arm

COG position (m)		moments of inertia (kg m^2)			
X	+0.060	I_{xx}	+0.055	I_{xy}	+0.000
Y	+0.000	I_{yy}	+0.064	I_{xz}	+0.000
Z	+0.000	I_{zz}	+0.009	I_{yz}	+0.000

Table 3.11: COG coordinates and inertia tensor for the rear left wishbone arm

that as point p_{40} and vector $\mathbf{u}_1$ have been previously defined during the chassis modeling, the revolute joint between the chassis and the rear left wishbone arm is automatically considered. The constraints for this element are rigid body constraints, eqs. (3.90) to (3.94). As 6 new variables and 5 constraints are added, a new DOF is defined. Considering the previously defined DOFs, the DOF number at this stage of the modeling is 11. The body mass is 2.7 kg. The position of the COG and the inertia tensor of this element, expressed in the reference set shown in fig. 3.13, are summarized in table 3.11.

$$\mathbf{u}_{40} \cdot \mathbf{u}_{40} - 1 = 0 \tag{3.90}$$

$$\mathbf{r}_{40,42} \cdot \mathbf{r}_{40,42} - \mathrm{L}^2_{40,42} = 0 \tag{3.91}$$

$$\mathbf{u}_1 \cdot \mathbf{u}_{40} - \cos\phi_{1,40} = 0 \tag{3.92}$$

$$\mathbf{r}_{40,42} \cdot \mathbf{u}_1 - \cos\phi_{40-42,1} = 0 \tag{3.93}$$

$$\mathbf{r}_{40,42} \cdot \mathbf{u}_{40} - \cos\phi_{40-42,40} = 0 \tag{3.94}$$

Rear right wheel knuckle – This element is defined by points p_{32}, p_{34} and unit vectors $\mathbf{u}_1$, $\mathbf{u}_{31}$ as shown in fig. 3.14. $\mathbf{u}_1$, $\mathbf{u}_{31}$ and (32–34) form the coordinate system of the element with origin in p_{32}. Two extra point (p_{33} and p_{35}) are defined. The first one is used to define

COG position (m)		moments of inertia (kg m^2)			
X	-0.117	I_{xx}	+0.284	I_{xy}	-0.156
Y	+0.080	I_{yy}	+0.169	I_{xz}	-0.032
Z	-0.031	I_{zz}	+0.370	I_{yz}	+0.035

Table 3.12: COG coordinates and inertia tensor for the rear right wheel knuckle

automatically the revolute joint between this element and the rear right wheel while the second one is employed to define the cylindrical joint of the rear right damper. 12 new variables (p_{33}, p_{34}, p_{35} and $\mathbf{u}_{31}$) are added. It is worth mentioning that as point p_{32} and unit vector u_1 have been previously defined during the rear right wishbone arm modeling, a revolute joint between this wishbone arm and the rear right wheel knuckle is automatically considered. The constraints for this element are rigid body constraints, eqs. (3.95) to (3.99), linear combinations, eqs. (3.100) and (3.101), and a cylindrical joint constraint, eq. (3.102). As 12 new variables and 12 constraints (5 scalar and 7 vectorial) are added, the number of DOF remains unchanged (i.e 11). In fact, only 12 of the 14 constraint equations are independent. Two of the three equations in eq. (3.102) are independent and p_{33} is first defined as a linear combination of the coordinate set and then aligned with p_{31} and p_{34}. To give a physical significance to these constraints, it can be said that before the definition of the rear right wheel knuckle, the rear right wishbone arm had one DOF. After this definition, a revolute joint between the wheel knuckle and the wishbone arm appears but the motion of this joint is constrained by the motion of the damper attached to the chassis. As a consequence, the DOF number remains unchanged despite the definition of new bodies and constraints.

This element is composed of the wheel knuckle, the damper and the brake caliper that are rigidly interconnected. The body mass is 14.1 kg. The position of the COG and the inertia tensor of this element, expressed in the reference set shown in fig. 3.14, are summarized in table 3.12.

$$\mathbf{u}_{31} \cdot \mathbf{u}_{31} - 1 = 0 \tag{3.95}$$

$$\mathbf{r}_{32,34} \cdot \mathbf{r}_{32,34} - \mathrm{L}^2_{32,34} = 0 \tag{3.96}$$

$$\mathbf{u}_1 \cdot \mathbf{u}_{31} - \cos\phi_{1,31} = 0 \tag{3.97}$$

$$\mathbf{r}_{32,34} \cdot \mathbf{u}_1 - \cos\phi_{32-34,1} = 0 \tag{3.98}$$

$$\mathbf{r}_{32,34} \cdot \mathbf{u}_{31} - \cos\phi_{32-34,31} = 0 \tag{3.99}$$

$$\mathbf{r}_{32,33} - \alpha_{33} \cdot \mathbf{r}_{32,34} - \beta_{33} \cdot \mathbf{u}_1 - \gamma_{33} \cdot \mathbf{u}_{31} = 0 \tag{3.100}$$

$$\mathbf{r}_{32,35} - \alpha_{35} \cdot \mathbf{r}_{32,34} - \beta_{35} \cdot \mathbf{u}_1 - \gamma_{35} \cdot \mathbf{u}_{31} = 0 \tag{3.101}$$

$$\mathbf{r}_{33,34} \wedge \mathbf{r}_{33,31} = 0 \tag{3.102}$$

Rear left wheel knuckle – This element is defined by points p_{42}, p_{44} and unit vectors $\mathbf{u}_1$, $\mathbf{u}_{41}$ as shown in fig. 3.15. $\mathbf{u}_1$, $\mathbf{u}_{41}$ and (42–44) form the coordinate system of the element with origin in p_{42}. Two extra point (p_{43} and p_{45}) are defined. The first one is used to define automatically the revolute joint between this element and the rear left wheel while the second one is employed to define the cylindrical joint of the rear left damper. 12 new variables (p_{43}, p_{44}, p_{45} and $\mathbf{u}_{41}$) are added. It is worth mentioning that as point p_{42} and unit vector

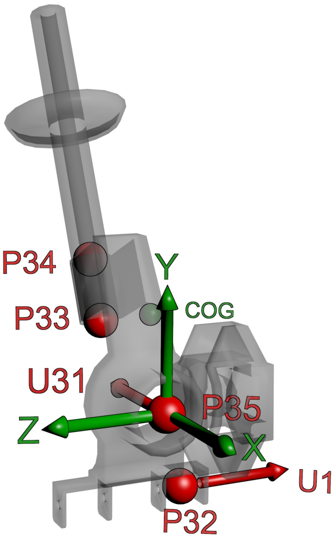


Figure 3.14: Points, vectors, COG, reference set of the rear right wheel knuckle

u_1 have been previously defined during the rear left wishbone arm modeling, a revolute joint between this wishbone arm and the rear left wheel knuckle is automatically considered. The constraints for this element are rigid body constraints, eqs. (3.103) to (3.107), linear combinations, eqs. (3.108) and (3.109), and a cylindrical joint constraint, eq. (3.110). As 12 new variables and 12 constraints (5 scalar and 7 vectorial) are added, the number of DOF remains unchanged (i.e 11). In fact, only 12 of the 14 constraint equations are independent. Two of the three equations in eq. (3.110) are independent and p_{43} is first defined as a linear combination of the coordinate set and then aligned with p_{41} and p_{44}. To give a physical significance to these constraints, it can be said that before the definition of the rear left wheel knuckle, the rear left wishbone arm had one DOF. After this definition, a revolute joint between the wheel knuckle and the wishbone arm appears but the motion of this joint is constrained by the motion of the damper attached to the chassis. As a consequence, the DOF number remains unchanged despite the definition of new bodies and constraints.

This element is composed of the wheel knuckle, the damper and the brake caliper that are rigidly interconnected. The body mass is 14.1 kg. The position of the COG and the inertia tensor of this element, expressed in the reference set shown in fig. 3.15, are summarized in

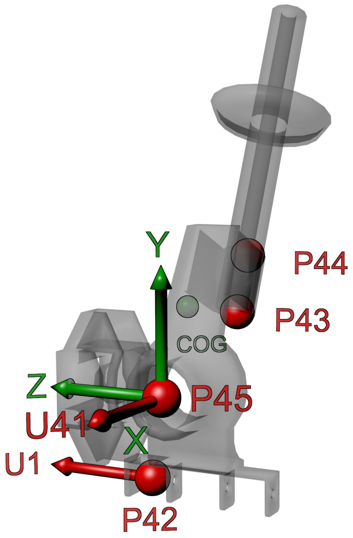


Figure 3.15: Points, vectors, COG, reference set of the rear left wheel knuckle

COG position (m)		moments of inertia (kg m^2)			
X	-0.117	I_{xx}	+0.284	I_{xy}	-0.156
Y	+0.080	I_{yy}	+0.169	I_{xz}	+0.032
Z	+0.031	I_{zz}	+0.370	I_{yz}	-0.035

Table 3.13: COG coordinates and inertia tensor for the rear left wheel knuckle

table 3.13.

$$\mathbf{u}_{41} \cdot \mathbf{u}_{41} - 1 = 0 \tag{3.103}$$
$$\mathbf{r}_{42,44} \cdot \mathbf{r}_{42,44} - \mathrm{L}_{42,44}^2 = 0 \tag{3.104}$$
$$\mathbf{u}_{1} \cdot \mathbf{u}_{41} - \cos \phi_{1,41} = 0 \tag{3.105}$$
$$\mathbf{r}_{42,44} \cdot \mathbf{u}_{1} - \cos \phi_{42-44,1} = 0 \tag{3.106}$$
$$\mathbf{r}_{42,44} \cdot \mathbf{u}_{41} - \cos \phi_{42-44,41} = 0 \tag{3.107}$$
$$\mathbf{r}_{42,43} - \alpha_{43} \cdot \mathbf{r}_{42,44} - \beta_{43} \cdot \mathbf{u}_{1} - \gamma_{43} \cdot \mathbf{u}_{41} = 0 \tag{3.108}$$
$$\mathbf{r}_{42,45} - \alpha_{45} \cdot \mathbf{r}_{42,44} - \beta_{45} \cdot \mathbf{u}_{1} - \gamma_{45} \cdot \mathbf{u}_{41} = 0 \tag{3.109}$$
$$\mathbf{r}_{43,44} \wedge \mathbf{r}_{43,41} = 0 \tag{3.110}$$

Front right wheel – This element is defined by points p_{17} and unit vectors $\mathbf{u}_{12}$, $\mathbf{u}_{13}$ and $\mathbf{u}_{14}$, as shown in fig. 3.16. $\mathbf{u}_{12}$, $\mathbf{u}_{13}$ and $\mathbf{u}_{14}$ form the coordinate system of the element with origin in p_{17}. 6 new variables ($\mathbf{u}_{13}$ and $\mathbf{u}_{14}$) are added. It is worth mentioning that as point p_{17} and vector $\mathbf{u}_{12}$ have been previously defined during the front right wheel knuckle modeling, the revolute joint between the wheel and the wheel knuckle is automatically considered. The constraints for this element are rigid body constraints, eqs. (3.111) to (3.115). As 6 new

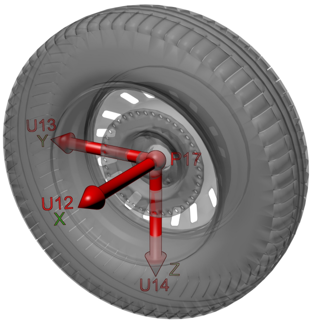


Figure 3.16: Points, vectors, COG, reference set of the front right wheel assembly

COG position (m)		moments of inertia (kg m^2)			
X	+0.065	I_{xx}	+0.534	I_{xy}	+0.000
Y	+0.000	I_{yy}	+0.300	I_{xz}	+0.000
Z	+0.000	I_{zz}	+0.300	I_{yz}	+0.000

Table 3.14: COG coordinates and inertia tensor for the front right wheel

variables and 5 constraints are added, a new DOF is defined. Considering the previously defined DOFs, the DOF number at this stage of the modeling is 12.

This element is composed of the wheel rim, the tire, the wheel hub, the brake disc and the drive flange that are rigidly interconnected. The body mass is 15.1 kg. The position of the COG and the inertia tensor of this element, expressed in the reference set shown in fig. 3.16, are summarized in table 3.14.

$$\mathbf{u}_{13} \cdot \mathbf{u}_{13} - 1 = 0 \tag{3.111}$$

$$\mathbf{u}_{14} \cdot \mathbf{u}_{14} - 1 = 0 \tag{3.112}$$

$$\mathbf{u}_{12} \cdot \mathbf{u}_{13} - \cos \phi_{12,13} = 0 \tag{3.113}$$

$$\mathbf{u}_{12} \cdot \mathbf{u}_{14} - \cos \phi_{12,14} = 0 \tag{3.114}$$

$$\mathbf{u}_{13} \cdot \mathbf{u}_{14} - \cos \phi_{13,14} = 0 \tag{3.115}$$

Front left wheel – This element is defined by points p_{27} and unit vectors $\mathbf{u}_{22}$, $\mathbf{u}_{23}$ and $\mathbf{u}_{24}$, as shown in fig. 3.17. $\mathbf{u}_{22}$, $\mathbf{u}_{23}$ and $\mathbf{u}_{24}$ form the coordinate system of the element with origin in p_{27}. 6 new variables ($\mathbf{u}_{23}$ and $\mathbf{u}_{24}$) are added. It is worth mentioning that as point p_{27} and vector $\mathbf{u}_{22}$ have been previously defined during the front left wheel knuckle modeling, the revolute joint between the wheel and the wheel knuckle is automatically considered. The constraints for this element are rigid body constraints, eqs. (3.116) to (3.120). As 6 new variables and 5 constraints are added, a new DOF is defined. Considering the previously defined DOFs, the DOF number at this stage of the modeling is 13.

This element is composed of the wheel rim, the tire, the wheel hub and the brake disc that are rigidly interconnected whose masses are presented in table 3.15. The position of the COG and the inertia tensor of this element, expressed in the reference set shown in fig. 3.17,

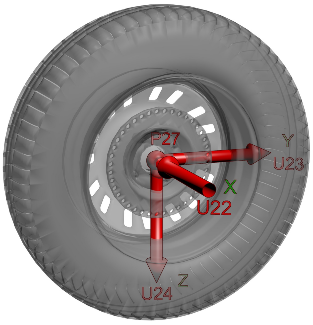


Figure 3.17: Points, vectors, COG, reference set of the front left wheel assembly

Body	number	mass (kg)
rim	1	6
tire	1	5.1
disk brake	1	2.7
hub	1	1.3
Total		15.1

Table 3.15: Front left wheel mass properties

COG position (m)		moments of inertia (kg m^2)			
X	+0.065	I_{xx}	+0.534	I_{xy}	+0.000
Y	+0.000	I_{yy}	+0.300	I_{xz}	+0.000
Z	+0.000	I_{zz}	+0.300	I_{yz}	+0.000

Table 3.16: COG coordinates and inertia tensor for the front left wheel

are summarized in table 3.16.

$$\mathbf{u}_{23} \cdot \mathbf{u}_{23} - 1 = 0 \tag{3.116}$$

$$\mathbf{u}_{24} \cdot \mathbf{u}_{24} - 1 = 0 \tag{3.117}$$

$$\mathbf{u}_{22} \cdot \mathbf{u}_{23} - \cos\phi_{22,23} = 0 \tag{3.118}$$

$$\mathbf{u}_{22} \cdot \mathbf{u}_{24} - \cos\phi_{22,24} = 0 \tag{3.119}$$

$$\mathbf{u}_{23} \cdot \mathbf{u}_{24} - \cos\phi_{23,24} = 0 \tag{3.120}$$

Rear right wheel – This element is defined by points p_{35} and unit vectors $\mathbf{u}_{31}$, $\mathbf{u}_{32}$ and $\mathbf{u}_{33}$, as shown in fig. 3.18. $\mathbf{u}_{31}$, $\mathbf{u}_{32}$ and $\mathbf{u}_{33}$ form the coordinate system of the element with origin in p_{35}. 6 new variables ($\mathbf{u}_{32}$ and $\mathbf{u}_{33}$) are added. It is worth mentioning that as point p_{35} and vector $\mathbf{u}_{31}$ have been previously defined during the rear right wheel knuckle modeling, the revolute joint between the wheel and the wheel knuckle is automatically considered. The constraints for this element are rigid body constraints, eqs. (3.121) to (3.125). As 6 new

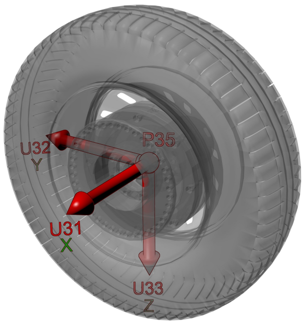


Figure 3.18: Points, vectors, COG, reference set of the rear right wheel

Body	mass (kg)
rim	6
tire	5.1
disk brake	3.2
hub	1.7
torque sensor	14
Total	30

Table 3.17: Rear right wheel mass properties

variables and 5 constraints are added, a new DOF is defined. Considering the previously defined DOFs, the DOF number at this stage of the modeling is 14.

This element is composed of the wheel rim, the tire, the wheel hub, the brake disc and the torque sensor that are rigidly interconnected and whose masses are presented in table 3.17. The position of the COG and the inertia tensor of this element, expressed in the reference set shown in fig. 3.18, are summarized in table 3.18.

$$\mathbf{u}_{32} \cdot \mathbf{u}_{32} - 1 = 0 \tag{3.121}$$

$$\mathbf{u}_{33} \cdot \mathbf{u}_{33} - 1 = 0 \tag{3.122}$$

$$\mathbf{u}_{31} \cdot \mathbf{u}_{32} - \cos \phi_{31,32} = 0 \tag{3.123}$$

$$\mathbf{u}_{31} \cdot \mathbf{u}_{33} - \cos \phi_{31,33} = 0 \tag{3.124}$$

$$\mathbf{u}_{32} \cdot \mathbf{u}_{33} - \cos \phi_{32,33} = 0 \tag{3.125}$$

Rear left wheel – This element is defined by points p_{45} and unit vectors $\mathbf{u}_{41}$, $\mathbf{u}_{42}$ and $\mathbf{u}_{43}$, as shown in fig. 3.19. $\mathbf{u}_{41}$, $\mathbf{u}_{42}$ and $\mathbf{u}_{43}$ form the coordinate system of the element with origin in p_{45}. 6 new variables ($\mathbf{u}_{42}$ and $\mathbf{u}_{43}$) are added. It is worth mentioning that as point p_{45} and vector $\mathbf{u}_{41}$ have been previously defined during the rear left wheel knuckle modeling, the revolute joint between the wheel and the wheel knuckle is automatically considered. The constraints for this element are rigid body constraints, eqs. (3.126) to (3.130). As 6 new variables and 5 constraints are added, a new DOF is defined. Considering the previously defined DOFs, the DOF number at this stage of the modeling is 15.

COG position (m)		moments of inertia (kg m^2)			
X	+0.077	I_{xx}	+0.647	I_{xy}	+0.000
Y	+0.000	I_{yy}	+0.393	I_{xz}	+0.000
Z	+0.000	I_{zz}	+0.393	I_{yz}	+0.000

Table 3.18: COG coordinates and inertia tensor for the rear right wheel

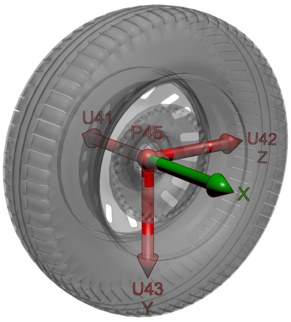


Figure 3.19: Points, vectors, COG, reference set of the rear left wheel

Body	mass (kg)
rim	6
tire	5.1
disk brake	3.2
hub	1.7
Total	16

Table 3.19: Rear left wheel mass properties

This element is composed of the wheel rim, the tire, the wheel hub and the brake disc that are rigidly interconnected and whose masses are presented in table 3.19. The position of the COG and the inertia tensor of this element, expressed in the reference set shown in fig. 3.19, are summarized in table 3.20.

$$\mathbf{u}_{42} \cdot \mathbf{u}_{42} - 1 = 0 \tag{3.126}$$

$$\mathbf{u}_{43} \cdot \mathbf{u}_{43} - 1 = 0 \tag{3.127}$$

$$\mathbf{u}_{41} \cdot \mathbf{u}_{42} - \cos\phi_{41,42} = 0 \tag{3.128}$$

$$\mathbf{u}_{41} \cdot \mathbf{u}_{43} - \cos\phi_{41,43} = 0 \tag{3.129}$$

$$\mathbf{u}_{42} \cdot \mathbf{u}_{43} - \cos\phi_{42,43} = 0 \tag{3.130}$$

COG position (m)		moments of inertia (kg m^2)			
X	+0.065	I_{xx}	+0.534	I_{xy}	+0.000
Y	+0.000	I_{yy}	+0.300	I_{xz}	+0.000
Z	+0.000	I_{zz}	+0.300	I_{yz}	+0.000

Table 3.20: COG coordinates and inertia tensor for the rear left wheel

Extra variables

The set of natural coordinates has been supplemented with angles and distances that describe the relative motion of some kinematic joints. This information allows for a better understanding of the model and provides some parameters required by the brake model and the tire model.

Distances – Four variables, s_{10}, s_{20}, s_{30} and s_{40}, have been introduced to define the strokes of the dampers and one more, s_{50} to define the steering translation. Four equations have been added for the dampers, eqs. (3.131) to (3.134), and another for the steering, eq. (3.135). The definition of these five distances has not affected the number of DOF as five equations have been introduced.

$$\mathrm{s}_{10} = |\mathbf{r}_{12,15}| \tag{3.131}$$

$$\mathrm{s}_{20} = |\mathbf{r}_{22,25}| \tag{3.132}$$

$$\mathrm{s}_{30} = |\mathbf{r}_{31,34}| \tag{3.133}$$

$$\mathrm{s}_{40} = |\mathbf{r}_{41,44}| \tag{3.134}$$

$$\mathrm{s}_{50} = |\mathbf{r}_{50,2}| \tag{3.135}$$

Angles – Four angles, ϕ_{10}, ϕ_{20}, ϕ_{30} and ϕ_{40}, have been introduced to define the wheel angles. For each one, two equations, never enabled at the same time, are necessary as explained in section 3.1.2. For the front wheels, the angles have been defined between the projection of two points of the knuckle onto the wheel plane and a vector contained in the latter, eqs. (3.136) to (3.139). For the rear wheels, the angles have been defined between the projection of a vector of the chassis onto the wheel plane and a vector contained in the latter, eqs. (3.140) to (3.143). The definition of these four angles has not affected the number of DOF as four equations have been introduced.

$$\mathbf{u}_{14} \cdot \mathbf{r}_{13,14} - (\mathbf{u}_{14} \cdot \mathbf{u}_{12})(\mathbf{r}_{13,14} \cdot \mathbf{u}_{12}) - \mathrm{L}_{14',14}\, cos\phi_{10} = 0 \tag{3.136}$$

$$\mathbf{u}_{14} \wedge \mathbf{r}_{13,14} - (\mathbf{u}_{14} \cdot \mathbf{u}_{12})\mathbf{u}_{12} \wedge \mathbf{r}_{13,14} - (\mathbf{r}_{13,14} \cdot \mathbf{u}_{12})\mathbf{u}_{12} \wedge \mathbf{u}_{12} - \mathbf{u}_{12}\, \mathrm{L}_{14',14}\, sin\phi_{10} = 0 \tag{3.137}$$

$$\mathbf{u}_{24} \cdot \mathbf{r}_{23,24} - (\mathbf{u}_{24} \cdot \mathbf{u}_{22})(\mathbf{r}_{23,24} \cdot \mathbf{u}_{22}) - \mathrm{L}_{24',24}\, cos\phi_{20} = 0 \tag{3.138}$$

$$\mathbf{u}_{24} \wedge \mathbf{r}_{23,24} - (\mathbf{u}_{24} \cdot \mathbf{u}_{22})\mathbf{u}_{22} \wedge \mathbf{r}_{23,24} - (\mathbf{r}_{23,24} \cdot \mathbf{u}_{22})\mathbf{u}_{22} \wedge \mathbf{u}_{22} - \mathbf{u}_{22}\, \mathrm{L}_{24',24}\, sin\phi_{20} = 0 \tag{3.139}$$

$$\mathbf{u}_{32} \cdot \mathbf{u}_{1} - (\mathbf{u}_{32} \cdot \mathbf{u}_{31})(\mathbf{u}_{1} \cdot \mathbf{u}_{31}) - \mathrm{L}_{u_{1'},u_1}\, cos\phi_{30} = 0 \tag{3.140}$$

$$\mathbf{u}_{32} \wedge \mathbf{u}_{1} - (\mathbf{u}_{32} \cdot \mathbf{u}_{31})\mathbf{u}_{31} \wedge \mathbf{u}_{1} - (\mathbf{u}_{1} \cdot \mathbf{u}_{31})\mathbf{u}_{31} \wedge \mathbf{u}_{31} - \mathbf{u}_{31}\, \mathrm{L}_{u_{1'},u_1}\, sin\phi_{30} = 0 \tag{3.141}$$

$$\mathbf{u}_{42} \cdot \mathbf{u}_{1} - (\mathbf{u}_{42} \cdot \mathbf{u}_{41})(\mathbf{u}_{1} \cdot \mathbf{u}_{41}) - \mathrm{L}_{u_{1'},u_1}\, cos\phi_{40} = 0 \tag{3.142}$$

$$\mathbf{u}_{42} \wedge \mathbf{u}_{1} - (\mathbf{u}_{42} \cdot \mathbf{u}_{41})\mathbf{u}_{41} \wedge \mathbf{u}_{1} - (\mathbf{u}_{1} \cdot \mathbf{u}_{41})\mathbf{u}_{41} \wedge \mathbf{u}_{41} - \mathbf{u}_{41}\, \mathrm{L}_{u_{1'},u_1}\, sin\phi_{40} = 0 \tag{3.143}$$

Kinematic guidance of the steering system – The steering system has been kinematically guided as the behavior of the steering wheel system is not of interest in this research. Moreover, for maneuver repeating with the XBW vehicle prototype, the steering system angle is controlled using the steering wheel angle recorded during the reference maneuver. As a consequence, the mean of the steering angles of the maneuvers can be employed to control the behavior of the steering system of the model when repeating the real test maneuvers with the MB model. The steering system displacement is calculated using the mean steering angle and the rack and pinion ratio. The general equations that relate the mean steering displacement (i.e. the reference displacement) and the steering displacement of the model are shown in eqs. (3.144) to (3.146).

$$\mathrm{s}_{50} - \mathrm{s}_{50,ref} \tag{3.144}$$

$$\dot{\mathrm{s}}_{50} - \dot{\mathrm{s}}_{50,ref} \tag{3.145}$$

$$\ddot{\mathrm{s}}_{50} - \ddot{\mathrm{s}}_{50,ref} \tag{3.146}$$

where $\mathrm{s}_{50,ref}$, $\dot{\mathrm{s}}_{50,ref}$ and $\ddot{\mathrm{s}}_{50,ref}$ are the reference variables (means of the repetition maneuvers or others if the MB model is controlled with a pad). As a rheonomic constraint is defined and no variables are created, the DOF number decreases by one.

Summary of variables and constraints

All the variables and constraints previously defined have been summarized in table 3.21 allowing for a better understanding of the modeling of the vehicle. In this table, the new variables (names and number), the constraints and the number of DOF introduced by each body are listed. The MB model has a total of 168 variables and 154 constraints without counting the ones of the subsystems. It is possible to give a physical interpretation to the 14 aforementioned DOFs: 6 DOFs go to the 3 rotations and the 3 translations of a free body, 4 DOFs to the 4 suspension deflections and 4 DOFs to the rotation of each wheel. There is no additional DOF for the steering system as it is kinematically guided.

Forces

Different kinds of forces can be considered depending on the type of maneuver. As the two maneuvers considered in this research are low speed maneuvers, the following forces have been taken into account:

- gravitational forces
- driving and braking torques of the rear wheels
- braking torques of the front wheels
- suspension forces
- tire forces

The tire and brake model are described later in section 3.1.3.

Gravitational forces – The gravitational forces have been applied in the opposite direction of unit vector $\mathbf{u}_z$ that is shown in fig. 3.1. The gravity has been taken as 9.81 m/s^2. Every body has a gravitational force applied in its COG.

Body	names of the variable	number	constraints	DOFs
chassis	$\mathbf{u}_1$, $\mathbf{u}_2$, $\mathbf{u}_3$, p_1, p_2, p_{10}, p_{11}, p_{12}, p_{20}, p_{21}, p_{22}, p_{30}, p_{31}, p_{40}, p_{41}	45	39	6
front right lower wishbone arm	$\mathbf{u}_{10}$, p_{13}	6	5	1
front right upper wishbone arm	$\mathbf{u}_{11}$, p_{14}, p_{15}	9	8	1
front right wheel knuckle	$\mathbf{u}_{12}$, p_{16}, p_{17}	9	9	0
front left lower wishbone arm	$\mathbf{u}_{20}$, p_{23}	6	5	1
front left upper wishbone arm	$\mathbf{u}_{21}$, p_{24}, p_{25}	9	8	1
front left wheel knuckle	$\mathbf{u}_{22}$, p_{26}, p_{27}	9	9	0
steering system	p_{50}, p_{51}	6	5	1
right tie rod	-	0	1	-1
left tie rod	-	0	1	-1
rear right wishbone arm	$\mathbf{u}_{30}$, p_{32}	6	5	1
rear left wishbone arm	$\mathbf{u}_{40}$, p_{42}	6	5	1
rear right wheel knuckle	$\mathbf{u}_{31}$, p_{33}, p_{34}, p_{35}	12	12	0
rear left wheel knuckle	$\mathbf{u}_{41}$, p_{43}, p_{44}, p_{45}	12	12	0
front right wheel	$\mathbf{u}_{13}$, $\mathbf{u}_{14}$	6	5	1
front left wheel	$\mathbf{u}_{23}$, $\mathbf{u}_{24}$	6	5	1
rear right wheel	$\mathbf{u}_{32}$, $\mathbf{u}_{33}$	6	5	1
rear left wheel	$\mathbf{u}_{42}$, $\mathbf{u}_{43}$	6	5	1
front right suspension distance	s_{10}	1	1	0
front left suspension distance	s_{20}	1	1	0
rear right suspension distance	s_{30}	1	1	0
rear left suspension distance	s_{40}	1	1	0
steering distance	s_{50}	1	1	0
front right wheel angle	ϕ_{10}	1	1	0
front left wheel angle	ϕ_{20}	1	1	0
rear right wheel angle	ϕ_{30}	1	1	0
rear left wheel angle	ϕ_{40}	1	1	0
kinematic guidance of the steering system	-	0	1	-1
Total	-	168	154	14

Table 3.21: Summary of variables and constraints

Driving and braking torques of the rear wheels – The driving and braking torques acting on the rear wheels of the MB model are inputs either given by the wheel torque sensor or by the user (predefined maneuver or pad commands). The driving torque is not given by an engine model as not being an objective of this research. Indeed, the development of a engine model is a hard task that here would not be worth. The wheel torque sensor measures both driving and braking torques of the rear right wheel thus eliminating the necessity of using a brake model. Both rear right and left torques have been supposed to be identical as the engine gearbox is outfitted with a non-locked differential. The torques for both rear wheels have to be introduced in the vector of generalized forces at the positions corresponding to angles ϕ_{30} and ϕ_{40}.

3.1.3 Models of the subsystems

A vehicle model not only includes the dynamics of its bodies but also the behavior of several subsystems like brakes, tires, steering system, engine, etc. In this thesis, the models of two subsystems have been considered: the tire and brake models. Both have been implemented according to the characteristics of the test maneuvers.

Tire model

Apart from the aerodynamic and gravitational forces, the rest of major forces and moments acting on a vehicle are transferred by the tires to the wheel rims. Consequently, the tire behavior and characteristics play a crucial role in the evaluation of vehicle dynamics. Over the last 60 years, the tire characteristics have been extensively and qualitatively described by many authors (Gillespie, 1992; Jazar, 2008; Popp and Schielhen, 2010; Wong, 2001). Numerous mathematical models have been developed (Chang et al., 2004; Pacejka, 2005; Pacejka and Sharp, 1991). They differ in accuracy and complexity depending on the areas of application discussed in section 1.2. As described in (Kuiper and Van Oosten, 2007), they can be classified in the following four different groups:

- *simple tire models* – They are aimed at static and quasi-static vehicle dynamics analysis and at the design of vehicle control systems. The vertical behavior consists of a linear or non-linear spring-damper model. The horizontal tire forces are accounted for by linear relationships between slip and resulting forces, and combined slip is not taken into account.

- *empirical models* –They are based on non-linear mathematical approximations of tire forces and moments or interpolation of test data, like the famous versions (1987, 1989, 1993, 1996, 2002) of the *Magic formula* (Bakker et al., 1987, 1989; Pacejka, 2002; Pacejka and Bakker, 1993; Pacejka and Besselink, 1997). Empirical tire models require full-scale tire measurements, data processing and parameter identification. These tire models are in general very accurate, and are used for vehicle dynamics analysis in a broad sense, ranging from non-linear handling to ride simulations.

- *physical models* – These models, which describe the kinematics and dynamics of the tire contact patch in detail, can be very complex. The parameters that describe the tire behavior have physical and/or geometrical significance, and extended tire measurements are not necessary. They are often tailored for a specific area of application almost unlimited: quasi-static behavior, non-linear handling, ride, comfort, durability, Noise Vibration and Harshness (NVH) and acoustics.

- *finite-element tire models* – The tire is modeled by a detailed finite element (FE) mesh for the complete tire structure including the compressed air. Almost any physical phenomenon can be taken into account and the resulting computational effort is very high. Their use is mostly restricted to detailed structural analysis with high non-linear deformations, hydroplaning and acoustic analysis. For vehicle dynamics analysis, the computational effort is yet too high for application on a regular basis.

The area of application of the tire model in this thesis is related to nonlinear vehicle handling in real–time. As a consequence, the model should be as simple as possible in order to maintain low computational cost while fulfilling the requirements imposed by the test maneuvers. As can be verified in fig. 3.28, the test track is slightly inclined (about 1.5°) and has also some bank angle in specific areas. Therefore, the tire model has to consider precisely situations where the vehicle stands still, starts or stops on an inclined test track. Then, some dynamics behaviors of the tire, like longitudinal and lateral deflections, are necessary in order to assess easily the equilibrium position of the vehicle at the beginning of the simulation while standing still. If the tire model does not manage these situations, any effort to validate the vehicle MB model would be worthless. A common difficulty in the modeling of tires is the calculation of the set of parameters that relate the longitudinal and lateral tire forces and moments to the longitudinal and lateral slips. Very few tire manufacturers sell these parameters (normally for a huge amount of money) and individual tire testing is even more expensive. Subsequently, most researchers employ tire parameters available in the literature if they correspond to their tires and, if this is not the case they extrapolate the tire parameters from tires with similar dimensions and characteristics.

In this research, part of the tire model *TMeasy* has been applied (Hirschberg et al., 2007; Rill, 2006a, 2007, 2009). This is an empirical and physical tire model, in the sense that first, curve fitting using few parameters is necessary to adjust the tire characteristic curves and then, dynamical behaviors of the tire are considered. It is similar to another recent empirical and physical tire model used in *ADAMS*: the `PAC2002` (Kuiper and Van Oosten, 2007). Both are used for low frequency applications. The basics of *TMeasy* and the extra characteristics that have been used in this work are reminded hereafter.

Geometric contact point – To calculate the geometric point of contact $\mathbf{P}$, it is first necessary to assess the contact normal for each tire. The collision detection algorithm that calculates the contact normals $\mathbf{e_n}$ is presented in section 3.2.2.

A scheme of the wheel with the points and vectors used for the tire modeling as well as the triangular face in contact with the tire, are shown in fig. 3.20. The unit vector $\mathbf{e_{yR}}$ defines the wheel center plane and is orientated positively regarding the rotation of the wheel. On the one hand, the unit vector $\mathbf{e_x}$, which is defined as the intersection between the wheel center plane and the stepped triangular face, gives the direction of the longitudinal tire force. On the other hand, the unit vector $\mathbf{e_y}$ gives the direction of the lateral tire forces and is defined as being perpendicular to the contact normal $\mathbf{e_n}$ and the unit vector $\mathbf{e_x}$. The unit vector $\mathbf{e_{zR}}$ is obtained from the scalar product of $\mathbf{e_x}$ and $\mathbf{e_{yR}}$, as can be seen in

$$\mathbf{e_{zR}} = \mathbf{e_x} \wedge \mathbf{e_{yR}} \tag{3.147}$$

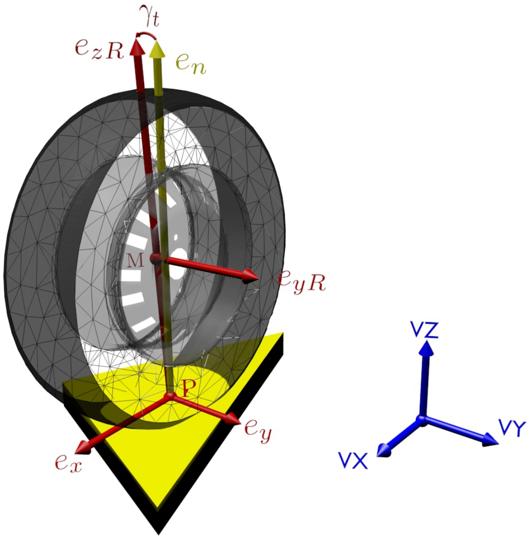


Figure 3.20: Points and vectors for the tire model

The geometric contact point **P** is located at the shortest distance from the wheel center **M** to the intersection between the wheel plane and the triangular face. The tire camber angle is defined as shown in eq. (3.148).

$$\gamma^* = \arcsin(\mathbf{e_{yR}}^{\mathrm{T}}\mathbf{e_n}) \tag{3.148}$$

When the tire has a camber angle, the geometric contact point **P** does not coincide with the effective point of the resulting vertical tire force. However, here, as the camber angle of the wheels of the XBW vehicle prototype is almost zero and as the maneuvers are performed at low speed thus avoiding the camber angle due to suspension kinematics, the effective point has been assumed to be the geometric contact point.

Velocity of the contact point – The velocity of the contact point $\dot{\mathbf{P}}$ is a basic magnitude for the calculation of the tire forces. This velocity is the one of the wheel center $\dot{\mathbf{M}}$ projected onto the contact triangular face, or said in another way, it is the velocity of the wheel center without the component oriented in the direction of the contact normal $\mathbf{e_n}$. As the four wheel centers are points of the vehicle MB model ($p_{17}, p_{27}, p_{35}, p_{45}$), their velocities are easily calculated. Then, the projection onto the contact triangular face is shown in eq. (3.149).

$$\dot{\mathbf{P}} = \dot{\mathbf{M}} - \left(\dot{\mathbf{M}} \cdot \mathbf{e_n}\right)\mathbf{e_n} \tag{3.149}$$

Finally, the velocity of the contact point can be projected on the longitudinal unit vector $\mathbf{e_x}$ and the lateral unit vector $\mathbf{e_y}$ as demonstrated in eqs. (3.150) and (3.151).

$$\nu_{\mathrm{x}} = \dot{\mathbf{P}} \cdot \mathbf{e_x} \tag{3.150}$$

$$\nu_{\mathrm{y}} = \dot{\mathbf{P}} \cdot \mathbf{e_y} \tag{3.151}$$

Wheel vertical load – The vertical tire force F_z has been calculated as a function of the normal tire deflection Δz and its velocity $\Delta\dot{z}$, as shown in

$$F_z = a\Delta z + d_R\Delta\dot{z} \tag{3.152}$$

where a and d_R are constants.

Generalized tire force – The longitudinal and lateral slips are demonstrated in eqs. (3.153) and (3.154).

$$s_x = \frac{-(\nu_x - r_D\Omega)}{r_D|\Omega|\hat{s}_x + \nu_n} \tag{3.153}$$

$$s_y = \frac{-\nu_y}{r_D|\Omega|\hat{s}_y + \nu_n} \tag{3.154}$$

where r_D is the dynamic rolling radius of the tire, Ω is the angular velocity of the wheel, ν_n is a small fictitious velocity and finally $\hat{s}_x$ and $\hat{s}_y$ are two coefficients that allow to give more weight to the longitudinal or to the lateral slips for the calculation of the generalized slip presented hereafter. The small fictitious velocity ν_n is necessary to cover situations in which the wheel locks. Indeed, in this particular situation, $r_D|\Omega| = 0$ and, without ν_n the slips ${s_x}^N$ and ${s_y}^N$ would tend to infinity, which is not acceptable from a numerical point of view. When choosing small values of ν_n ($\nu_n > 0$), the singularity is avoided, and the generalized slip points in the direction of the sliding velocity when the wheel is locked. Lastly, in normal driving conditions, ν_n does not affect the tire model behavior as $r_D|\Omega| >> \nu_n$. In this research, $\nu_n = 10^{-7}$ has been taken.

Both slips can be combined to get the generalized slip, as shown in eq. (3.155). Up to this point of the description of the tire model, no characteristic curves that relate the slip to the tire forces have been assumed. Several approximations can be considered, like for instance the one proposed by the authors of this tire model (Hirschberg et al., 2007) or those of the *Magic Formula*. For the sake of simplicity, in this work, a simple linearized model, shown in fig. 3.21, has been chosen. In this figure μ is the road–tire friction coefficient, F is the generalized tire force, F_z is the vertical tire force and s_c is the critical slip. First, this choice has been motivated by the fact that the tire behavior for low speed maneuvers is well approximated by means of a linear function. Then, as the tire parameters have been approximated thanks to similar tires, precise approximations of the tire characteristics would not be justified. The longitudinal and lateral tire forces are finally obtained from

$$s_g = \sqrt{(s_x)^2 + (s_y)^2} \tag{3.155}$$

$$F_x = F\frac{s_x}{s_g} \tag{3.156}$$

$$F_y = F\frac{s_y}{s_g} \tag{3.157}$$

First order tire dynamics – As mentioned before, in this research, the first order tire dynamics are vital to model properly the tire behavior when the vehicle stands still, starts or stops on an inclined test track or when the wheels lock. The forces acting in the contact patch deflect longitudinally and laterally the tire as shown in fig. 3.23. According to the *TMeasy* model, the tire forces F_x and F_y can be reasonably represented by the first order

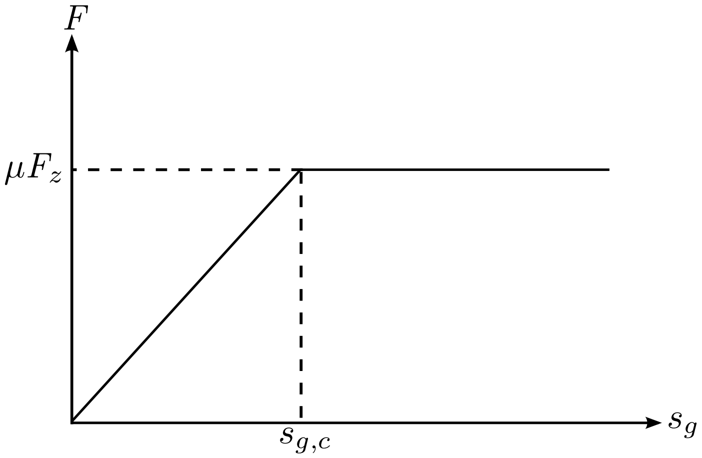


Figure 3.21: Approximations of the generalized tire characteristics

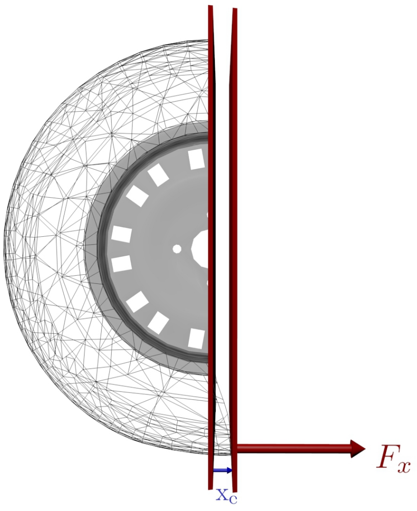


Figure 3.22: Longitudinal tire deflection due to the contact forces

approximations shown in eqs. (3.158) and (3.159).

$$\underbrace{F_x(\nu_\mathrm{x} + \dot{\mathrm{x}}_\mathrm{e})}_{F_x^D} \simeq F_\mathrm{x}(\nu_\mathrm{x}) + \frac{\partial F_\mathrm{x}}{\partial \nu_\mathrm{x}}\left[(\nu_\mathrm{x} + \dot{\mathrm{x}}_\mathrm{e}) + \nu_\mathrm{x}\right] = \underbrace{F_x(\nu_\mathrm{x})}_{F_x^S} + \frac{\partial F_\mathrm{x}}{\partial \nu_\mathrm{x}}\dot{\mathrm{x}}_\mathrm{e} \tag{3.158}$$

$$\underbrace{F_y(\nu_\mathrm{y} + \dot{\mathrm{y}}_\mathrm{e})}_{F_y^D} \simeq F_\mathrm{y}(\nu_\mathrm{y}) + \frac{\partial F_\mathrm{y}}{\partial \nu_\mathrm{y}}\left[(\nu_\mathrm{y} + \dot{\mathrm{y}}_\mathrm{e}) + \nu_\mathrm{y}\right] = \underbrace{F_y(\nu_\mathrm{y})}_{F_y^S} + \frac{\partial F_\mathrm{y}}{\partial \nu_\mathrm{y}}\dot{\mathrm{y}}_\mathrm{e} \tag{3.159}$$

where $\dot{\mathrm{x}}_\mathrm{e}$ and $\dot{\mathrm{y}}_\mathrm{e}$ are the longitudinal and lateral tire deflections, F_x^D and F_y^D are the dynamic tire forces and, F_x^S and F_y^S are the steady–state tire forces. In steady–state, that is when $\dot{\mathrm{x}}_\mathrm{e} = 0$ and $\dot{\mathrm{y}}_\mathrm{e} = 0$, the tire forces F_x^S and F_y^S are given by eqs. (3.156) and (3.157) respectively.

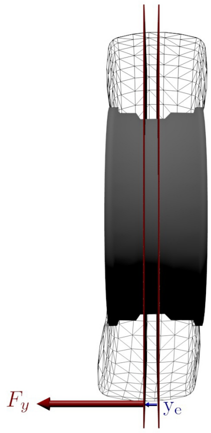


Figure 3.23: Lateral tire deflection due to the contact forces

Their derivatives with respect to the contact point velocity are given in

$$\frac{\partial F^{\mathrm{S}}_{\mathrm{x}}}{\partial \nu_{\mathrm{x}}} = \frac{\partial F^{\mathrm{S}}_{\mathrm{x}}}{\partial \mathrm{s_x}} \frac{\partial \mathrm{s_x}}{\partial \nu_{\mathrm{x}}} = \frac{\partial F^{\mathrm{S}}_{\mathrm{x}}}{\partial \mathrm{s_x}} \frac{-1}{\mathrm{r_D}|\Omega|\hat{\mathrm{s}}_{\mathrm{x}} + \nu_n} \tag{3.160}$$

$$\frac{\partial F^{\mathrm{S}}_{\mathrm{x}}}{\partial \nu_{\mathrm{x}}} = \frac{\partial F^{\mathrm{S}}_{\mathrm{y}}}{\partial \mathrm{s_y}} \frac{\partial \mathrm{s_y}}{\partial \nu_{\mathrm{y}}} = \frac{\partial F^{\mathrm{S}}_{\mathrm{y}}}{\partial \mathrm{s_y}} \frac{-1}{\mathrm{r_D}|\Omega|\hat{\mathrm{s}}_{\mathrm{y}} + \nu_n} \tag{3.161}$$

Subsequently, eqs. (3.160) and (3.161) can be substituted into eqs. (3.158) and (3.159), as shown in eqs. (3.162) and (3.163).

$$F^{\mathrm{D}}_{\mathrm{x}} \simeq \frac{F}{\mathrm{s_g}}\mathrm{s_x} + \frac{\partial F^{\mathrm{S}}_{\mathrm{x}}}{\partial \mathrm{s_x}} \frac{-1}{\mathrm{r_D}|\Omega|\hat{\mathrm{s}}_{\mathrm{x}} + \nu_n} \dot{\mathrm{x}}_{\mathrm{e}} = \frac{F}{\mathrm{s_g}} \frac{-(\nu_{\mathrm{x}} - \mathrm{r_D}\Omega)}{\nu^*_{\mathrm{x}}} + \frac{\partial F^{\mathrm{S}}_{\mathrm{x}}}{\partial \mathrm{s_x}} \frac{-1}{\nu^*_{\mathrm{x}}} \dot{\mathrm{x}}_{\mathrm{e}} \tag{3.162}$$

$$F^{\mathrm{D}}_{\mathrm{y}} \simeq \frac{F}{\mathrm{s_g}}\mathrm{s_y} + \frac{\partial F^{\mathrm{S}}_{\mathrm{y}}}{\partial \mathrm{s_y}} \frac{-1}{\mathrm{r_D}|\Omega|\hat{\mathrm{s}}_{\mathrm{y}} + \nu_n} \dot{\mathrm{y}}_{\mathrm{e}} = \frac{F}{\mathrm{s_g}}\mathrm{s_y} + \frac{\partial F^{\mathrm{S}}_{\mathrm{y}}}{\partial \mathrm{s_y}} \frac{-1}{\nu^*_{\mathrm{y}}} \dot{\mathrm{y}}_{\mathrm{e}} \tag{3.163}$$

where $\nu^*_{\mathrm{x}} = \mathrm{r_D}|\Omega|\hat{\mathrm{s}}_{\mathrm{x}} + \nu_n$ and $\nu^*_{\mathrm{y}} = \mathrm{r_D}|\Omega|\hat{\mathrm{s}}_{\mathrm{y}} + \nu_n$. The tire dynamic forces can also be calculated by considering that the tire consists of two spring–damper elements, one in the longitudinal direction and the other in the lateral direction. Both equations are presented in eqs. (3.164) and (3.165).

$$F^{\mathrm{D}}_{\mathrm{x}} \simeq \mathrm{c_x x_e} + \mathrm{d_x}\dot{\mathrm{x}}_{\mathrm{e}} \tag{3.164}$$

$$F^{\mathrm{D}}_{\mathrm{y}} \simeq \mathrm{c_y y_e} + \mathrm{d_y}\dot{\mathrm{y}}_{\mathrm{e}} \tag{3.165}$$

where $\mathrm{c_x}$, $\mathrm{c_y}$, $\mathrm{d_x}$, $\mathrm{d_y}$ are the stiffness and damping parameters for the longitudinal and lateral tire deflections. After that, eq. (3.162) can be combined with eq. (3.164) to yield a first order differential equation, shown in eq. (3.166) for the longitudinal deflection of the tire. The first order differential equation for the lateral deflection, shown in eq. (3.167) is obtained using eqs. (3.163) and (3.165).

$$\left(\nu^*_{\mathrm{x}}\mathrm{d_x} + \frac{\partial F^{\mathrm{S}}_{\mathrm{x}}}{\partial \mathrm{s_x}}\right) \dot{\mathrm{x}}_{\mathrm{e}} = -\frac{F}{\mathrm{s_g}}(\nu_{\mathrm{x}} - \mathrm{r_D}\Omega) - \nu^*_{\mathrm{x}}\mathrm{c_x x_e} \tag{3.166}$$

$$\left(\nu^*_{\mathrm{y}}\mathrm{d_y} + \frac{\partial F^{\mathrm{S}}_{\mathrm{y}}}{\partial \mathrm{s_y}}\right) \dot{\mathrm{y}}_{\mathrm{e}} = -\frac{F}{\mathrm{s_g}}\nu_{\mathrm{y}} - \nu^*_{\mathrm{y}}\mathrm{c_y y_e} \tag{3.167}$$

Both differential equations have to be integrated at each time step to calculate the tire forces by means of eqs. (3.164) and (3.165). To this end, in this research, both differential equations are solved for every iteration (corresponding to a time step) of the Newton–Raphson method shown in eq. (3.15). As the variables of eqs. (3.166) and (3.167) are constant during each iteration, both equations become first order differential equations with constant coefficients. Their analytical solutions are presented in eqs. (3.168) and (3.169).

$$\begin{gathered} x_e = \frac{a_2}{a_1} + a_3\, e^{-a_1 \Delta t_i} \\ a_1 = \frac{\nu_x^* c_x}{\left(\nu_x^* d_x + \dfrac{\partial F_x}{\partial s_x}\right)} \qquad a_2 = -\frac{{}^{F}\!/\!_{s_g}(\nu_x - r_D \Omega)}{\left(\nu_x^* d_x + \dfrac{\partial F_x}{\partial s_x}\right)} \qquad a_3 = x_{e0} - \frac{a_2}{a_1} \end{gathered} \tag{3.168}$$

$$\begin{gathered} y_e = \frac{b_2}{b_1} + b_3\, e^{-b_1 \Delta t_i} \\ b_1 = \frac{\nu_y^* c_y}{\left(\nu_y^* d_y + \dfrac{\partial F_y}{\partial s_y}\right)} \qquad b_2 = -\frac{{}^{F}\!/\!_{s_g}\, \nu_y}{\left(\nu_y^* d_y + \dfrac{\partial F_y}{\partial s_y}\right)} \qquad b_3 = y_{e0} - \frac{b_2}{b_1} \end{gathered} \tag{3.169}$$

where x_{e0} and y_{e0} are the initial longitudinal and lateral tire deflection for each integration time step. Consequently, for each iteration of an integration time step, x_e and y_e are calculated using eqs. (3.168) and (3.169), and $\dot{x}_e$ and $\dot{y}_e$ are obtained by means of eqs. (3.166) and (3.167). When the integration time step changes, x_{e0} and y_{e0} must be updated with the latest values of x_e and y_e. Solving the differential equations analytically at each iteration allows calculating efficiently and accurately the tire forces.

Locked wheel with tire sliding – A small modification has to be made to the previous equations to allow the tire model to take into account situations in which the wheel is locked but the tire is sliding. Although this situation is not common in normal driving conditions, it occurs much more frequently in simulation. For example, at the beginning of a simulation, an initial location and orientation are given to the vehicle and then, it is dropped on the test track. Subsequently, the wheel rotation velocities are zero and, longitudinal and lateral slips appear due to the suspension deflections and the test track inclination. When a tire is sliding, for example laterally, y_e increases until reaching a maximum value which should be maintain during all the sliding.

In such a situation, as $\Omega = 0$, eq. (3.169) becomes eq. (3.170). It can be seen from eq. (3.170) that $e^{-b_1 \Delta t_i}$ should be equal to one to get $y_e = y_{e0}$. This means that the tire lateral deflection during the iterations of the Newton–Raphson method y_e is equal to the tire lateral deflection at the beginning of the corresponding integration time step y_{e0}, and so on for the next integration time steps until the tire stops to slide. $e^{-b_1 \Delta t_i} \simeq 1$ implies that $-b_1 \Delta t_i \simeq 0$. As $\left.\dfrac{\partial F_y}{\partial s_y}\right|_{s_g > s_c} = 0$, it follows that $b_1 = \dfrac{c_y}{d_y}$. Unfortunately, after having substituted c_y, d_y and Δt_i, $-b_1 \Delta t_i \neq 0$. Therefore, eqs. (3.168) and (3.169) have to be modified in order to maintain y_e constant during the sliding. In the *TMeasy* model, it has been chosen to change the derivative as shown in eq. (3.171). In this way, when $s_g < s_c$, the correct derivative is well approximated (if linear tire curves are used, the derivative is exact) and, when $s_g > s_c$, the derivative is largely greater than zero, thus making $-b_1 \Delta t_i \simeq 0$. The same modification has to be made to the derivative of the longitudinal tire forces.

$$\begin{aligned}
&\mathrm{y_e} = \frac{\mathrm{b_2}}{\mathrm{b_1}} + \left(\mathrm{y_{e0}} - \frac{\mathrm{b_2}}{\mathrm{b_1}}\right) \mathrm{e}^{-\mathrm{b_1}\Delta \mathrm{t_i}} \\
&\mathrm{b_1} = \frac{\nu_n \mathrm{c_y}}{\left(\nu_n \mathrm{d_y} + \left.\dfrac{\partial F_y}{\partial \mathrm{s_y}}\right|_{\mathrm{s_g} > \mathrm{s_c}}\right)} \qquad \mathrm{b_2} = -\frac{F/_{\mathrm{s_g}}\, \nu_\mathrm{y}}{\left(\nu_n \mathrm{d_y} + \left.\dfrac{\partial F_y}{\partial \mathrm{s_y}}\right|_{\mathrm{s_g} > \mathrm{s_c}}\right)}
\end{aligned} \tag{3.170}$$

$$\frac{\partial F_\mathrm{y}}{\partial \mathrm{s_y}} \to \frac{F_\mathrm{y}}{\mathrm{s_y}} = F_\mathrm{y} \frac{1}{\mathrm{s_y}} = \frac{F}{\mathrm{s_g}} \mathrm{s_y} \frac{1}{\mathrm{s_y}} = \frac{F}{\mathrm{s_g}} \tag{3.171}$$

Transition to stand–still – The *TMeasy* model is able to handle situations where the vehicle stands still. In such situations, $\nu_\mathrm{x} = 0$, $\nu_\mathrm{y} = 0$, $\Omega = 0$ and $\mathrm{s_g} = 0\cdot$ Subsequently, eqs. (3.166) to (3.169) become eqs. (3.172) to (3.175) respectively.

$$\left(\nu_n \mathrm{d_x} + \left.\frac{\partial F_\mathrm{x}^\mathrm{S}}{\partial \mathrm{s_x}}\right|_{\mathrm{s_g}=0}\right) \dot{\mathrm{x}}_\mathrm{e} = -\nu_n \mathrm{c_x} \mathrm{x_e} \tag{3.172}$$

$$\left(\nu_n \mathrm{d_y} + \left.\frac{\partial F_\mathrm{y}^\mathrm{S}}{\partial \mathrm{s_y}}\right|_{\mathrm{s_g}=0}\right) \dot{\mathrm{y}}_\mathrm{e} = -\nu_n \mathrm{c_x} \mathrm{y_e} \tag{3.173}$$

$$\begin{aligned}
&\mathrm{x_e} = \mathrm{a_3}\, \mathrm{e}^{-\mathrm{a_1}\Delta \mathrm{t_i}} \\
&\mathrm{a_1} = \frac{\nu_n \mathrm{c_x}}{\left(\nu_n \mathrm{d_x} + \left.\dfrac{\partial F_x}{\partial \mathrm{s_x}}\right|_{\mathrm{s_g}=0}\right)} \qquad \mathrm{a_3} = \mathrm{x_{e0}}
\end{aligned} \tag{3.174}$$

$$\begin{aligned}
&\mathrm{y_e} = \mathrm{b_3}\, \mathrm{e}^{-\mathrm{b_1}\Delta \mathrm{t_i}} \\
&\mathrm{b_1} = \frac{\nu_n \mathrm{c_y}}{\left(\nu_\mathrm{y}^* \mathrm{d_y} + \left.\dfrac{\partial F_y}{\partial \mathrm{s_y}}\right|_{\mathrm{s_g}=0}\right)} \qquad \mathrm{b_3} = \mathrm{y_{e0}}
\end{aligned} \tag{3.175}$$

At stand–still, $\mathrm{x_e}$ and $\mathrm{y_e}$ will decay in time according to eqs. (3.174) and (3.175). This implies that F_x and F_y are never completely constant but small values of ν_n yield large time constants. In this way, the tire model considers the stick–slip phenomena.

Brake model

Similarly to the tire model, the brake model must be able to manage situations where the vehicle stands still, starts or stops on an inclined test track. To this end, the tangential force model developed in (Dopico et al., 2011) has been used since it takes into account stiction and sliding. The parameters of the model have been obtained from experimental data.

3.2 Simulation environment

A self-developed driving simulator has been setup to make the use of the vehicle model easier. Indeed, the vehicle MB model is programmed in `Fortan` and has no convenient graphical output. For that purpose, an open–source 3D graphics toolkit, *OpenSceneGraph*, has been used to provide the simulator with realistic 3D graphics. As a consequence, the simulation code has two different parts: one in `Fortan` containing the vehicle MB model and another in `C++` that covers the 3D outputs, the collision detection and in a near future the communication

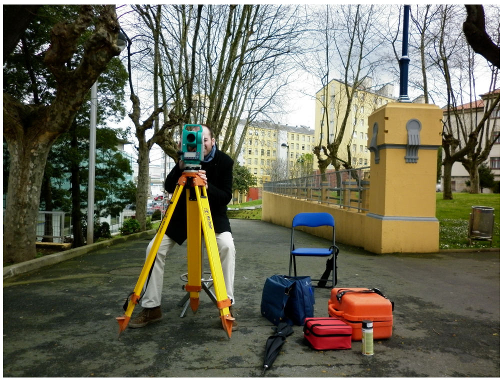


Figure 3.24: Topographical survey with the total station

with the DAS. Hereafter the road profile, the collision detection algorithm and the graphical environment are presented.

3.2.1 Road profile

In order to properly validate any vehicle model through field testing, a true road profile is vital for the simulation. In this research, the test track is an almost flat area of the campus of the engineering school. The information about the road profile has been obtained through a topographical survey of the test track using a total station (model *SET530R* from *Sokkia*). Figure 3.24 shows a photo of the use of the total station for the topographical survey. About three hundred points spaced out, on average, every 1.5 m have been collected. The 3D scattered points are shown in fig. 3.25 where the vertical scale has been magnified in order to visualize better the surface. These points have then been interpolated using the natural neighbor interpolation method and regular spacing, as can be seen in fig. 3.26. A drawback inherent to this method is that the convex hull of the 3D scattered points is used by the interpolation method. As a result, undesirable new points that lie outside the limits of the actual test track are created. To solve this problem, the projections of the 3D scattered points on the horizontal plane are used to identify the points that define the contour of the 2D point set. An 2D alpha shape algorithm has been employed to detect this contour as shown in fig. 3.27. As the 3D point set is an almost flat surface, the points that lie in its contour will be the same as the ones of the 2D contour. Once the contour has been defined, a Delaunay triangulation constrained into the alpha shape is performed on the interpolated and regularly spaced points. The result is similar to the one using only the interpolation method except that now undesirable points outside the test track are no more generated. Finally the points and triangles set is saved in *Wavefront* format in order to be passed to the simulation environment. The triangle mesh of the test track surface is shown in fig. 3.28.

3.2.2 Collision detection

The collision detection strategy and algorithms employed in this work are based on the ones developed in the LIM for the real–time simulation of an excavator (Dopico et al., 2011). These methods have been specially tailored for real–time simulations where the MB models have

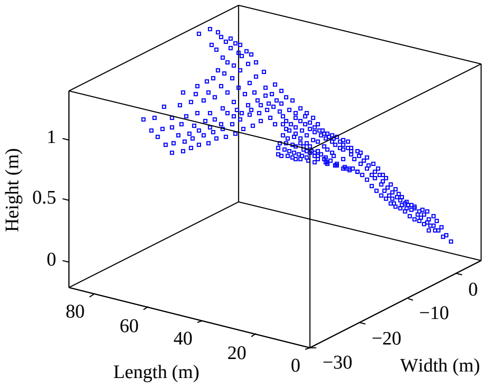


Figure 3.25: 3D scattered points collected during the topographical survey

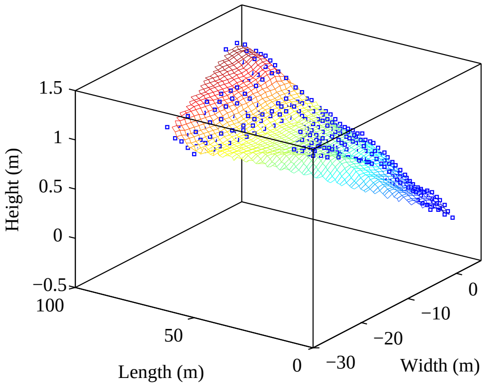


Figure 3.26: Interpolation of the 3D scattered points

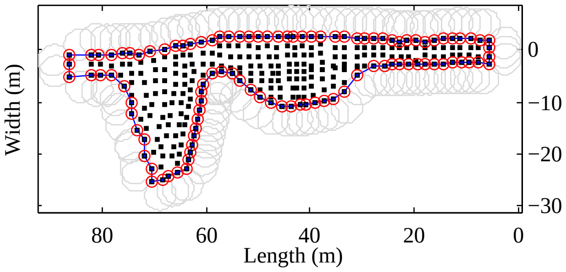


Figure 3.27: Contour detection using the alpha shape algorithm

to interact potentially with a high number of bodies. To this end, the geometry of complex CAD environment and bodies (like the test track) has been approximated by meshes of triangular faces and the geometry of the bodies of the MB systems (like the wheels) by spheres of different sizes. Figure 3.29 shows the four identical spheres that have been employed to approximate the geometry of the tires. It can be noted that if the vehicle would laterally enter in contact with some body, the approximation of the tire geometry would not be valid.

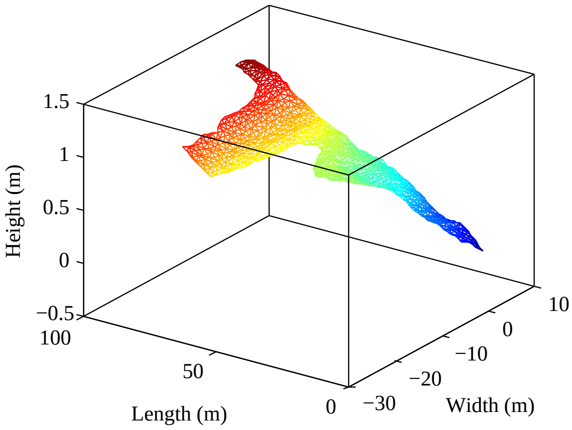


Figure 3.28: 3D model of the test track

However, as this situation can never occur in this work, it has not been necessary to further detailed this geometry. Then, each triangular face of a CAD environment or body has its own normal vector and its own properties of stiffness and friction and each sphere is characterized also depending on the material properties and curvature of the MB model.

After having defined the contact geometry of the different bodies, the detection of the triangular faces that are in contact with the primitive spheres has to be carried out. The most likely contact scenarios are presented in (Dopico et al., 2011) for spheres, yielding to the normal vector of the contact and, in the same paper, the normal forces are calculated. This vector is then used by the tire subroutine to compute the tire forces and moments. It is worth mentioning that as the test track shown in fig. 3.28 is a smooth and almost flat surface, only one contact point exist for each tire. At each time step, the contacts taking place have to be determined efficiently. This involves checking all the spheres against all the faces and their edges leading to intensive computational loads at each time step for realistic environments. In order to speed up this process, the collision detection algorithm uses an octree-based hierarchical decomposition of the entire scene mesh whose principal idea is to generate a tree-based hierarchical structure that is used to quickly reject the polygons not involved in potential collisions, in order to reduce the number of polygons tested against contact with the primitive objects that represent the geometry of the models.

3.2.3 Graphical environment

The 3D graphical environment is intended to reproduce the real environment where the test maneuvers have been realized, as well as the vehicle itself. First, a 2D map in *AutoCAD* of the campus of the engineering school has been used to locate the buildings and the roads. After that, the campus buildings have been extruded outwards in their right places using an open–source 3D content creation software, *Blender*. The road profile obtained in section 3.2.1 has been merged into the 3D objects of the roads. Textures have been applied to roads, buildings, sidewalks, etc, using photos taken on the campus. Lastly, a sky dome and a sun have been added to reproduce the sky and to give lighting to the whole scene. A general view of the 3D objects of the campus is shown in fig. 3.30. Figures 3.31 and 3.32 show the real

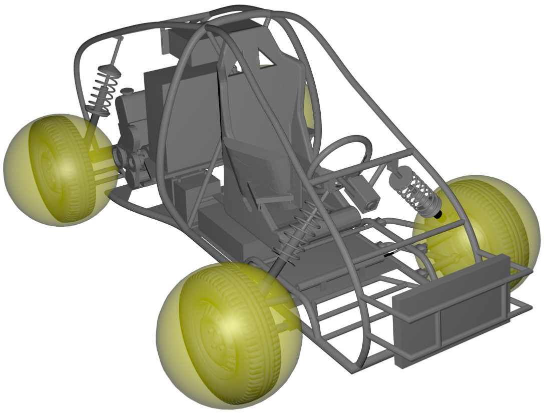


Figure 3.29: Spheres used for the collision detection of the tires

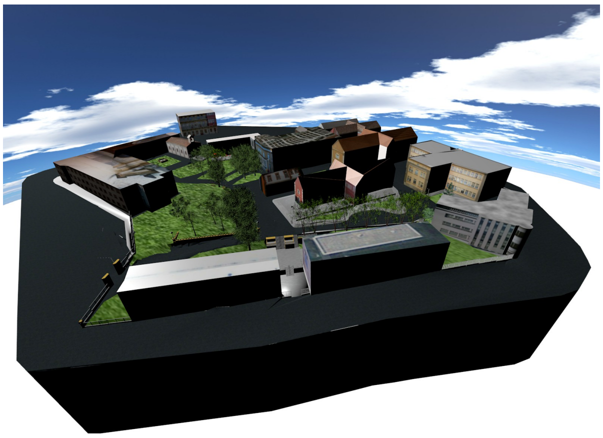


Figure 3.30: 3D model of the campus with the skydome in the background

and virtual surroundings of the test track.

The geometry of the 3D objects for the vehicle have been obtained from the CAD objects, which have been later textured. In order to animate these objects, the vehicle MB model passes to the *OpenSceneGraph* library the position and orientation of each body at each diplay time step.

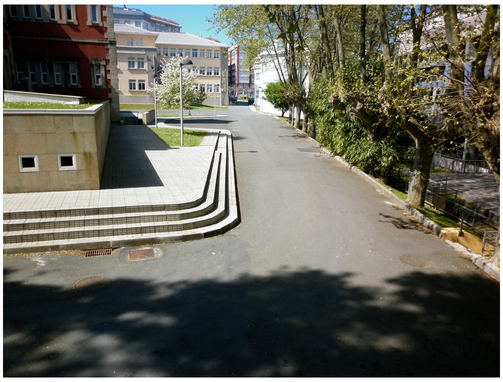


Figure 3.31: Photo of the test track

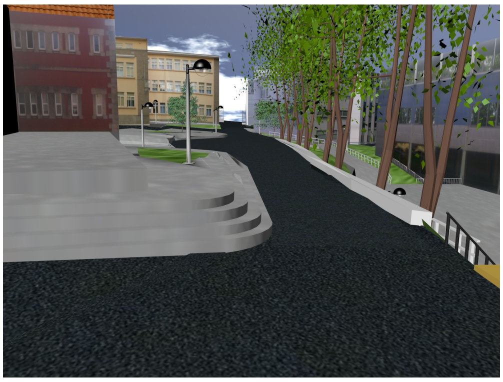


Figure 3.32: 3D surroundings of the test track

Appendix B

Works derived from this thesis

Part of this thesis has been financed by the Spanish Government through the BES-2013-063598 fellowship. Several works have arisen from this thesis, including journal papers, conference communications, and a software library released as open source, called MBDE[1] . Moreover, other journal papers have been submitted, but they are undergoing the revision process as of the time of writing. The list of publications is exposed hereafter.

Journal papers

Roland Pastorino, Emilio Sanjurjo, Alberto Luaces, Miguel Á. Naya, Wim Desmet, Javier Cuadrado. Validation of a Real-Time Multibody Model for an X-by-Wire Vehicle Prototype Through Field Testing. Journal of Computational and Nonlinear Dynamics, 10(3):031006, 2015.

José L. Torres-Moreno, José L. Blanco-Claraco, Antonio Giménez-Fernández, Emilio Sanjurjo, Miguel Á. Naya. Online Kinematic and Dynamic-State Estimation for Constrained Multibody Systems Based on IMUs. Sensors, 16(3):333, 2016.

Submitted Journal papers

Emilio Sanjurjo, Miguel Á. Naya, Javier Cuadrado, Arend Schwab. Roll Angle Estimator Based on Angular Rate Measurement for Single Track Vehicles. Vehicle System Dynamics (under review).

Emilio Sanjurjo, Miguel Á. Naya, José Luis Blanco-Claraco, José Luis Torres-Moreno, Antonio Giménez-Fernandez. Multibody-Based State Observers: a Review and Benchmark (under review).

Conference communications

Emilio Sanjurjo, Roland Pastorino, Daniel Dopico, Miguel Á. Naya. Validación Experimental de un Modelo Multicuerpo de un Prototipo de Vehículo Automatizado, in XIX Congreso Nacional de Ingeniería Mecánica, Castellón, Spain, 2012.

[1] See https://github.com/MBDS/mbde-matlab

Emilio Sanjurjo, Roland Pastorino, Pasquale Gallo, Miguel Á. Naya. Implementation Issues of an On Board Real-Time Multibody Model, in 3rd Joint Int. Conference on Multibody System Dynamics (IMSD 2014) and 7th Asian Conference on Multibody Dynamics (ACMD 2014), Busan, Korea, 2014.

José L. Torres-Moreno, José L. Blanco-Claraco, Emilio Sanjurjo, Miguel Á. Naya, Antonio Giménez-Fernández. Towards Benchmarking of State Estimators for Multibody Dynamics, in 3rd Joint Int. Conference on Multibody System Dynamics (IMSD 2014) and 7th Asian Conference on Multibody Dynamics (ACMD 2014), Busan, Korea, 2014.

Emilio Sanjurjo, Alfonso Varela, Miguel Á. Naya. Modelización de los Subsistemas de Tracción y Frenado de un Automóvil para su Uso en un Simulador, in XX Congreso Nacional de Ingeniería Mecánica, Málaga, Spain, 2014.

Emilio Sanjurjo, José L. Blanco-Claraco, José L. Torres-Moreno, Miguel Á. Naya. Testing the Efficiency and Accuracy of Multibody-Based State Observers, in ECCOMAS Thematic Conference on Multibody Dynamics 2015, Barcelona, Spain, 2015.

José L. Torres-Moreno, José L. Blanco-Claraco, Emilio Sanjurjo, Antonio Giménez-Fernández, Miguel Á. Naya. A Testbed for Benchmarking State Observers in Multibody Dynamics, in ECCOMAS Thematic Conference on Multibody Dynamics 2015, Barcelona, Spain, 2015.

Emilio Sanjurjo, Edoardo Sinigaglia, Miguel Á. Naya. Multibody-based State Observer for Navigation Applications, in 4th Joint Int. Conference on Multibody System Dynamics (IMSD 2016), Montreal, Canada, 2016.

Emilio Sanjurjo, Edoardo Sinigaglia, Miguel Á. Naya. Observador de Estados Basado en Modelo Multicuerpo Aplicado a un Vehículo Automóvil, in XXI Congreso Nacional de Ingeniería Mecánica, Elche, Spain, 2016 (to be presented).

Appendix C

Resumen extendido

Introducción

Las simulaciones multicuerpo son extensivamente empleadas en la industria para reducir el tiempo de desarrollo y el coste de nuevos productos. Aunque inicialmente las simulaciones se empleaban simplemente como herramientas de análisis, el aumento de la capacidad computacional disponible y las formulaciones multicuerpo eficientes han permitido simular mecanismos de cierta complejidad más rápido que el tiempo que tardan en hacer una cierta maniobra en realidad. Esta nueva capacidad ha extendido el uso de simulaciones multicuerpo a aplicaciones de tiempo real, en las que las simulaciones interactúan con elementos reales. Cuando las simulaciones interactúan con personas, como en el caso de simuladores de conducción o de entrenamiento de operarios, se conoce como "human-in-the-loop". Cuando son otros elementos los que interactúan con la simulación, como por ejemplo controles de estabilidad para vehículos en los que el vehículo es simulado, se habla de "hardware-in-the-loop".

Desde sus inicios, el Laboratorio de Ingeniería Mecánica de la Universidad de A Coruña ha centrado su investigación en métodos de simulación multicuerpo eficientes. Una de las líneas de investigación seguidas en este laboratorio desde el año 2008 ha sido el empleo de simulaciones multicuerpo en observadores de estados.

Los observadores de estados son herramientas matemáticas que permiten obtener información de un sistema cuando no es posible instalar un sensor que provea de dicha información, ya sea por motivos técnicos o económicos. Los observadores de estados emplean un modelo dinámico del sistema del que se quiere conocer el estado con información de sensores disponibles en el sistema en estudio. De esta manera puede obtener más y/o mejor información acerca del sistema que con ambos elementos por separado. Uno de los algoritmos más conocidos y empleados para diseñar observadores de estados es el filtro de Kalman, y sus variantes para sistemas no lineales. Cada paso de tiempo del filtro de Kalman tiene dos etapas. En la primera, llamada predicción, el estado evoluciona de acuerdo con el modelo dinámico empleado en el filtro. En la segunda etapa, llamada corrección, se incorpora la información proporcionada por los sensores para mejorar la predicción.

Aunque la combinación de modelos multicuerpo con observadores es deseable, el éxito obtenido hasta ahora ha sido limitado. El motivo es que la estructura matemática de los observadores de estados y la de las simulaciones multicuerpo suelen ser muy distintas. Mientras que el filtro de Kalman fue formulado para sistemas lineales de primer orden y con variables independientes, las simulaciones multicuerpo son problemas de segundo orden, no lineales, y que, en general, tienen restricciones. Por eso surgen distintas posibilidades que se estudian en esta tesis.

Los observadores de estados son ampliamente utilizados en el sector de la automoción, ya que se emplean para medir a bajo coste magnitudes como el ángulo de deslizamiento lateral del vehículo, que es una magnitud crítica para controlar la estabilidad del vehículo. Por ese motivo, en esta tesis se desarrollan observadores de estados que luego se aplican al campo de la automoción.

Para la realización de esta tesis, el Laboratorio de Ingeniería Mecánica dispone de un vehículo instrumentado. El modelo multicuerpo de dicho vehículo también estaba en un estado avanzado al comienzo de esta tesis, aunque ha tenido que ser modificado para mejorar tanto su comportamiento como su eficiencia computacional.

Objetivos

El principal objetivo de esta tesis es obtener un método que permita obtener un observador de estados basado en una simulación multicuerpo de un vehículo, y que se pueda ejecutar a bordo de dicho vehículo. Para conseguir este objetivo, se han fijado una serie de metas parciales:

- Desarrollar diferentes estrategias para combinar modelos multicuerpo y filtros de Kalman. Estas estrategias se prueban con mecanismos sencillos para entender su funcionamiento, y para evaluar el aumento del coste computacional al aumentar el tamaño del sistema.

- Mejorar el modelo multicuerpo ya disponible del vehículo, reduciendo su coste computacional y mejorando su precisión.

- Instalar el modelo multicuerpo a bordo del vehículo. Para ello hay que desarrollar un programa informático que lea la información de los sensores instalados a bordo del vehículo y ejecute la simulación multicuerpo a bordo del vehículo y en tiempo real.

- Implementar un observador de estados basado en el modelo multicuerpo del vehículo y evaluar su coste computacional y la precisión obtenida.

Estructura de la tesis

El cuerpo principal de esta tesis se organiza de la siguiente manera:

Capítulo 1 El primer capítulo constituye la introducción de la tesis y sirve para situar el trabajo.

Capítulo 2 En este capítulo se hace una revisión bibliográfica de observadores de estados, en general primero, y particularizando para vehículos terrestres después, para finalizar tratando las publicaciones en las que se combinan observadores de estados con modelos multicuerpo.

Capítulo 3 Este capítulo presenta las ecuaciones de los filtros de Kalman continuos y discretos, las formulaciones multicuerpo empleadas en esta tesis, y los observadores de estados basados en modelos multicuerpo que se han evaluado.

Capítulo 4 El cuarto capítulo trata la aplicación de los métodos descritos en el capítulo previo a dos mecanismos, un cuadrilátero y un pentalátero articulados. Se evalúan la precisión obtenida con cada método para tres configuraciones de sensores diferentes en cada mecanismo, y cinco frecuencias de muestreo. También se compara el coste computacional de cada método.

Capítulo 5 En este capítulo se detallan las mejoras realizadas al modelo multicuerpo de automóvil, y la aplicación de un observador de estados a dicho modelo, incluyendo el desarrollo de los modelos de sensores empleados.

Capítulo 6 Este capítulo presenta las conclusiones del trabajo realizado en esta tesis e indica futuras líneas de investigación.

Metodología

El trabajo realizado en esta tesis se puede dividir en dos bloques. El primero consiste en implementar observadores de estados ya existentes en trabajos anteriores y también diseñar nuevos métodos para posteriormente evaluar y comparar sus propiedades. La segunda parte consiste en implementar un observador de estados para el caso concreto de un vehículo automóvil, y para ello se parte de la experiencia adquirida en el primer bloque.

El desarrollo de la primera parte se ha basado completamente en simulación. Para ello se han considerado dos mecanismos, un cuadrilátero articulado, y un pentalátero articulado. Se han seleccionado estos mecanismos porque son relativamente sencillos, a pesar de ser mecanismos de cadena cinemática cerrada. Al tener dos mecanismos con un modelo de distinto tamaño se puede evaluar el incremento de coste computacional al variar el tamaño del modelo. Para cada mecanismo se han realizado dos modelos multicuerpo. El primero se emplea como si fuera un mecanismo real, mientras que el segundo representa un modelo del mecanismo real. Es decir, ambos modelos son muy similares, pero se introducen pequeñas modificaciones que representan errores de modelización.

Posteriormente se modelizan los sensores que se consideran en el mecanismo real. Esto se hace relacionando la señal del sensor con las variables del modelo multicuerpo. Para simular el ruido que tendrían las señales de los sensores si fueran reales, se añade una secuencia de números pseudoaleatorios siguiendo una distribución gaussiana. La secuencia de números pseudoaleatorios empleada ha sido la misma en todos los experimentos realizados para que la comparación entre los distintos métodos sea imparcial.

A continuación se construye el observador de estados. La fase de predicción del filtro de Kalman se realiza con el modelo multicuerpo que tiene errores de modelización, mientras que en la fase de corrección se usan las medidas de los sensores construidas a partir de la simulación que representa al mecanismo real.

Todo el desarrollo realizado en esta parte de la tesis se ha realizado en MATLAB®, y el código creado se distribuye como código libre[1].

Para la realización del segundo bloque de la tesis la primera tarea ha sido mejorar el modelo multicuerpo ya existente del vehículo. Las modificaciones se han centrado en caracterizar mejor la geometría del vehículo, añadir un modelo de resistencia a la rodadura, y mejorar la caracterización del modelo de freno empleado en el modelo multicuerpo. Los parámetros de los modelos de resistencia a la rodadura y del freno se han determinado experimentalmente.

Otra mejora realizada al modelo de vehículo fue la sustitución de coordenadas naturales por coordenadas relativas para realizar el modelo de las ruedas. La modelización con coordenadas naturales funciona bien para el resto del vehículo, pero en el caso de las ruedas, la modelización en coordenadas naturales hace que el proceso de integración del modelo se vuelva inestable cuando se aumenta la velocidad. Esta inestabilidad no se produce con el cambio a coordenadas relativas para las ruedas, reduciéndose además el coste computacional al facilitarse la integración del modelo.

Para la realización del modelo multicuerpo del vehículo se ha empleado la biblioteca de simulación de sistemas multicuerpo MBSLIM[2], desarrollada en el Laboratorio de Ingeniería Mecánica. Este modelo también se ha instalado a bordo del prototipo y se han hecho pruebas ejecutándolo mientras se conduce el vehículo.

A partir de este modelo se ha desarrollado un observador de estados para el vehículo. De todos los métodos considerados en la primera parte de la tesis, el método más adecuado para este problema es el que se ha denominado como errorEKF, ya que ha demostrado una buena precisión y el menor coste computacional de todos los métodos que se han probado en esta tesis. Este método tiene una característica distintiva, ya que en lugar de estimar directamente los estados de interés, lo que se hace es ejecutar la simulación multicuerpo sin modificaciones, y luego se ejecuta el observador de estados, que estima los errores cometidos por la simulación, para a continuación corregirla. Esta estructura hace que este método se pueda implementar sobre simulaciones ya existentes con cambios mínimos en el código, ya que admite cualquier formulación multicuerpo o integrador.

Para la evaluación del observador de estados del vehículo se ha seleccionado una maniobra realizada con el prototipo y se han empleado los datos grabados durante la realización de la misma para ejecutar el observador de estados. Con el mismo set de datos grabados se han realizado dos experimentos: uno de ellos empleando los datos sin modificaciones, y otro en el que se ha añadido ruido a las medidas proporcionadas por el GPS instalado en el vehículo y empleando sólo parte de los datos, para emular los datos de un receptor GPS de menor calidad que el equipo instalado en el vehículo experimental. El primer test sirve para demostrar que el

[1] Véase https://github.com/MBDS/mbde-matlab

[2] Véase http://lim.ii.udc.es/MBSLIM/

algoritmo es estable, mientras que con el segundo se verifica su respuesta en el caso de tener un receptor GPS como el que se podría haber en un vehículo comercial, en el que no sería viable económicamente instalar un equipo de precisión como el del prototipo.

Experimentos y resultados

Nuevamente, en esta sección cabe discutir los resultados de cada bloque descrito en la sección anterior por separado.

En el caso de los mecanismos planos estudiados en esta tesis (cuadrilátero y pentalátero articulados) se han considerado tres configuraciones distintas de sensores en cada uno. En cada configuración se han empleado el mismo número de sensores que grados de libertad tiene el mecanismo en estudio. Las configuraciones de sensores consideradas para el cuadrilátero articulado han sido un codificador angular en la primera barra, un giróscopo en el acoplador, y un giróscopo en la primera barra. De todas estas configuraciones, las dos primeras han ofrecido resultados aceptables con todos los métodos considerados, aunque los resultados del método en tiempo continuo, llamado CEKF, han empeorado más rápido que otros métodos al reducir la frecuencia de muestreo de los sensores, llegando incluso a desestabilizarse el algoritmo para las frecuencias de muestreo más bajas. Con la tercera configuración de los sensores, sin embargo, sólo los métodos basados en el UKF (acrónimo de "unscented Kalman filter") han conseguido obtener resultados aceptables, ya que con los otros métodos el sistema resulta no observable, por lo que los errores de posición iniciales no son recuperables, y los filtros tienden a desestabilizarse con el paso del tiempo.

Las configuraciones de sensores consideradas para el pentalátero articulado han sido equivalentes: codificadores angulares en la primera y última barras, giróscopos en los acopladores, y giróscopos en la primera y última barras. Los resultados obtenidos con este mecanismo han sido cualitativamente muy similares a los anteriores: con las dos primeras configuraciones todos los métodos han funcionado correctamente, con la excepción del CEKF para frecuencias de muestreo bajas. Con la tercera configuración de los sensores, los únicos métodos que han funcionado han sido los de la familia de los UKF, como en el caso anterior.

En cuanto al coste computacional, los métodos UKF han presentado el coste computacional más alto de todos los métodos, y además el tiempo de ejecución aumenta notablemente al pasar del cuadrilátero articulado al pentalátero articulado, tardando aproximadamente el doble en el caso del pentalátero. En el otro extremo está el errorEKF, que ha sido el método más rápido, y el tiempo de computación para el problema del pentalátero se ha incrementado solo un 27 % con respecto al cuadrilátero. Esto hace del errorEKF el método más adecuado para ser implementado en un problema más complejo, como el del vehículo.

Con el vehículo se han realizado dos experimentos. En el primero se han empleado todos los sensores disponibles sin modificaciones, y se ha ejecutado el algoritmo, demostrando su estabilidad, y que comportamiento del modelo multicuerpo resulta corregido para seguir el movimiento de los sensores, pero no hay sensores redundantes con los que validar los resultados. En el segundo experimento los datos del GPS se han modificado para emular los resultados que podrían haber sido obtenidos con un GPS de bajo coste. Para ello se ha reducido la frecuencia de muestreo de 50 Hz de la señal original a 5 Hz, y se ha añadido un ruido gaussiano con desviación típica de 1.786 m en horizontal y 2.551 m en vertical. Este ruido se corresponde con una precisión mejor de 3.5 m el 95 % del tiempo en horizontal, y 5 m en vertical. Empleando la señal del GPS original como referencia para comparar los resultados proporcionados por el observador, el error cuadrático medio obtenido durante la simulación ha sido de 0.562 m. En cuanto al coste computacional, la maniobra estudiada ha durado 55.744 s, mientras que el observador de estados se ha ejecutado en 32.06 s en un ordenador con un procesador Intel Core i5 CPU 650 a 3.20 GHz con 4 Gb de memoria RAM.

Conclusiones y trabajo futuro

En esta tesis se han desarrollado diversos observadores de estados basados en modelos multicuerpo, y se han comparado con otras formulaciones ya existentes en la bibliografía. Estos métodos se han probado en simulación con mecanismos sencillos para evaluar tanto su comportamiento como su coste computacional. Se ha visto que los métodos que ofrecen una mejor precisión son los de la familia de los UKF, pero su coste computacional hace que no se puedan aplicar a mecanismos de tamaño moderado en tiempo real. El método más eficiente ha sido el que emplea el filtro de Kalman de formulación indirecta, llamado errorEKF, que además ha obtenido una buena precisión.

Posteriormente, el método errorEKF ha sido implementado para el modelo multicuerpo de un vehículo, y verificado con datos experimentales. Además, el tiempo total requerido por el algoritmo para la maniobra probada ha sido de 32.06 s, cuando la maniobra real ha tenido una duración de 55.744 s.

El trabajo realizado en esta tesis deja sin cubrir algunos aspectos que se pueden convertir en futuras líneas de investigación. El primero de ellos es la verificación de que el observador de estados puede funcionar a bordo del vehículo sin problemas de estabilidad, y verificando que cada paso de tiempo se ejecuta en tiempo real. Es posible que sea necesario ajustar el límite de iteraciones de la integración del modelo multicuerpo para garantizar este punto.

También queda por verificar la precisión que presentan las magnitudes que no se miden, como el ángulo de deriva. Esto se puede hacer en simulación o con el vehículo real, aunque para hacerlo con el vehículo real habría que añadir más sensores.

Como futura extensión del método errorEKF se podría añadir la estimación de fuerzas de entrada al modelo. Esto se puede conseguir añadiendo los errores de aceleración a los estados del filtro y después haciendo un estudio de dinámica inversa en el que se calculen las fuerzas necesarias para poder corregir las desviaciones en aceleración. Este método debería ser más preciso para fuerzas que varíen despacio que el modelo de camino aleatorio normalmente empleado en el filtro de Kalman aumentado.

Por último, todos los ruidos de la planta y de los sensores se han considerado blancos y gaussianos en esta investigación, pero algunos de ellos podrían no serlo, principalmente los ruidos de la planta, que suelen ser ruidos de baja frecuencia debidos a errores en los modelos de fuerzas. Añadir la caracterización estadística de este ruido puede contribuir a mejorar la precisión de los filtros.

Trabajos derivados de la realización de esta tesis

Parte de esta tesis ha sido financiada por el Gobierno de España a través de la ayuda predoctoral BES-2013-063598. De la realización de esta investigación se han derivado varios trabajos, incluyendo artículos en revistas científicas, y participaciones en congresos nacionales e internacionales. Además, se han enviado dos artículos más a revistas científicas y están en proceso de revisión. La lista de trabajos se detalla a continuación:

Artículos de revista

Roland Pastorino, Emilio Sanjurjo, Alberto Luaces, Miguel Á. Naya, Wim Desmet, y Javier Cuadrado. Validation of a Real-Time Multibody Model for an X-by-Wire Vehicle Prototype Through Field Testing. Journal of Computational and Nonlinear Dynamics, 10(3):031006, 2015.

José L. Torres-Moreno, José L. Blanco-Claraco, Antonio Giménez-Fernández, Emilio Sanjurjo, y Miguel Á. Naya. Online Kinematic and Dynamic-State Estimation for Constrained Multibody Systems Based on IMUs. Sensors, 16(3):333, 2016.

Artículos de revista enviados

Emilio Sanjurjo, Miguel Á. Naya, Javier Cuadrado, y Arend Schwab. Roll Angle Estimator Based on Angular Rate Measurement for Single Track Vehicles. Vehicle System Dynamics (en revisión).

Emilio Sanjurjo, Miguel Á. Naya, José Luis Blanco-Claraco, José Luis Torres-Moreno, y Antonio Giménez-Fernandez. Multibody-Based State Observers: a Review and Benchmark (en revisión).

Comunicaciones en congresos

Emilio Sanjurjo, Roland Pastorino, Daniel Dopico, Miguel Á. Naya. Validación Experimental de un Modelo Multicuerpo de un Prototipo de Vehículo Automatizado, en XIX Congreso Nacional de Ingeniería Mecánica, Castellón, España, 2012.

Emilio Sanjurjo, Roland Pastorino, Pasquale Gallo, Miguel Á. Naya. Implementation Issues of an On Board Real-Time Multibody Model, en 3rd Joint Int. Conference on Multibody System Dynamics (IMSD 2014) and 7th Asian Conference on Multibody Dynamics (ACMD 2014), Busan, Corea, 2014.

José L. Torres-Moreno, José L. Blanco-Claraco, Emilio Sanjurjo, Miguel Á. Naya, Antonio Giménez-Fernández. Towards Benchmarking of State Estimators for Multibody Dynamics, en 3rd Joint Int. Conference on Multibody System Dynamics (IMSD 2014) and 7th Asian Conference on Multibody Dynamics (ACMD 2014), Busan, Corea, 2014.

Emilio Sanjurjo, Alfonso Varela, Miguel Á. Naya. Modelización de los Subsistemas de Tracción y Frenado de un Automóvil para su Uso en un Simulador, en XX Congreso Nacional de Ingeniería Mecánica, Málaga, España, 2014.

Emilio Sanjurjo, José L. Blanco-Claraco, José L. Torres-Moreno, Miguel Á. Naya. Testing the Efficiency and Accuracy of Multibody-Based State Observers, en ECCOMAS Thematic Conference on Multibody Dynamics 2015, Barcelona, España, 2015.

José L. Torres-Moreno, José L. Blanco-Claraco, Emilio Sanjurjo, Antonio Giménez-Fernández, Miguel Á. Naya. A Testbed for Benchmarking State Observers in Multibody Dynamics, in ECCOMAS Thematic Conference on Multibody Dynamics 2015, Barcelona, España, 2015.

Emilio Sanjurjo, Edoardo Sinigaglia, Miguel Á. Naya. Multibody-based State Observer for Navigation Applications, en 4th Joint Int. Conference on Multibody System Dynamics (IMSD 2016), Montreal, Canadá, 2016.

Emilio Sanjurjo, Edoardo Sinigaglia, Miguel Á. Naya. Observador de Estados Basado en Modelo Multicuerpo Aplicado a un Vehículo Automóvil, en XXI Congreso Nacional de Ingeniería Mecánica, Elche, España, 2016 (pendiente de ser presentado).